Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Image Caption Generation using Convolutional Neural Network and Long Short Term Memory

Afeefa Nazneen N Z and Dr. Shreedhara K S
Department of Computer Science and Engineering, UBDTCE, Davangere, Karnataka, India
E-mail: afeefanaz05@gmail.com

ABSTRACT

Image captioning is a subject that has steadily gained the interest of many scholars working in the field of Artificial Intelligence (AI). Picture captioning, which combines the abilities of Computer Vision (CV) and Natural Language Processing (NLP), generates natural language descriptions based on the content of a picture or image. It is an essential part of scene interpretation. While a computer can't tell what an image is about, the human brain can. Thisis resolved by developing a model using Deep Learning (DL) techniques, vast datasets, and computer power. In this paper,Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) techniques are combined to implement the proposed project. Since the proposed model is data-dependent, the generated captions for photographs will only contain words from the proposed model's vocabulary.

Keywords: Image Caption, Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Deep Learning (DL), Recurrent Neural Network (RNN).

1. INTRODUCTION

In this paper, the goal of the Image Caption Generation job is to understand an image's context and represent it in a English language by utilising Natural Language Processing (NLP) and Computer Vision (CV) concepts. The Image Caption is the text that appears next to an image and identifies or characterises the image.

The Bottom-Up method and the Top-Down method are both used to describe images. A bottom-up technique generates contents from an input image, which are then merged to make a caption. The top-down approach, which employs architectures such as Recurrent Neural Networks (RNN), provides a semantic representation of an input image before decoding it into a caption **[1]**.

For example, caption creation can be useful for captioning scenes for visually impaired persons, classifying films and images set up in various circumstances for more effective outcomes, responding to visual queries, and understanding context **[4]**.

Convolutional Neural Networks (CNNs), which are specific critical neural systems capable of creating information with an information shape and these are the most extensively used models used today **[2]**. During image processing, CNN is an excellent approach for extracting information from images.

On the other hand, a neural network is simply a circuit of biological neurons or a network of neurons/nodes. Convolutional refers to the mathematical combination of two functions to create a third. This model employs a multilayer perception version and is composed of one or more convolutional layers that can act as connective tissue or be entirely pooled.

Furthermore, these layers provide feature maps, which allow for the recording of an image area that will be separated into rectangles and transmitted for extra nonlinear processing.

The following are the primary benefits of using CNN **[3]**:

- In image recognition problems, the accuracy will be very high.
- It can recognise important features without human intervention.

CNN's drawbacks will be as follows:

- The amount of training data required will be more.
- Object position and orientation cannot be encoded.

Long Short Term Memory (LSTM), a type of Recurrent Neural Network (RNN), provides output from the prior step as an input into the current state. The LSTM assists in the creation of a visual description. In RNN, recurrent means reoccurring regularly and each node in the RNN architecture functions as a memory cell.

The key advantages of LSTM are:

- Handwriting Generation
- Question and Answering Chatbot
- Machine Translation
- Image Captioning etc

1.1 Problem Statement

Every day, we are surrounded with images, whether they are in magazines, newspapers, or on social media. Humans are capable of identifying and accurately describing photographs, while machines require image training before they can write captions for images. In this paper, based on the information provided above, the proposed project is designed to generate captions for photos.

1.2 Objectives

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

- In drive, preprocessing the data that is stored or uploaded in the newly created folder.
- Image descriptions will be useful for persons who are blind in interpreting the content of photographs on the internet.
- Create a functioning Image Caption Generator model by combining CNN and LSTM.
- Extracting image features from Xception model and training the model.

2. LITERATURE REVIEW AND SYSTEM REQUIREMENTS

2.1 Review of Literature on VGG16 and Xception Models

Ali Ashraf Mohamed **[1]** highlighted the experimental findings in this work utilising the MSCOCO dataset. In their proposed work, they have introduced a feature called guiding network for the encoder/decoder design. The guiding network approach was detailed in this research, which largely deals with learning the vector via a neural network, i.e. v = g (A), where A is the set of annotation vectors. This study addressed the essential topic of developing natural language descriptions from visual input, which has long been investigated in computer vision. As a result, complex systems that integrate visual basic recognizers with a codified language such as and-or graphs or logic systems have emerged.

