ICICC 2021

Session ID : SS_22

AUTOMATIC IMAGE CAPTIONING USING DEEP LEARNING

*Parul Diwakar

Affiliated to Guru Gobind Singh Indraprastha University

Email id - paruldiwakar700@gmail.com

Abstract

Past years have definitely been very crucial to development of artificial intelligence and scene recognition has become the most contributed field of Computer Vision, and it has attracted the attention of many young researchers. Generating natural language descriptions automatically according to the content observed in a picture, is an important part of scene understanding, which combines the knowledge of computer vision along with natural language processing. The application of image captioning is vast and profound. Tremendous progress in scene recognition problems is credited to the availability of large databases like MS COCO, Places,ImageNet, etc, and the development of CNNs (Convolutional Neural Networks) for gathering high-level features. This paper aims to describe the methodology used and possible improvements.

1. Introduction

Processing the world in a single glance is one of the most accomplished features of the human brain; it takes only a few tens of milliseconds to categorise an object and/or its environment, emphasizing an important role of feedforward processing in visual recognition. One of the mechanisms responsible for efficient visual recognition by humans is our susceptibility to understand and retrieve scenarios and corresponding information by perceiving our environment regularly. Brain constantly registers new inputs every now and then, developing its own extensive dataset.

Writing a set of instructions to perform such a complex task i.e taking an image as an input and generating a relevant caption as the output was thought of as an implausible problem even by the most advanced researcher in the field of Computer Vision. But with the emergence of Deep Learning Algorithms, this problem can be solved if we have the appropriate dataset. CNNs were inspired by the hierarchical patterns of layers and increasing processing complexity with the depth of the primate brain. These architectures together with recent large databases (e.g., ImageNet) have obtained astonishing performance on object classification tasks(Karpathy, Toderici, Shetty, Leung, Sukthankar1 & Li ,2014) .

1.1Need

It is important to understand how this task is relevant to the real world . Few applications where a solution to this problem can be very useful (Aditya Upadhyaya ,2016).

- improving Google Image Search, as then every image could be first converted into a caption and then search can be performed using it.
- Aid to the blind — Converting images to text and text into voice to guide them.
- Self driving cars — Automatic driving is one of the biggest challenges and if we can properly caption the scene around the car, it can make the system more powerful.
- Using CCTV cameras and caption generation we can raise the alarms as soon as there is detection of some questionable activity.

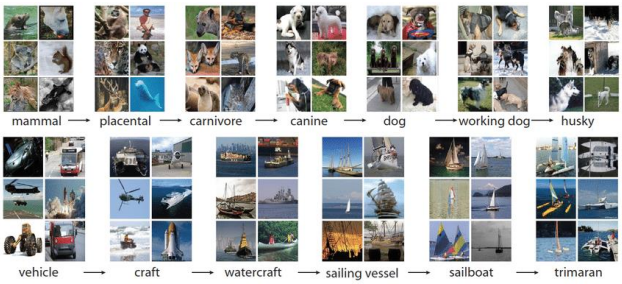
Some of the ongoing applications of the similar kind are given as follows[2].

- Google Photos: Classify your photo into Mountains, sea etc.
- Facebook: Using AI to classify, segmenting and finding patterns in pictures.
- FedEx and other courier services have been using handwritten digit recognition system to detect pin code correctly.
- Picasa : Using facial Recognition to identify your friends and you in a group picture.
- Tesla/Google Self Drive Cars: All the self drive cars are using image/video processing with neural networks to attain their goal.

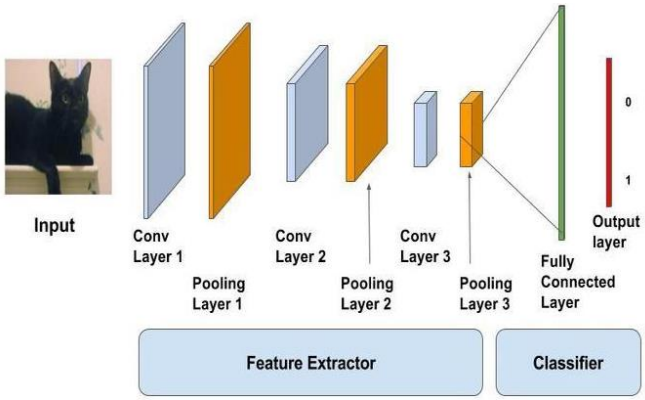
This paper follows this pattern: in Section 2 ImageNet database is briefly introduced along with its construction. In Section 3, brief introduction to CNNs and ResNet50. In Section 4, introduction to Recurrent Neural Networks(RNNs) and LSTM. In Section 5, Flickr8k Dataset(used to train the model). In Section 6, methodology used , Section 8 discussing the outputs generated by the model and finally the conclusion and future scope.

