Image Captioning with voice for visually impaired

Ranjith Cheguri
Computer Software Engineering
San Jose State University
(ranjithkumaryadav.cheguri@sjsu.edu)

Vinay Kovuri
Computer Software Engineering
San Jose State University
(vinay.kovuri@sjsu.edu)

Sai Harshith Reddy Gaddam
Computer Software Engineering
San Jose State University
(sairharshithreddy.gaddam@sjsu.edu)

Nachiket Wattamwar
Computer Software Engineering
San Jose State University
(nachiket.wattawar@sjsu.edu)

Abstract

Describing an image content with voice could help visually impaired people to a greater extent in perceiving the surroundings. Many voice assistants like siri, alexa and google assistant are available to convert text data to voice but we in this project have developed a service which can describe the images and read out so that they can understand better about the things going on. Generating captions is a challenging task as it should combine natural language processing and computer vision. A deep recurrent network is modelled with Convolution neural network (CNN) for generating features and Long Short Term Memory network (LSTM) for mapping words and then form sentence formation in english are used to achieve this. The model designed has attained BLEU-1 score probability of around 0.59.

Keywords: VGG, CNN,RNN, image captioning, natural language description

1. INTRODUCTION

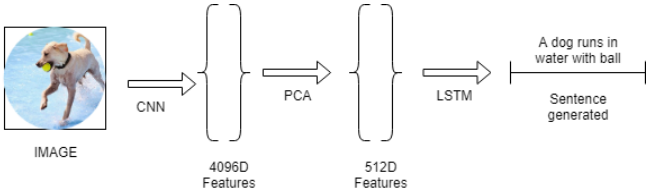
Vision impairment means inability to perfectly perceive the things as they are and this decreased ability to see things cannot be aided with spectacles or contact lenses. over 290 million people are visually impaired in the world, this project model integrated with the mobile or web application can be very beneficial. Current screen readers and braille displays provide aid to some extent but with the rapid advancements in the technology and increased use of pictures we see another technology that can provide increased satisfaction is this audio generated image captioning. Nowadays every mobile has a camera functionality with which our user can take a photograph of the surrounding and get the audio generated caption using our service. There seems a lot of scope in web, a visually impaired user browsing the web with the service integrated in browser it can describe the image as he goes through the content, Imagine a person opening a blog article about social service and the browser reads all the content even with images describing for example: ‘a group of people helping the flood victims’ or like ‘a person distributing the milk packets’.

2. RELATED WORK

There are many approaches [10,23,27,47,9] proposed for image captioning model in the literature. In the very beginning of image captioning, there were models for object detection, which describe all objects in the image. Then after the concept of neural networks, there were many papers proposed regarding encoder-decoder framework[15,12], which works in the way of converting the image into arrays and then mapping the images to sequences, which are natural language models. So, the image will be fed to CNN[6,9] and their output will be given to RNN networks, which then provide natural language description of the images. However, there is a drawback for this approach, as the vocabulary size will be very less if there is lesser dataset size. Dense captioning [16]was another approach proposed in order to handle localization tasks simultaneously. Ranzato[35] also proposed one of the finest approaches sequence level training algorithm.

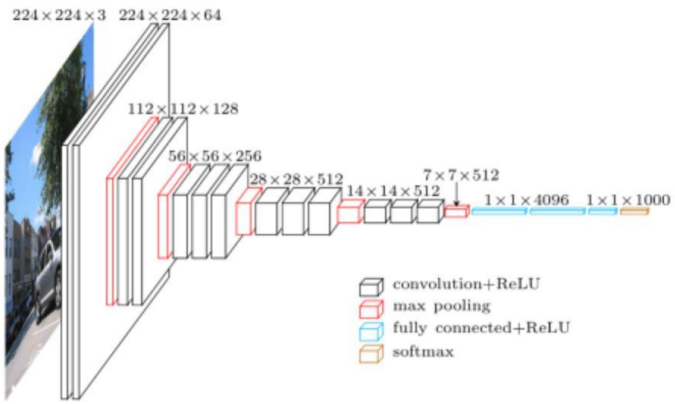
Decision-making is one of the popular approach used in the gaming industry at present. The agent will be undergone reward-based training with the environment, which gets a positive reward, if the prediction is a correct and negative reward if the prediction is wrong. Recently, there were many achievements using reward-based training such as Alpha Go, Google machine has won against top gamer. But reward based has not been implemented till now with image captioning. In this paper, proposed a model, which is implemented using a policy network and value network with visual embedding layer. It works with the basic encoder-decoder framework but also with lookahead global guidance, which increases vocabulary size and also gets rid of local guidance word prediction issue.

