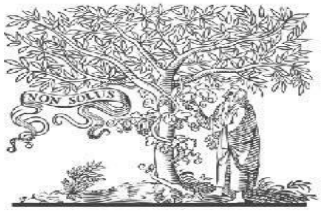
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 05th Apr 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 04

10.48047/IJIEMR/V12/ISSUE 04/11

Title **IMAGE CAPTION GENERATOR**

Volume 12, ISSUE 04, Pages: 76-83

Paper Authors

Dr. O. Aruna, Vutukuri Vijaya Lakshmi, Yadalaparapu Nagendra Babu, Yadlapalli Sri Krishna Teja, Siri Lalitha Adapa

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Image Caption Generator

Dr. O. Aruna[1], Associate Professor, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.

Vutukuri Vijaya Lakshmi[2], **Yadalaparapu Nagendra Babu**[3], **Yadlapalli Sri Krishna Teja**[4], **Siri Lalitha Adapa**[5]
[2,3,4,5] UG Students, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.
arunasri52@gmail.com[1], vijayalakshmiv988@gmail.com[2],
nagendrababuyadalaparapu@gmail.com[3], yadlapalli.krishnateja@gmail.com[4],
sirilalithaadapa@gmail.com[5]

Abstract

When we saw an image automatically our brain recognizes the content/description of an image. But how the system analyses or recognizes an image, is the problem that occurs in most of the scene understanding scenarios. Computer vision (CV) researchers focused on this problem and gave a solution to it by using Deep Learning algorithms. Deep Learning algorithms are the most powerful algorithms which help to make tasks accurately and efficiently without human intervention. In this paper, we are predicting the caption for an image through Deep Learning algorithms called CNN and RNN. A Convolutional Neural Network (CNN) is used to extract the features and objects from an image. These features are then fed into a particular RNN-based model known as a Long Short Term Memory (LSTM), which is in charge of correctly ordering the extracted features and providing an accurate description of the image in a language like English. Deep Learning systems have made picture captioning can be done easily with the help of a dataset. The dataset we used is the Flickr8k dataset which is developed by Shadab Hussain.

Keywords: CNN, Deep Learning, Flickr8k, LSTM, RNN, Xception.

1. Introduction

Creating a textual description of an image's content is a task in computer vision and natural language processing which is also known as image captioning. Image captioning has a goal to develop a model that can automatically generate human-like descriptions of images, allowing computers to better understand and interpret visual content.

Image captioning requires the use of both computer vision and natural language processing techniques. Computer vision algorithms are used to extract visual features from the image, which can then be used to generate a textual description. Natural language processing techniques are used to generate the text description, taking into account the visual features and ensuring that the resulting description is coherent and grammatically correct.

According to the objects seen in an image, the image caption generator automatically generates natural language descriptions. It belongs to the category of scene

understanding, which blends computer vision and natural language processing expertise. From our Literature survey CNN is used to extract the image's characteristics, and STM is used to generate descriptions. Since the existing CNN-based model VGG16 performs less accurately than the Xception model we concluded to use the Xception model for feature extraction and LSTM for generating descriptions of the image.

2. Literature Survey

We researched so many reference papers based on that papers we implemented an Image caption generator by using Xception and LSTM methods. Many researchers in the past have used Inception and vgg16 models. Those models are very slow to train a dataset. These problems are overcome to implemented an Xception model. This model trains a dataset at 70% faster than the previous model.

In[1] This paper presents the model that generates captions to the given image by using a pre-trained machine learning model called VGG16. The Keras backend is to be built by using the TensorFlow library. This is used to train and build deep neural networks. Hence the human-given sentence and generated sentence are very similar.

In[2], the authors proposed ECANN (Extended Convolutional Atom Neural Network) model for Image captioning. They used Freiburg Groceries Dataset and Grocery Store Dataset for captioning images. They both produced 99% accuracy which is greater than other models.