2. Imagenet Database

ImageNet is an image database with a total of 14 million images and 22 thousand visual categories. It is publicly available for research and educational use, and has played an important role in the deep learning revolution. ImageNet was constructed in 2009 through Internet search and crowdsourcing. The research team obtained a vocabulary of categories from WordNet (Stanford Vision Lab, Stanford University, Princeton University , 2016).



3. ResNet50 CNNs



A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance to various features in the image and be able to differentiate one image from the another. Pooling layer is

responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction.

Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus contributing to the process of effectively training the model.

ResNet-50 used for images is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

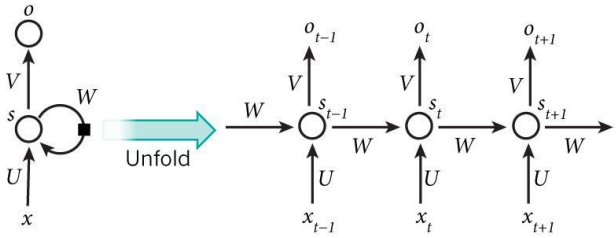
4. Recurrent Neural Networks (RNNs)

Recurrent neural networks (RNN) are a class of neural networks that is powerful for modelling sequence data such as time series or natural language. After producing the output, it is copied and sent back into the recurrent network. For making

Fig. 3 LSTM

a decision, it considers the current input and the output that it has learned from the previous input(Denny, Britz 2015).

The Long Short Term Memory(LSTMs) are a class of RNNs to process the sequence input (for example partial captions).



5. Flickr8k Dataset

There are many open source datasets available for this problem, like Flickr 8k (containing8k images), MS COCO (containing 180k images), Flickr 30k (containing 30k images), etc.

This dataset consists of 8k images each , all images having 5 captions per image. These images are divided into two sets as follows:

- Training Set — 6k images
- Test Set — 2k images

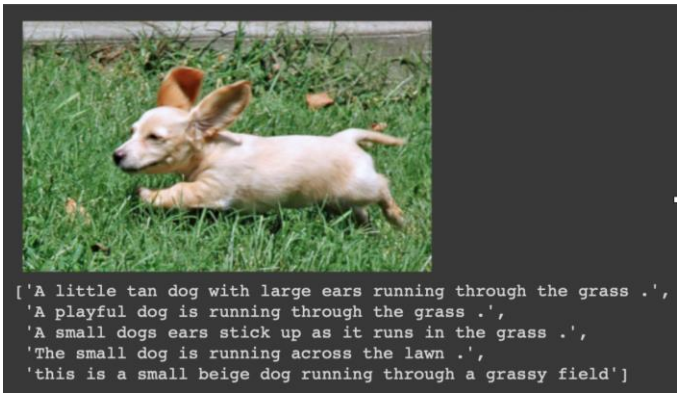


Fig. 4 Image with its 5 corresponding caption

The training set is fed to the model to train it and finally the model is tested on the testing data to generate captions.

6.Methodology Used

Understanding of Machine Learning and Deep Learning concepts like Image Processing and manipulation, Multi-layered Perceptrons, CNNs, RNNs , Automated Feature Engineering, Gradient Descent, Forward propagation, Back propagation, Overfitting, Applied Probability, Text Processing, Python syntax and data structures, Keras library, etc is required .

6.1 Data Collection

Flickr8k dataset consists of 8k images, all images having 5 captions per image.

- Training Set : 6k images
- Test Set : 2k images

Along with images, we receive some text files related to the images as well. One of the files is "Flickr8k.token.txt" that contains the name of each image and its five corresponding captions. The data is presented in the "Flickr8k.token.txt" file as <image name>#i <caption>, i□[1,4] i.e. the name of the image, caption number and the caption.

6.2 Data Cleaning

Following cleaning operations were performed on the captions to make the model robust to outliers and make less mistakes :

1. Words are converted to their lower cases.
2. Removing punctuations.
3. Removing words of length less than 2.
4. Can remove special symbols like @,<,# etc.

Only those words which occur more than or equal the threshold set in the corpus. For example if we take threshold = 10, we get a Vocabulary of 8424 words and Total Words 373837.

6.3 Loading of Training and Testing sets

The text file "Flickr_8k.trainImages.txt" contains contains the reference to the images in the training set. Two tokens in all captions must be added as follows:

<s> —> Start sequence token which will be added at the beginning of every caption.