In this project image features are generated with CNN, the last layer of the pre-trained model VGG16 is popped and used as input for our current model. 4096 dimensional generated image is fed as input to PCA (principal component Analysis) to compress the size. The description of the model is given below.



3. IMPLEMENTATION

We have used VGG16 pre-trained image classifier model. VGG16 model was first developed during an image categorization challenge called ImageNet Large scale visual recognition challenge in 2014. VGG stands for Oxford Visual Geometry Group and 16 in VGG16 implies the number of layers present in the network. Due to lack of hardware resources, it is not easy to build a model by ourselves. This pre-trained model can be used for any image classification problem we wish to solve. This model has been built on high-end computation systems and this model was trained on 10 lakh images and has been used for many image classification problems. The model has ConvNet layers and Max pooling is used to reduce the volume size. The next 2 layers are fully connected layers with 4096 nodes which are followed by a softmax classifier. The input size for this model is 224*224 RGB image and produce an output layer of shape (X,4096), describes the classification of 4096 different objects. For example, the classification of food items such as a sandwich, pizza etc.



3.1 Implementation of Convolution Neural Network for Image scaling:

As image classification and captioning are challenging tasks nowadays. We have used Flickr 30k images for training the model. As there are plenty images and training all these images on such deep neural network would be time-consuming and also requires very costly resources. As an exception, we used the pre-trained model of VGG16, which consists of 15 deep neural network layers with CONV2D, Max-pooling layers. As the output of this pretrained model would be (1,4096), which defines VGG16 could provide a classification of 4096 objects for a single image provided. As ours is not an image classification problem, but captioning hence the addition of our own model to VGG16 could lead us to the prediction model.

As described above, the given image has been undergone with various pre-processing and also the case with the descriptions of each and every image. The final image output from the pre-processing would be the size of (1,4096) array, which would be the input of our model, as the last layer of VGG16 model would be the size of (1,4096). Below is the model architecture on which images are trained and developed captions for all those images.

Initially, the pre-processed image will be fed to the input_1 layer, which is the similar last layer of VGG16 pre-trained model. As every deep neural network model will face weights drain issue, as the number of layers in the network increases. So there's an ideal concept of Dropout to efficiently decrease the perceptrons in each layer. Here, we have used Dropout of 50%, which represents that on each successive iteration of the image on the model, 50% perceptrons will be removed, which would help with regularization of weights. We initially trained in different dropouts with 10%, 20%, and 60%. Ideally, we got good results at Dropout 50%.

In the next layer, we used a Dense layer, which reduces the features extracted to (X,256). As it is ideally 1/16 factor and which would be ideal to reduce the features instead of reducing the features into 4 factor, as it will be an additional 2 layers and eventually reduce the training performance. So the output of this layer would be of size (X,256).

3.2 Implementation of Recurrent Neural Networks for embedding Sequences:

On the other hand, created another model for generating sequences, in which each description is of length 34. As for each image there would be 5 descriptions and the maximum limit for each image is of 34 characters. We used input layer to insert into the model and the output of this layer also would be (X,34). As in our model, we merged both models CNN and RNN, in order to map the outputs of both models, we need to do padding and embedding for both models output layers as the concat layer expects both inputs of the same size.

3.3 Concatenation of CNN and RNN:

So, introduced Padding layer on CNN side converting (X,256) size to (X,34,256) and embedding layer on RNN side which converts from (X,34) to (X,34,356). So padding of 34 will be done to CNN the last layer and embedding of 256 applied to RNN model the last layer results in both the outputs to the same shape of (X,34,256). These two models are concatenated using concatenate method of keras framework, which would be eventually fed to LSTM layer.

The output of the Concatenate layer would be of a shape (X,34,512)

Finally, applying the LSTM layer to Fully connected layer to produce natural language descriptions. We have tried with different outputs for each layer with regularization. The model described above is the final model which acquired very good results of the BLEU score of 0.58.