In[3], the authors introduced avtmNet(Adaptive Visual-text Merging network) for accurate captioning of images. To overcome the disadvantages in the existing encoder-decoder model the avtmNet model is highly preferable.

In[4], the authors implemented the R-CNN method to train the network. Due to the decreasing nature of size for R-CNN, accuracy also decreases. To overcome this problem faster R-CNN is used.

In[5], the authors introduced the VGG16 model to train the ImageNet dataset. But it takes a lot of time for training. So, the authors concluded to use the Xception architecture to reduce the time.

In[6], the authors implemented an Inception architecture to train the model. A combination of CNN and LSTM gives an accurate description of an image.

In[7], the authors proposed a combination of top-down and bottom-up mechanisms for extracting salient features of the images.

In[8], the authors proposed Multimodal Recurrent Neural Network to generate captions for an image. Performance is also evaluated by using full-frame and region-level experiments.

In[9], the authors proposed three different methods for image captioning. CNN, RNN and YOLO(You Only Look Once) which identifies objects very efficiently similar to human beings.

In[10], the authors proposed some challenges while building the model. Those are train and performance of the

model. And the number of parameters used for training is also a rich set.

3. Existing Methodology

3.1 VGG16

VGG16 is a pre-trained convolutional neural network. It has 16 layers, in Vgg16 for each image there is one individual vector and in VGG16 more than one million images can be loaded from the database, here pre-trained network can label images into 1000 object groups such as animals and pens. It is used for large-scale image recognition and it has 3x3 convolutions but by using more filters it is trained for 2-3 weeks on 4 GPUS. VGG16 is mainly used to extract image features. The input size for this architecture is 224*224*3. VGG16 uses a Deep Neural Network which is used to extract more information.

3.2 INCEPTION

Convolutional neural networks are the foundation of the deep learning model called Inception. Inception is used for image classification. The inception is made of four parallel layers.

1x1 convolution

3x3 convolution

5x5 convolution

3x3 max pooling

Convolution means which is used to transform an image by applying kernel over every pixel and its neighbors across the complete image. Pooling is the method used to reduce the dimensions. In Inception, the size of the images to be is 299x299x3 pixels size, and it uses the ImageNet dataset for the training process. Inception is used when we have different sizes of an object in the image and it is difficult to detect the perfect information of the object from different images then we need to give an accurate size filter. For instance, consider two images of the same object but there is a difference in the size of the images.

Figure 1: Image of a Car occupying a large region

Figure 2: Image of a Car occupying a small region

Here the object in Figure 1 occupies more region. So it needs higher size filters whereas the object in Figure 2 is less size so it requires lesser size filters. Inception allows internal layers to select which filter size will be appropriate to gain the

required information, even if the object size in the image is different, accordingly the layers will work to recognize the objects.

4. Proposed Methodology

4.1 Xception

In this paper, we implemented CNN-based Xception architecture to extract features from the image. Xception is a learned/pre-trained network that can be trained on an ImageNet database that contains millions of images. This network can classify 1000 image objects easily. Xception architecture contains 71 layers deep and it takes an input image size of 299x299x3 where 3 represents the number of channels. Here it is an RGB channel. This architecture uses a Depthwise Separable Convolution model. It is the extreme version of the Inception model. out of 71 layers, 36 are convolutional layers and those are organized into 14 modules. These Convolutional layers are used to learn features by using 2 spatial dimensions called height and weight and a channel dimension. Hence Convolutional layer has the task of mapping cross-channel correlations and spatial correlations. The input data is mapped into three or four different spaces that are smaller than the original input size using cross-channel correlations. Spatial correlations are used to map these in smaller 3D spaces. This process can be done by 1x1,3x3,5x5 convolutions respectively. In Xception architecture we have depthwise and pointwise convolutions. Depthwise convolution is a channel spatial convolution whereas pointwise is the 1x1 convolution which is used to change the dimension. As shown in Figure 3 in Inception architecture first pointwise convolutions are applied before filters and after the pooling layer which reduces the dimensions and those reduced dimension inputs are further applied to different types of filters.

