Image Caption using CNN & LSTM

Article · May 2020

CITATIONS
2

READS
6,101

1 author:

Ali Ashraf Mohamed
Helwan University
1 PUBLICATION **2** CITATIONS

SEE PROFILE

Image Caption using CNN & LSTM

1.Introduction :-

Nowadays , Machine learning is a trend in Artifical Intelligence . Recently , we apply AI in building a powerful performance and highly intelligent machines . Machine learning has a subset called deep learning , it provides high accuracy with its results so its performance is high too through its output . In our research paper , deep learning is used in apps of image description . Image description provides the process of describing the content from an image . The idea is based on the detection of objects and what actions in the input image . Bottom-up and top-down approaches are two main approaches of image description . Bottom-up approaches generate contents in an input image , and then combine them into a caption . Top-down approaches generate a semantic representation of an input image that is then decoded into a caption using various architectures like recurrent neural networks . Image description could have many benefits , for instance by helping visually impaired people better understand the content of images on the web . Now , we will explain what exactly will happen . What do you see in the below picture?

Well some of you might say “A black dog in a grassy area”, some may say “ black dog with white spots” , and others might say “ A dog on grass and some yellow flowers “ . All of these captions are relevant for this image of course and there may be some others too . The point I want to show; it’s so easy for us, as humans , to have a glance at a picture and describe it . But , can a computer program produces a relevant caption as humans ?

2.Related Work : -

In this section , we will talk about the experimental results were carried out by MSCOCO dataset . For encoder/decoder framework , they have added a feature called guiding network in their proposed work . The method that called guiding network , mainly deals to learn the vector by a neural network $v=g(A)$ where A is the set of annotation vectors .

Generating natural language descriptions from visual data is an important problem . it has long been studied in computer vision . Hence, this had led to complex systems consists of visual primitive recognizers combined with a structured formal language like And-Or Graphs or logic systems .Recently, the problem of still image description with natural text has gained a huge interest.

3. Methodology:-

Here We use CNN and LSTM to achiave our goal (image caption generator)

we start from what is CNN and how can benefit from it in our problem ?

Convolutional Neural Network is an artificial deep learning neural network. It is used for image classifications , computer vision ,image recognition and Object detection.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat,etc). It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images.

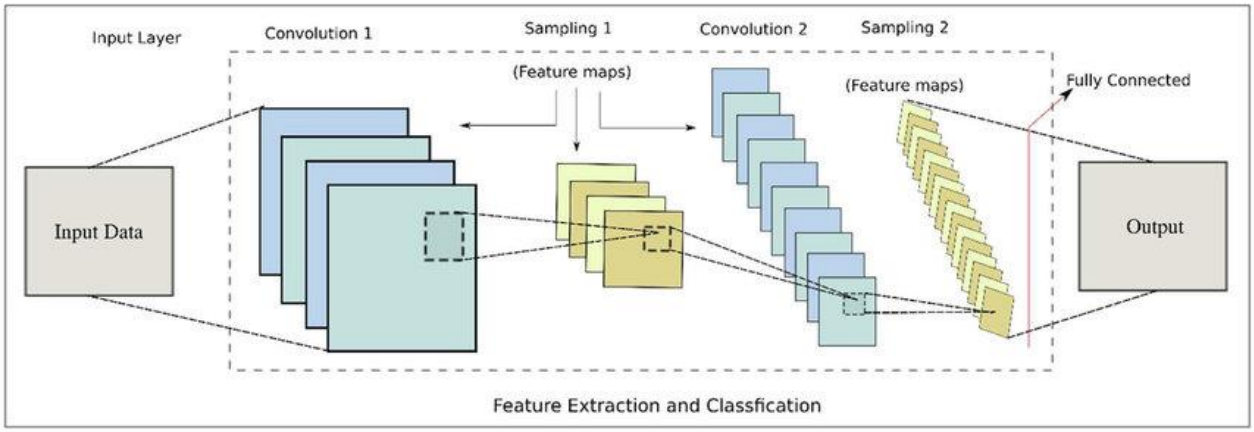
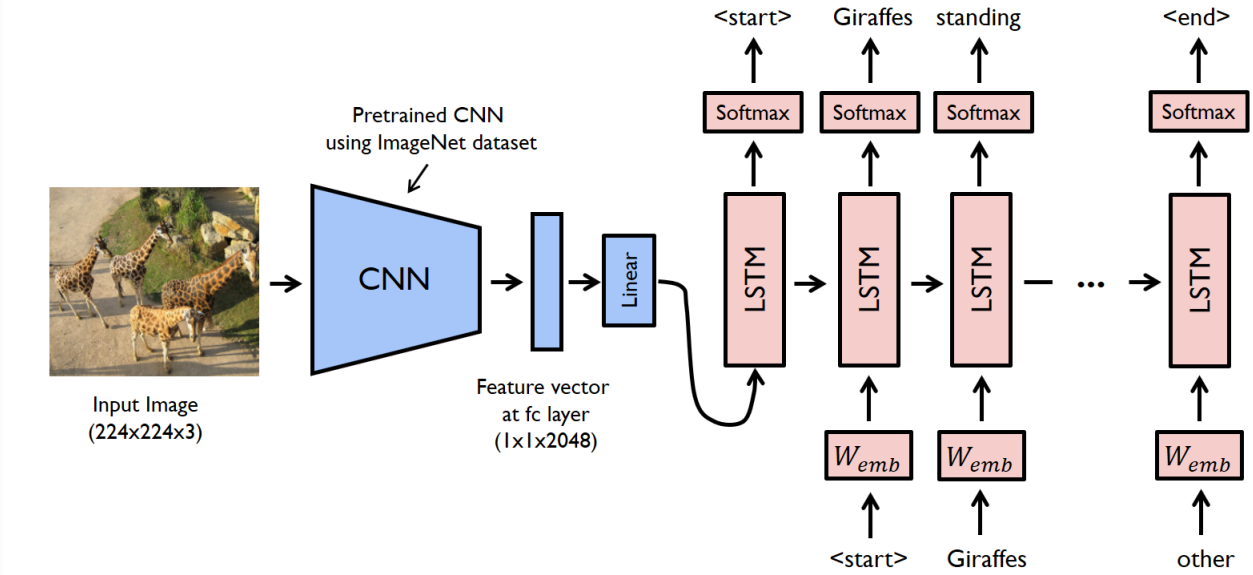


