THE UNIVERSITY OF ALABAMA AT BIRMINGHAM.

EE697
Graduate Project

Image Caption Generation using Transfer learning

Under the guidance of Dr. Leon Jololian, Associate Chair, Dept. of Electrical and Computer Engineering

Manish Shetty

Table of contents

1. Introduction

- Humans have the ability to interpret the environment around them without having to undergo vigorous training.

- For a machine learning algorithm to do the same task, it has to be trained with large dataset.

- With the help of Computer Vision (CV), machines can now interpret their surroundings from images or videos and mimic the complexity of human vision system.

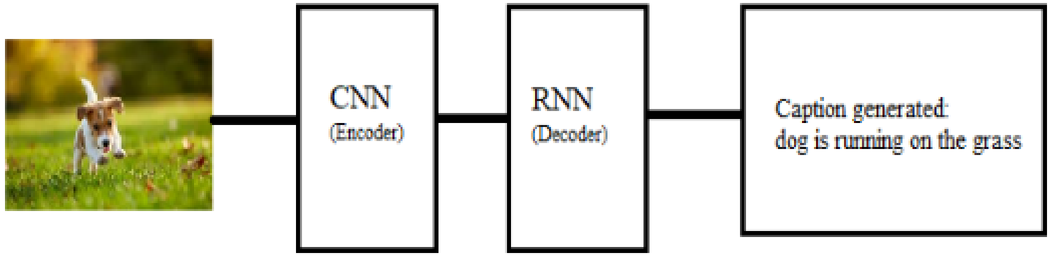

Figure 1 Basic Image captioning block diagram

- The deep learning model should be powerful to detect and understand the objects in the image and express the relation between the objects in a natural language.

- Image captioning algorithm combines two major fields of artificial intelligence: Natural Language Processing(NLP) and Computer Vision (CV).

- Figure 1 shows the simple block diagram of an Image captioning Process.

- In an Image captioning model, a Convolutional neural network (CNN) extracts the image feature from an image present.

- The model uses Recurrent neural network (RNN), to generate a description for the image with the help of the feature extracted from the CNN.

- The dataset used in this project is ‘Flickr8k dataset’. It contains 8000 images that are paired with 5 different captions for each image.

- The image captioning model, when combined with a Text-To-Speech (TTS) can be used to aid the visually impaired.

- Text-To-Speech conversion of the caption generated is implemented in this project using ‘pyttsx3’ which is a text-to-speech conversion library in Python.

2. Architecture

- With advancements in Deep learning models, state-of-the-art image captioning models are developed.
- An image captioning model uses an encoder-decoder architecture for caption generation.
- The architecture of the image captioning model is discussed below.

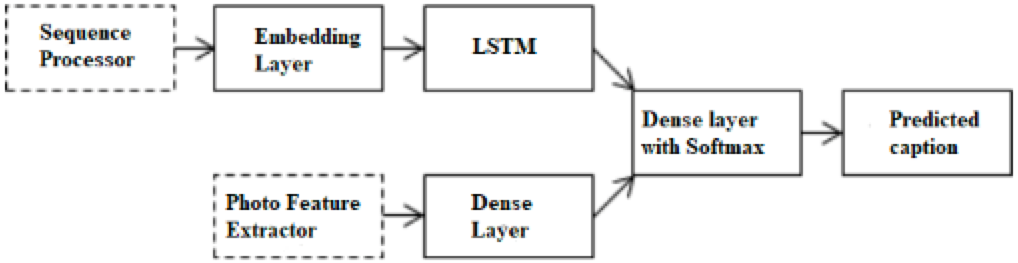

Figure 2 Merge model architecture

Merge model

- The encoder-decoder neural network uses merge model to generate captions.
- Merge model combines the encoded input image with the encoded text description.
- In this model, the CNN handles only the image vector and the RNN deals with the caption prefix.
- The image vector and the caption prefix are then merged in a separate layer which generates the caption.