4. EXPERIMENTS

For experiments, we train the model and we test it with different pictures taken from the surrounding. To make sure it has a use case with proper purpose we need to make sure it is easily accessible to the end user. For this, it is necessary to test and experiment it with various input data and compare the results obtained.

4.1 DATA

The data used here is flickr8K_set which has images along with respective captions for each image. The model uses this data to learn from it and generate relevant captions for new images.

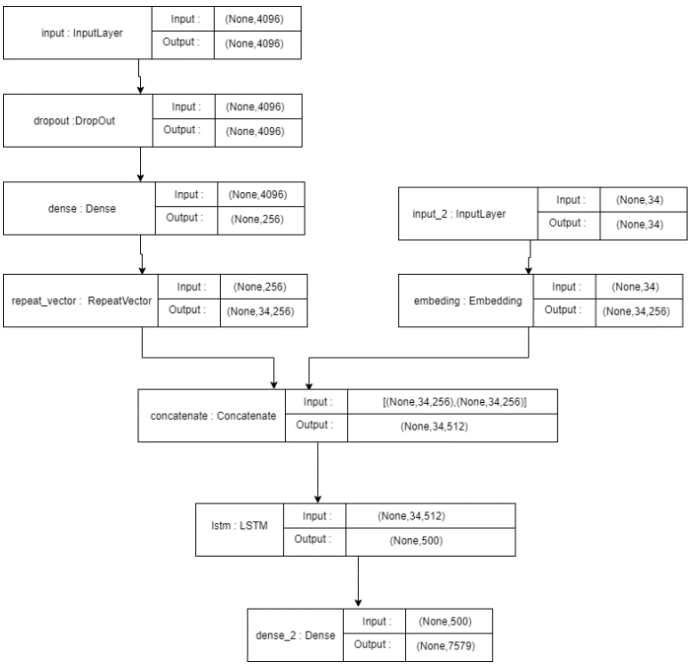
Converting the images into model understandable is done by the initial steps of the training model. The cleanliness of the data determines the accuracy of the output of the model. Using different data on the same model and training it for proper weights increases the accuracy of the model.

4.2 Analysis of the quantitative aspect of the model:

When the model is trained with a certain type of data then the output produced when given a different image is not to the accuracy of the ideal model. If the quantity of the input data is less then the model poorly trained and produces irrelevant results. However when the model is trained with loads of data then the overfitting occurs and the result is relevant.

4.3 Output Analysis

The model should output data and it should be relevant to the human intuition when one sees an image. This analysis helps to determine the actual utilization of the model for practical applications. Weighing the output accordingly determines the error factor necessary to know the quality of the model.



5. CONCLUSION

In our project, we have tried to develop a prediction model to generate the captions and converting the caption to speech. To help the end user use our project we have developed an android application through which an image can be captured to generate the captions in real time. After the model was trained using Flickr8k dataset, we were able to achieve an accuracy of 59% (BLEU-0.59). The accuracy can further be improved by training the model using a bigger dataset and higher epochs. We have made an attempt to solve the problem of visually impaired people who have difficulty to get a sense of the surroundings. Future work can include generation of captions from videos which can help the visually impaired to get real-time caption generation. We can integrate our work with the Google glass which has a camera inbuilt in it which will identify everything you see through the google glass helping out the visually impaired.

6. REFERENCES

[1] Y. Bengio, R. Collobert, and J. Weston. Curriculum learning. In ICML, 2009. 4

[2] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. In arXiv:1504.00325, 2015. 5

[3] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, Li-Jia Li, Snap Inc., Google Inc. Deep Reinforcement Learning-based Image Captioning

[4] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoderdecoder for statistical machine translation. In EMNLP, 2014. 2

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In arXiv:1412.3555, 2014. 2, 5

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierachical image database. In CVPR, 2009. 2

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In CVPR, 2015. 1, 2, 6

[8] D. Elliott and F. Keller. Image description using visual dependency representations. In EMNLP, 2013. 2

[9] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From from captions to visual concepts and back. In CVPR, 2015. 2

[10] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In ECCV, 2010. 2