Figure 1

Secondly , what is LSTM ?

LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

we merged this two models in one model called a CNN-RNN model.in general Our approach draws on the success of the top-down image generation models listed above. We use a deep convolutional neural network to extract the visual image features and Semantic features are extracted from the semantic tagging model. Visual features from CNN and semantic features from tagging model are concatenated and feed as the input to a Long-Short-Term Memory (LSTM) network, which then generates captions



3.1 Our Model :-

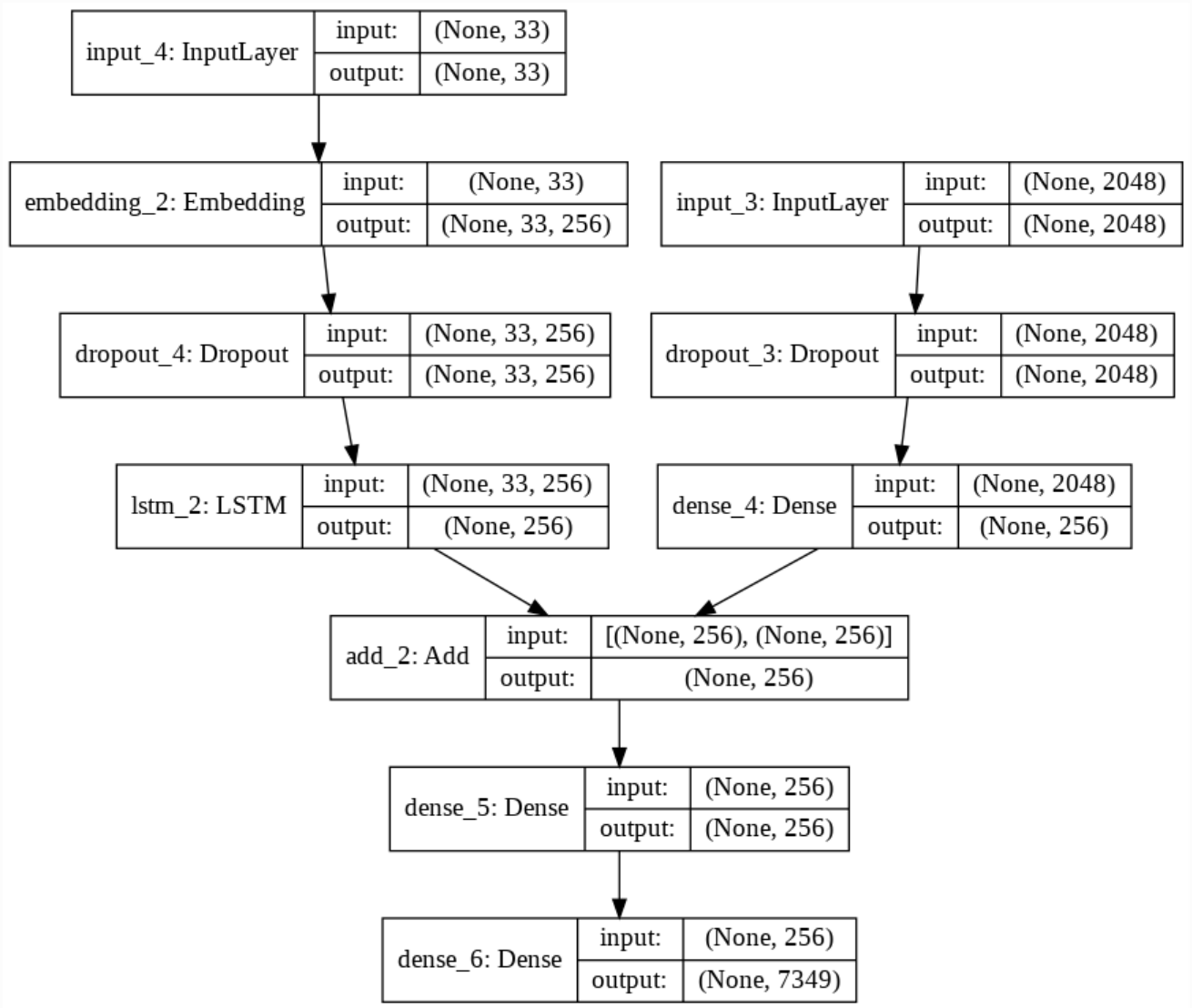
Our model consists of 3 main phases:

A. **Image Feature Extraction:** The features of the images from the Flickr 8K dataset is extracted using the **Xception** model due to the performance of the model in object identification .Becouse it is accurate than VGG16. We will see that later . The **Xception** is a convolutional neural network which consists of consists of 36, as this model configuration learns very fast. These are processed by a Dense layer to produce a 2048 vector element representation of the photo and passed on to the LSTM layer.

B. **Sequence processor** :The function of a sequence processor is for handling the text input by acting as a word embedding layer. The embedded layer consists of rules to extract the required features of the text and consists of a mask to ignore padded values. The network is then connected to a LSTM for the final phase of the image captioning.

C. **Decoder**: The final phase of the model combines the input from the Image extractor phase and the sequence processor phase using an additional operation then fed to a 256 neuron layer and then to a final output Dense layer that produces a softmax prediction of the next word in the caption over the entire vocabulary which was formed from the text data that was processed in the sequence processor phase. The structure of the network to understand the flow of images and text is shown in the Figure 2.

3.1.1 Model Archtecture:



Xception Model Archtecture Figure2

Input_3 : the input here the features vector from the pretrained model (VGG16 - Xception) input shape with VGG16 is 4096 but for Xception is 2048

Input_4 : the input here the Sequence of words (Caption)

Add_2 : Decoder phase

4-Experiments:

4.1. Data :

There was a challenge in choosing the dataset to train our model on it . There was more than one dataset to fulfill our purpose.including (MS-COCO dataset(containing 180k images) , Flickr30k(containing 30k images) , Flickr8k(containing8k images)) , but we choice Flickr8k to train our model on it

. Because we are linked with time and sources of our computers . the dataset divided into 6,000 images for training , 1,000 for validation and 1,000 for testing.

4.2PreProcessing for caption (Description of Image):

Each Image has 5 descriptions(captions) ,The main Function here is Clean() function which takes all descriptions and performs a basic data clean :

1-Removing punctuations

2-Removeing Words that contain numbers

3-Converting all description in lowercase

4-Removing special tokens(like '%', '$', '#', etc.)

,we applied tokenization to our dataset and we used fixed vocabulary size of 8,464

4.3 PreProcessing for images:

we Know that we cant fed image directly to our model for that we should do some preprocessing before feding it to our model:

1- Resize each image to (299 * 299) Xception model or (224 * 224) for VGG16
2- Flatten it
3- Scaling image pixels (normalization)

4.4 Evaluation:

There were many challenges, The first challenge is a difference in choice of convolutional feature extractor. For identical decoder architectures we use VGG16 and Xception ,This two models were trained on Imagenet dataset to perform image classification on 1000 different classes of images. However, our purpose here is not to classify the image but just get fixed-length informative vector for each image. This process is called automatic feature engineering. so we remove the last softmax layer from the tow models and extract a 2048 length vector (bottleneck features) for every image

The second challenge is a single model versus ensemble comparison. While other methods have reported performance boosts by using ensembling, in our results we report a single model performance.