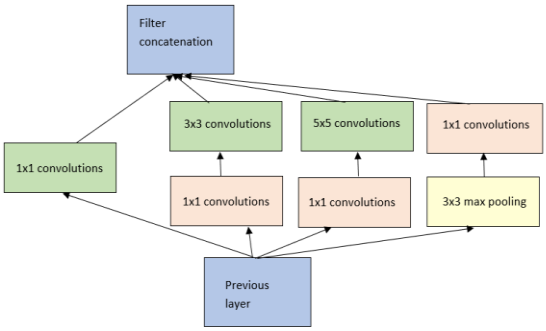


Figure 3: Inception Architecture

But the Xception architecture has a small difference first it applies the filters on the input image and then it reduces the dimensions by using 1x1 convolution as shown in Figure 4.

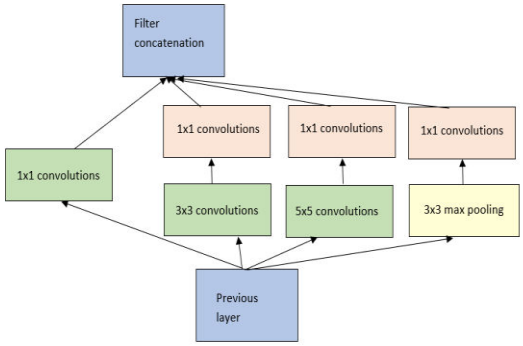


Figure 4: Xception Architecture

4.2 Long Short Term Memory (LSTM)

LSTM is a type of RNN (Recurrent Neural Network) which is used to uncover the underlying relationships in the given sequential data. Since the forgetting and saving mechanism of LSTM makes the structure popular. LSTMs have both Long Term Memory (LTM) and Short Term Memory (STM) to make the task accurate and effective and it uses the concept of gates. LSTM uses 4 gates. Forget Gate, Learn Gate, Use Gate, and Remember Gate. The system design of LSTM is shown in Figure 5.

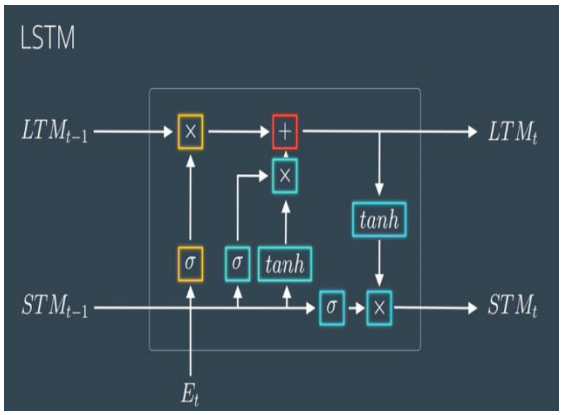


Figure 5: LSTM Architecture

Learn Gate takes an event which is also known as current input and STM as input and those are multiplied with a weight matrix that contains bias and the result is passed to the tanh function which gives the resultant matrix N_t. To keep only relevant information we need to calculate Ignore Factor i_t, for that we need to combine previous Short Term Memory output and event and multiply with the weight matrix and the result is passed to the sigmoid function. After that, both results are multiplied to give the Learn Gate Result.

Long Term Memory is the input for Forget Gate, which determines what knowledge should be retained as essential and what information is unnecessary. To obtain the Forget Factor f_t, the STM_{t-1} and the current event are combined, and they are multiplied by the weight matrix. The result is then put through the sigmoid function to obtain the Forget Gate result.

Remember Gate uses Forget Gate output and Learn Gate output to produce the output by adding them.

To create current event output, we need to utilize Use Gate. U_t is created by biassing the prior LTM before it is applied to the tanh function. In order to create Vt, the previous STM and current event are combined and fed via the sigmoid function. The use gate output which also serves as the STM for the subsequent cell is created by multiplying the two outputs.

