EasyChair Preprint

№ 5740

RICA: Real-Time Image Captioning Application

Suraj Dahake, Aditya Ohekar, Shubham Ilag and Aasim Shah

June 7, 2021

RICA: Real-Time Image Captioning Application

Suraj Dahake, Aditya Ohekar, Shubham Ilag and Aasim Shah
Department of Information Technology
Datta Meghe College of Engineering, Airoli, Navi Mumbai, India - 400708
Email: {surajdahake1, adityaohekar, shubham.ilag123, shah.aasim21}@gmail.com

***Abstract*—Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English. The recent advances in Deep Learning based Machine Translation and Computer Vision have led to excellent Image Captioning models using advanced techniques like Deep Reinforcement Learning. While these models are very accurate, these often rely on the use of expensive computation hardware making it difficult to apply these models in real time scenarios, where their actual applications can be realised. In this paper, we carefully follow some of the core concepts of Image Captioning and its common approaches and present our simplistic encoder and decoder based implementation with significant modifications and optimizations which enable us to run these models on low-end hardware of hand-held devices. We also compare our results evaluated using various metrics with state-of-the-art models and analyze why and where our model trained on MSCOCO dataset lacks due to the trade-off between computation speed and quality. Using the state-of-the-art TensorFlow framework by Google, we also implement a first of its kind Android application to demonstrate the real time applicability and optimizations of our approach.**

1. Introduction

For a machine to be able to automatically describe objects in an image along with their relationships or the actions being performed using a learnt language model is a challenging task, but with massive impact in many areas. For example it could help people with visually impairment better understand visual inputs, thereby acting as an assistant or a guide.

Not only must the model be able to solve the computer vision challenges of identifying the objects in an image, but it must also be intelligent enough to capture and express the object's relationships in natural language. So, image caption generation has long been considered as a difficult problem.

Image captioning can be used for a variety of use cases such as assisting the blind using text to speech by real time responses about the surrounding environment through a camera feed, enhancing social media experience by converting captions for images in social feed as well as messages to speech. Its purpose is to mimic the human ability to comprehend and process huge amounts of visual information into a descriptive language, making it an attractive problem in the field of AI. Assisting young children in recognizing objects as well as learning the English language. Captions for every image on the internet can lead to faster and descriptively accurate image searches and indexing. In robotics, the perception of the environment for an agent can be given a context through natural language representation of the environment through the captions for the images in the camera feed.

2. Related Work

In this section we take a look at some of the work previously undertaken in this problem domain. Earlier image captioning methods relied on templates instead of a probabilistic generative model for generating the caption in natural language. Farhadi et al. (2010) [3], use triplets of <object, action, scene> along with the predetermined template to generate caption. They discriminately train a multi label Markov Random Field to predict the values of the triplets. Kulkarni et al. (2011) [4], detect objects from the image, predict a set of attributes and prepositions (spatial information against other objects) for every object, construct a labelled Conditional Random Field graph and generate sentences using the labels and a template. These approaches do not generalize well as they fail to describe the previously unseen composition of objects even if the individual objects were present in the training data. Also, the problem with template approach is their proper evaluation.

3. Architecture

3.1. Model

In our implementation we follow an approach similar to Show and Tell [1] by introducing an encoder-decoder architectural system. We have used CNN and LSTM algorithms to train the model. The encoder being the pretrained InceptionV4

Convolutional Neural Network by Google [16] and

. with Long Short Term Memory Cells. Encoder InceptionV4 is used to transform raw images *I* into a fixed length embedding *F* which represent the convolved features for the images. These embeddings are obtained by running a forward pass till the penultimate layer i.e., the *average pool layer* of the InceptionV4 model.

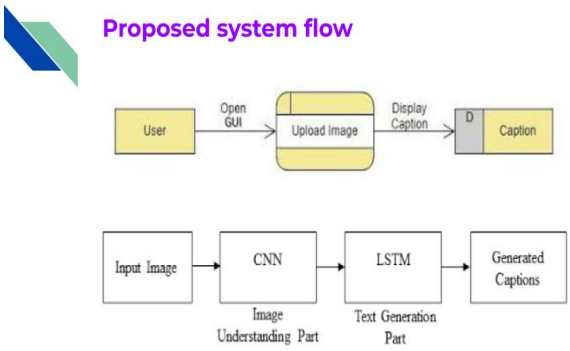
The decoder in our model has two phases, namely, training and inference. The decoder is responsible for learning the word sequences given the convolved features and original caption. The decoder's hidden state h_t is initialised using these image embeddings features *F* at timestep $t = 0$. Hence the basic idea of encoder-decoder model is demonstrated by the following equations.