Photo feature extractor

- The encoder system extracts the feature of the image which interprets the content in the photo.
- A pre-trained model VGG16 is used to pre-compute the photo features.
- The extracted features are saved to a file which is later accessed while training the model.

Long Short-Term Memory (LSTM)

- LSTM is a special type of recurrent neural network which is used to model the sequence data.
- A RNN is used to predict a text sequence based on the previous input.
- Since typical RNNs cannot store input for a long time, LSTM is used to help the RNN.
- LSTM contains computer like memory which can read, write and delete information from its memory based on its significance.

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Figure 3 VGG16 model summary

Dense Layer

- It is a layer of neural networks where each neuron receives inputs from all the neurons in the previous layer forming a dense layer.
- All the neurons in the present layer are connected to the neurons in the previous layer.
- The dense layer changes the dimensions of your vector by applying rotation, scaling, translation transform to the vector.

Embedding Layer

- It is used to process textual data in neural network models.
- Embedding layer converts the into numbers before applying to the model.
- One-hot encoding each word in a sentence would not be efficient.
- Each word is translated into a fixed size vector.

3. Methodology

- It may take days or hours to compute the photo every time the model is trained.
- Transfer learning process is used to extract the photo feature.
- It is a shortcut process where a machine learning model trained on a problem is used to a train a second model with similar problem statement or result.
- A pre-trained model VGG16 is used to extract the features from the image.
- The dataset used is Flickr8k dataset which contains 8000 images and its corresponding descriptions.
- The images in the dataset are reshaped to a size of 224x244 pixel image which is the preferred size for the VGG16 model.

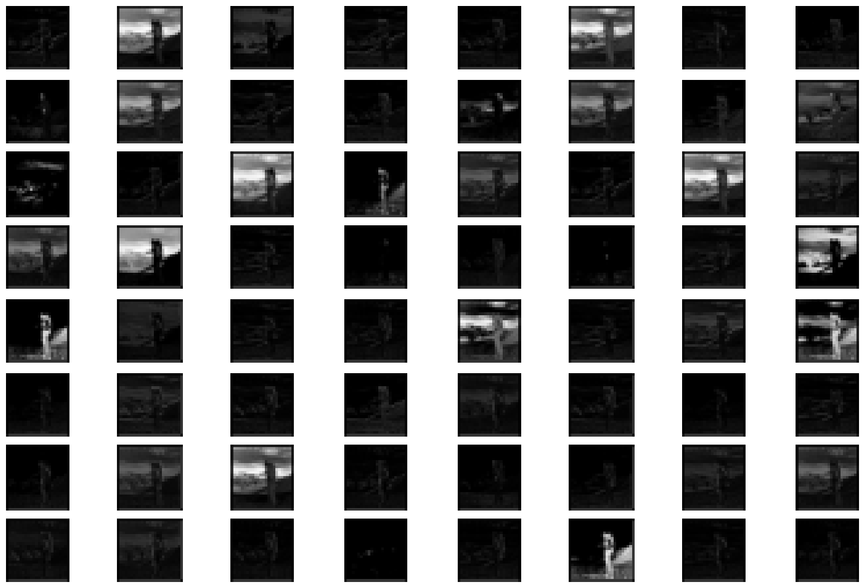

Figure 4 Feature extraction map

- The VGG16 model extracts the image features from the photos in the dataset and stores them in a file which is used later to train the model.
- The computed features are a 1-dimensional 4096 element vectors.