5. Implementation

For implementing an Image Caption generator, the prerequisites of our project are:

5.1 Prerequisites

Need to install

1. Python on your computer
2. Keras, Tensorflow, Numpy, tqdm, pillow, cv2, and jupyter notebook through command prompt by using pip command.
3. Download the Flickr8k dataset.

- Keras is a deep learning API developed by Google for implementing neural networks.

- Tensorflow is a tool for numerical calculation that speeds up and simplifies the creation of neural networks and machine learning.
- Pillow is an important tool for dealing with images. It provides various image-processing features that are similar to image-processing software like photoshop.
- Numpy is used for scientific computing like array manipulation and linear algebra operations.
- Tqdm is a library used for creating progress bars/smart bars.

After importing the installed libraries, we need to perform data preprocessing. Data preprocessing involves:

1. Converting all the data into lowercase.
2. Removing punctuations.
3. Removing single letters.
4. Removing all the words which contain numbers.

After cleaning the data we need to extract features from the images by using the Xception model. Further, need to tokenize the vocabulary for an understanding of computers. After creating the data generator and building the model, we need to train and test the model for performance.

6. Results and Discussions

```
Dataset:  6000
Descriptions: train= 6000
Photos: train= 6000
Vocabulary Size: 7577
Description Length:  32
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 32)]	0	[]
input_1 (InputLayer)	[(None, 2048)]	0	[]
embedding (Embedding)	(None, 32, 256)	1939712	['input_2[0][0]']
dropout (Dropout)	(None, 2048)	0	['input_1[0][0]']
dropout_1 (Dropout)	(None, 32, 256)	0	['embedding[0][0]']
dense (Dense)	(None, 256)	524544	['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, 7577)	1947289	['dense_1[0][0]']

```
Total params: 5,002,649
Trainable params: 5,002,649
Non-trainable params: 0
```

Figure 6: Training the model

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Predicted Outputs

start two people are sitting on the shore near the water end

Figure 7: Prediction on a test image

start man is paddling through the water end

Figure 8: Prediction on a test image

start group of people are sitting on the water end

Figure 9: Prediction on a test image

6.1 Comparison of Results:

BLEU-1: 0.337906
BLEU-2: 0.189114
BLEU-3: 0.134397
BLEU-4: 0.059243

Figure 10: Xception Model

BLEU-1: 0.128364
BLEU-2: 0.063430
BLEU-3: 0.041126
BLEU-4: 0.012124

Figure 11: VGG16 Model

As we saw in the figures, Figure 10 and Figure 11, the BLEU score for the Xception model is higher than the VGG16 model.

7. Conclusion

The combination of Long Short-Term Memory (LSTM) and Xception architecture has shown great promise in the task of image captioning. LSTM networks can effectively capture the temporal dependencies in the text descriptions of images, while Xception can extract relevant visual characteristics of the picture.

7.1 Limitations

1. It does not rely on specific concerns and situations.
2. The network adaptability is low.

7.2 Future Scope

The project is to be extended for more developments like improving performance and generating more accurate captions for input images.

References

[1]. Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation in European Conference on Computer Vision.

[2]. Tiwary, T., Mahapatra, R.P. An accurate generation of image captions for blind people using extended convolutional atom neural network.

[3]. Heng Song, Junwu Zhu, Yi Jiang, avtmNet: Adaptive Visual-Text Merging Network for Image Captioning.

[4]. THES, Rohith SriSai, Mukkamala, Rella, Sindhusha, Veeravalli, Sainagesh Object Detection and Identification.

[5]. Jin, J.; Fu, K.; Cui, R.; Sha, F.; Zhang, C. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. arXiv 2015.

[6]. Sreejith S P, Vijayakumar A (2021): Image Captioning Generator using Deep Machine Learning.

[7]. Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, Lei Zhang; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6077-6086

[8]. Andrej Karpathy, Li Fei-Fei; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9]. Pradyuman Tomar, Sameer Haider, Sagar. This work is licensed under a Creative common attribution 4.0 International License paper.

[10]. Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra and Nand Kumar Bansode.