$F = encoder(I)$; $X_{t=0} = F$; $O_t = decoder(X_{t:0 \rightarrow t})$

The training process in the RNN with LSTM Cell based decoder works on a probabilistic model in which the decoder maximizes the probability of word *p* in a caption given the convolved image features *F* and previous words $X_{t:0 \rightarrow t}$. To learn the whole sentence of length *N* corresponding to the features *F,* the decoder uses its recurrent nature to loop over itself over a fixed number of timesteps *N* with the previous information (features and sampled words at timestep *t*) stored in its cell's memory as a state. The decoder can alter the memory C_t as it unrolls by adding new state, updating or forgetting previous states through the LSTM's forget f_t, input i_t and output o_t memory gates.

3.2. Application

In our implementation we customize and optimize our encoder-decoder model to function in a real-time environment and to work on mobile devices. For this we use Google's numerical computation library, TensorFlow and implemented the model in Colab. The main advantage of using TensorFlow for any Deep Learning model is its data-flow graphs based computations, which basically comprise of nodes, representing mathematical operations and edges, representing the multidimensional data arrays (tensors) communicated between nodes. The flexible architecture of TensorFlow enabled us to deploy our model's computation on CPUs and GPUs and thus helped us utilize inherent parallelism in primitive operations and computations.



During the training process of the model using TensorFlow, files related to checkpoints and computation graph's definition and metadata are generated after every training epoch.

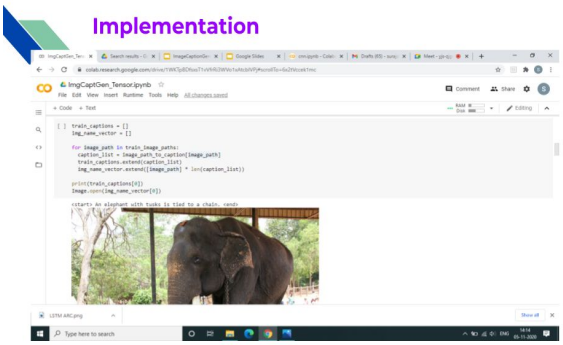


Fig. 2 - Colab model output screenshot 1

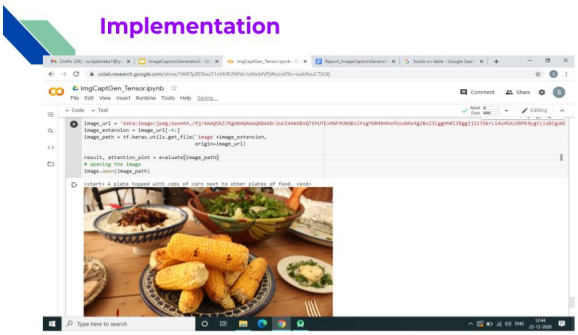


Fig. 3 - Colab model output screenshot 2

4. Experiments

4.1. Dataset

We have used the MSCOCO training dataset with 82780 images, each with five ground truth captions. For the RICA app, we have currently trained the model with 30000 images..

4.2. Implementation

We have implemented the model in the Google Colab, which gives pre-built tensorflow and keras library features.

We have also used Android Studio as we want to build and deploy the RICA application.

- Our encoder, InceptionV4, which uses residual connections, not only performs better than GoogLeNet used by Vinyals on the ImageNet task, but is also faster.
- The LSTM's hidden dimension, word and image embeddings in our model are all fixed to 256, instead of 512.
- In our model, the encoder and decoder are stitched into a single graph, hence only a single TensorFlow session needs to be loaded to run the entire model.
- We generate captions using a single trained model, instead of using an ensemble of trained models.
- To avoid the overhead of exploring the complete search space of beam search tree, we generate captions greedily, hence speeding up the real-time inference.

During training, we use the average pooling layer (after the final convolutional layer) from a pre-trained inceptionV4 to encode the image (resized to 299 × 299 × 3) resulting in a vector of dimension 1536. Currently our model supports only JPEG and PNG image formats.

5. Conclusion

We have created a RICA application to introduce possible use-cases for our model. In the near future, we will fully train the model and deploy the application. Its accuracy will be very good after this. By making use of Inception architecture and by simplifying the overall flow design, we optimize our model to perform well in real time (on mobile devices) and manage to obtain captions.

References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan (2015) Show and tell: A neural image caption generator. *CVPR 1, 2*

[2] K. Xu (2015) Show, attend and tell: Neural image caption generation with visual attention. *inProc. Int. Conf. Mach. Learn.*

[3] Farhadi A. et al. (2010). Every Picture Tells a Story: Generating Sentences from Images. *Daniilidis K., Maragos P., Paragios N. (eds) Computer Vision – ECCV 2010. Lecture Notes in Computer Science*, vol 6314. Springer, Berlin, Heidelberg

[4] Kulkarni G, Premraj V, Dhar S, Li S, Choi Y, Berg AC, Berg TL (2011) Baby Talk: Understanding and Generating Image Descriptions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (20-25 June 2011)

[5] R. Kiros and R. Z. R. Salakhutdinov (2013) Multimodal neural language models. *inProc. Neural Inf. Process. Syst. Deep Learn. Workshop*

[6] Andrej Karpathy, Li Fei Fei (2015) Deep Visual-Semantic Alignments for Generating Image Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (April 2017), vol 39, issue 4:664 – 676