- The descriptions in the dataset also needs to be pre-processed before the model is trained.
- . The texts in the description are cleaned in the following way to reduce the size of the vocabulary to work with:
  - All the punctuations are removed.
  - All the characters are converted to lowercase.
  - Remove all the numbers.
  - Remove all words that are one character in length.
- The images in the dataset have their own unique identifiers. The cleaned descriptions are stored in a file with one image identifier and description per line.
- LSTM predicts the next word in the sequence based on the previous input. Two tokens ‘startseq’ and ‘endseq’ are used to start and end the generation process respectively.
- The descriptions are encoded by the embedded layer and is loaded in the model as an input.
- The pre-computed photo features are also loaded into the model as a second input.
- The image captioning model used in this project is based on ‘Merge Model’ where the images and the text description are handled by two different neural networks.
- CNN computes the photo feature and RNN encodes the text description.
- The convolutional network expects the input to be a vector of 4096 elements.

- A dense layer processes this extracted feature and produce a 256 element image representation.
- The input text descriptions are fed to an embedding layer which has a pre-defined length of 34 words.
- The output from the embedding layer is given to LSTM which produces a 256 element vector.
- A dropout rate of 50% is used to reduce the overfitting of the training dataset.

- The outputs from both input model are merged by a decoder using an addition operation.
- The output from the decoder is fed to a dense 256 neuron layer and then to a final dense layer which makes a softmax prediction of the caption.
- Figure 5 gives the general idea of the layers in the Image captioning model.
- The model is trained for 6 epochs to obtain maximum performance from the model.
- The model with the lowest loss value is used in generating caption for a new image.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 34)]	0	
input_1 (InputLayer)	[(None, 4096)]	0	
embedding (Embedding)	(None, 34, 256)	1940224	input_2[0][0]
dropout (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_1 (Dropout)	(None, 34, 256)	0	embedding[0][0]
dense (Dense)	(None, 256)	1048832	dropout[0][0]
lstm (LSTM)	(None, 256)	525312	dropout_1[0][0]
add (Add)	(None, 256)	0	dense[0][0] lstm[0][0]
dense_1 (Dense)	(None, 256)	65792	add[0][0]
dense_2 (Dense)	(None, 7579)	1947803	dense_1[0][0]

Figure 5 Image captioning model summary

- The quality of the caption generated by the trained model is evaluated using a metric Bilingual Evaluation Understudy Score (BLEU).
- It is implemented using python's Natural Language Toolkit Library (NLTK).
- It evaluates the caption generated by the trained model against a set of reference sentences.
- A score closer to 1.0 is a good result and a score closer to 0.0 is a bad result.
- The following BLEU scores were obtained for the trained model which had a low loss:
    - BLEU-1: 0.559516
    - BLEU-2: 0.308131
    - BLEU-3: 0.208197
    - BLEU-4: 0.094835

4. Results

- The feature extraction function of the VGG16 model is redesigned to extract the feature from a single photo.
- A single image is fed to the model which then generates a caption.
- Figure 6 shows the caption generated for an image in the test dataset in the Flickr8k dataset.
- Figure 7 and Figure 8 shows the generated caption for a new image which is used to test the accuracy of the model.

startseq dog is running through the grass endseq

Figure 6 Caption generated for image in test dataset

startseq man climbing up rock face endseq

Figure 7 Caption generated for a random image

- The loss of the model improves with each epoch.
- Figure 9 shows the accuracy and the loss value performance of the trained model in each epoch.
- The accuracy of the trained model increases with each epoch. In contrary to the accuracy the loss value decreases with each epoch.
- At the end of the 6th epoch, the accuracy of the trained model was 31% and the loss value was 3.31.