<e> —> End sequence token which will be added at the end of every caption.

The text file "Flickr_8k.testImages.txt" contains the reference to the images in the testing set.

6.4 Preprocessing Image Data

Input to a Machine Learning model is provided in the form of a vector. For making the processing and predicting process easy for the model, each image is needed to be converted into a fixed sized vector which can then be fed as input to the respective neural network. We can use transfer learning by using the ResNet50 model. Using automated feature engineering, we get a fixed length relevant vector for each image. Removal of the last softmax layer from the model to obtain a vector of length 2048.

6.5 Preprocessing Caption Data

The caption will be predicted word by word. Captions are treated like target variables that the model is learning to predict. Since the caption will be predicted as one word at a time, each word is encoded into a fixed sized vector. All the unique words are

Fig. 4 Image with its 5 corresponding caption

mapped to an index and vice versa including the tokens added.

6.6 Word Embeddings

Each word/index is mapped to a 50-length vector using a pre-trained GLOVE word embedding model. Each sequence contains 35 indices, where each index is a vector of length 50.

Therefore x = 35*50 = 1750. Where 35 is the length of the maximum length of a caption available in the training data.

6.7 Model Architecture

As the input consists of a partial caption and an image vector Functional API is used which allows us to merge Models.

input_image (224x224 —> 2048 —> 256 dimensions)
input_caption(batch_size x 35 —> batch_size x 35 x 50 —> LSTM —> 256 dimensions)
input_image + input_caption —> 256 dimensions —> 1848 dimensions —> softmax —> probable_word

6.8 Caption Generation

The vocabulary this example uses = {and, black, dog, in, runs, snow, the, white}

The caption is generated iteratively, one word per iteration:

Iteration 1:

Input: Image vector + "start" (as partial caption)

Probable word: "black"

Iteration 2:

Input: Image vector + "start black"
Probable word: "and"

Iteration 3:

Input: Image vector + "start black and"
Probable word: "white"

Iteration 4:

Input: Image vector + “start black and white”
Probable word: “dog”

Iteration 5:

Input: Image vector + “start black and white dog”
Probable word: “runs”

Iteration 6:

Input: Image vector + “start black and white dog runs”
Probable word: “in”

Iteration 7:

Input: Image vector + “start black and white dog runs in”
Probable word: “the”

Iteration 8:

Input: Image vector + “start black and white dog runs in the”
Probable word: “snow”

Iteration 9: Input: Image vector + “start black and white dog runs in the snow”
Probable word: “end”

Fig. 5 Test image
Caption Generated: black and white dog runs in the snow .

7.Outputs

Below are a few examples of the captions generated by the model.

Fig. 6 the dog jumps over the bar.

Fig. 7 the dog jumps over bar .

Fig. 7 two women dressed in dresses .

Fig. 8 snowboarder is skiing down snowy hill .

Fig. 11 two children are skiing down snowy hill .

Fig. 9 man in blue shirt is jumping into the air .

Fig. 10 black and white dog is running through the grass .

8.Future Scope & Conclusion

The captions predicted, though were relevant for most images but sometimes did not match the content of the image. This can be solved by training the model on a larger dataset.

This was a basic project and is capable of a lot of improvements. Some of the the modifications that can improve the model are as follows (Harshall Lamba, 2018):

- Feeding the model with a larger dataset like Flickr30k.
- Hyper parameter tuning (for example, batch size, dropout rate, number of layers, learning rate, batch normalization etc.).
- Use of cross validation set to check overfitting.
- Using BLEU Score to evaluate and measure the performance of the model.
- Writing the code in a proper object oriented way.

Fig. 8 snowboarder is skiing down snowy hill .

Fig. 7 the dog jumps over bar .

Fig. 9 man in blue shirt is jumping into the air

Machine Learning and Deep Learning is an ever growing and has caught interest among a lot of students. Hence more and more viable and likewise more models are being developed regularly with higher accuracy. We can expect more advance image captioning since it has such important applications.

References

Harshall Lamba (2018). towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8

Stanford Vision Lab, Stanford University, Princeton University (2016). http://www.image-net.org

Aditya Upadhyaya (2016) https://www.quora.com/What-are-10-15-applications-of-image-captioning-Deep-Learning

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar1 & Li Fei-Fei (2014). https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Karpathy_Large-scale_Video_Classification_2014_CVPR_paper.pdf

DennyBritz(2015).https://www.kdnuggets.com/2015/10/recurrent-neural-networks-tutorial.html