In our evaluation, we compare directly only with results which use the comparable Xception and VGG16 features by BLUE Score

5. Implementation:

The implementation of the model was done using the Python . Keras 2.0 was used to implement the deep learning model. Tensorflow library is installed as a backend for the Keras framework for creating and training deep neural networks. TensorFlow is a deep learning library developed by Google. The neural network was trained on google colab .

Also we used This API :

1- Keras Model API
2- Keras pad_sequences() API
3- Keras Tokenizer API
4- Keras VGG16 API
5- Keras Xception API

6.Results:

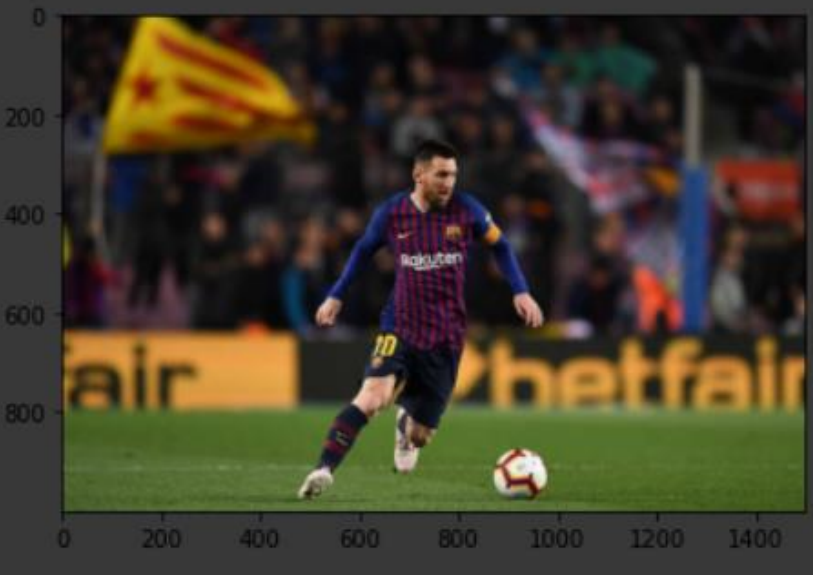
We use **BLUE** score It is an algorithm, which has been used for evaluating the quality of machine translated text. We can use BLEU to check the quality of our generated caption. BLEU is language independent . It lies between [0,1]. Higher the score better the quality of caption

Metric	Xception	VGG16
BLUE-1	0.550179	0.460179
BLUE-2	0.350132	0.320112
BLUE-3	0.150275	0.100245
BLUE-4	0.050874	0.040572

6.1-Test With VGG16

start man in red down and jeans is standing in field of baby end

start man in red down and jeans is standing in field of the ocean end



Test with Xception :

We can see the difference between VGG16 and Xception.

7. Conclusion and Future work:-

In this paper, we have implemented a deep learning approach for the captioning of images. The sequential API of Keras was used with Tensorflow as a backend to implement the

deep learning architecture to achieve a effective BLEU score of 55.01% with Xception Model

we can improve our result by alot of modifications for example:

1. Using a larger dataset.
2. Changing the model architecture, e.g. include an attention module.
3. Doing more hyper parameter tuning (learning rate, batch size, number of layers, number of units, dropout rate, batch normalization etc.).
4. Use the cross validation set to understand overfitting.
5. Using Beam Search instead of Greedy Search during Inference.

8.Referncs :-

- https://www.groundai.com/project/learning-to-evaluate-image-captioning/1
- https://cs231n.github.io/understanding-cnn/
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
- https://arxiv.org/abs/1909.09586
- https://arxiv.org/pdf/1603.05201.pdf
- https://arxiv.org/pdf/1612.07600.pdf
- https://arxiv.org/abs/1703.09137
- https://arxiv.org/abs/1601.03896
- https://www.researchgate.net/publication/321787151_Deep_learning_in_big_data_Analytics_A_comparative_study
- http://proceedings.mlr.press/v37/xuc15.pdf
- https://cs231n.github.io/transfer-learning/
- https://keras.io/api/applications/vgg/#vgg16-function
- https://keras.io/api/applications/xception/
- https://keras.io/api/applications/vgg/#vgg19-function
- https://github.com/hlamba28/Automatic-Image-Captioning/blob/master/Automatic%20Image%20Captioning.ipynb
- http://proceedings.mlr.press/v37/xuc15.pdf