Figure 8 Caption generated for a new image

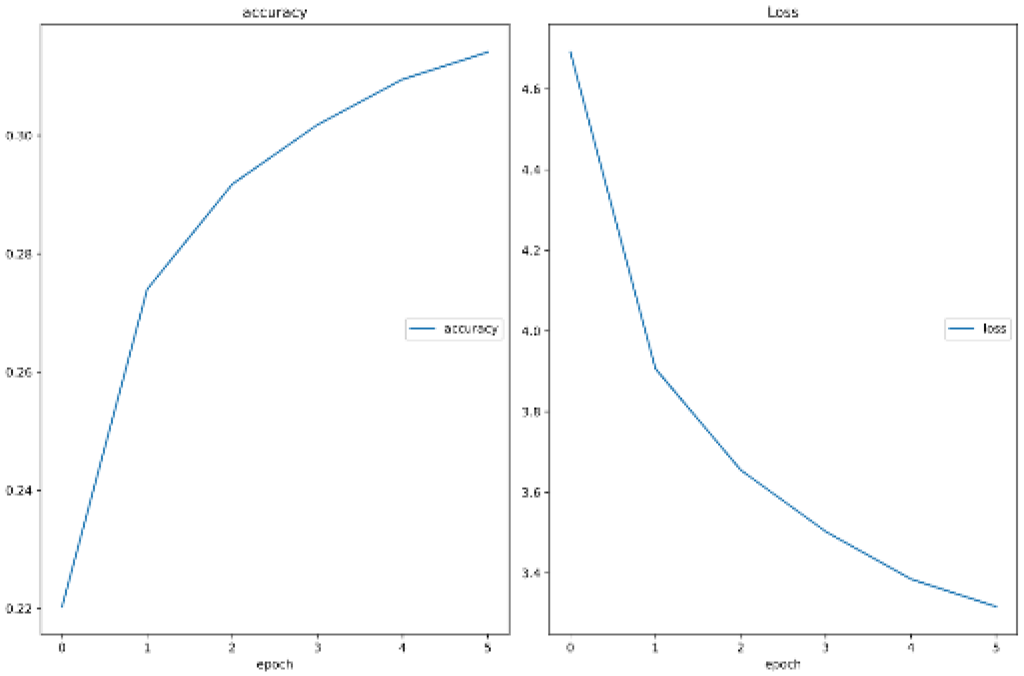

Figure 9 accuracy vs epoch and loss vs epoch plot

5. Conclusion and Future Works

- An end-to-end neural network is presented which can takes an image as an input an generates a reasonable description.
- To increase efficiency of the model, we use transfer learning method.
- Convolutional neural network is used to encode the image into a feature representation and a recurrent neural network is used to generate captions.
- The accuracy of this model is about 30% which is not a great accuracy level.
- One way to increase the accuracy is to use visual attention mechanism. The attention mechanism decides what part of the detail in the image is relevant and worth paying attention.
- Another way to improve the caption is to implement pre-trained word embeddings. These word embeddings can increase the performance of the Natural Language Processing model.
- Google’s Word2Vec and Stanford’s GloVe are the two most popular word-level pre-trained word embeddings.

References

1. L. Skovajsová, "Long short-term memory description and its application in text processing," 2017 Communication and Information Technologies (KIT), Vysoke Tatry, 2017, pp. 1-4, doi: 10.23919/KIT.2017.8109465.
2. Tanti, M., Gatt, A., & Camilleri, K. (2017, August 25). What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? Retrieved November 27, 2020, from https://arxiv.org/abs/1708.02043v2
3. V. Kesavan, V. Muley and M. Kolhekar, "Deep Learning based Automatic Image Caption Generation," 2019 Global Conference for Advancement in Technology (GCAT), BANGALURU, India, 2019, pp. 1-6, doi: 10.1109/GCAT47503.2019.8978293.
4. Hodosh, Micah, Peter Young, and Julia Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics." Journal of Artificial Intelligence Research 47 (2013): 853-899.
5. Kishore Papineni IBM T. J. Watson Research Center, et al. BLEU: a Method for Automatic Evaluation of Machine Translation. 1 July 2002, dl.acm.org/doi/10.3115/1073083.1073135.
6. Xu, Kelvin, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 19 Apr. 2016, arxiv.org/abs/1502.03044.
7. Bengio, Yoshua. Deep Learning of Representations for Unsupervised and Transfer Learning. 27 June 2012, proceedings.mlr.press/v27/bengio12a.html.

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM.

Thank you!