2.2 Review of Literature on CNN and LSTM

Swarnim Tripathi and Ravi Sharma **[2]** used CNN and LSTM to learn the caption for the image in their study. An image caption generation system employs CV and NLP criteria to comprehend the image's link in English. In this research, they cautiously follow a number of critical photographic captioning ideas and well-known methodologies. They describe the Keras library, NumPy, and Jupyter notebooks as well as the Flickr dataset and CNN used for image categorization in this study.

Chetan Amritka and Vaishali Jabade **[3]** present a model that can generate original descriptions from photographs in this study. For this task, they used the Flickr_8k dataset, which has 8000 pictures and five descriptors per image. In this study, CNNand RNN are both used. To categorise photos, a pre-trained Convolutional Neural Network is employed. This network functions as an image encoder. The model's final hidden layer is fed into the Recurrent Neural Network (RNN) which is a sentence decoder. When compared to the content of the original image, the caption's sequence or sentence looks to stray off track or forecasts the incorrect text.

2.3 Review of Literature on Attention Based DeepLearning Model

H.M. Abdullah Zia, Inaam Ilahi and others **[4]** recent advances in DL have created various chances to solve issues that have plagued real-world applications for more than tenyears. In this paper they implemented the automatic caption generation in Urdu.

Despite the fact that Urdu is Pakistan's official language and is widely spoken and understood in the Pakistan-Indian subcontinent, no activity has been done to make captions in Urdu. There research intends to close this gap by creating an attention-based DL model that is tailored to the Urdu language using sequence modeling approaches. They created an Urdu dataset by translating 700 "man" photos from the "Flickr8k" dataset. Using this dataset to test their proposed method, they demonstrate that it can generate an Urdu BLEU score of 0.83. They also discuss how the generated captions can be improved in terms of grammar.

2.4 System Requirements

Based on the literature review, in this paper the proposed project is being built with the following hardware and software specifications.

Hardware Requirements

- Processor: Pentium IV or above
- Disk: 2 Giga Bytes (minimum)
- Random Access Memory: 1 Giga Bytes (minimum)

Software Specifications

- Operating System: Windows XP or later version
- Platform: Jupyter Notebook or Google Colab
- Language: Python

3. SYSTEM METHODOLOGY

3.1 AI, ML and DL in Brief

Artificial Intelligence (AI) applications such as Machine Learning (ML) use algorithms to read or analyse data, learn from that data and then use what they have learnt to make judgments. The ability of machines (i.e. computers or ideally computer programmes) to learn from previous behavior or data and predict events without being expressly programmed to do so is known as Machine Learning **[5]**.

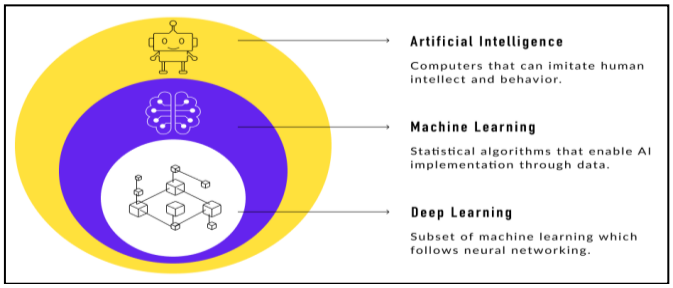


Fig. 3.1 Difference between AI, ML and DL

DL is a subset of ML; it uses layers of structure to construct Artificial Neural Networks (ANNs), which are capable of learning and making sensible decisions on their own **[5]**. There are numerous neurons or perceptrons in each layer of an ANN.

Deep Learning is used to analyse larger datasets and delivers conclusions with high precision resulting in high output performance. DL is used in image description applications. The process of describing the information contained in a picture is known as image description.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

In DL, nothing is explicitly programmed. In essence, DL is an ML class that uses a large number of nonlinear processing units to extract and transform features. DL comes into play when there are lot of inputs and outputs. Deep Learning is implemented via neural networks. The biological neuron,which is essentially a brain cell, acts as inspiration for these neural networks **[14]**. Deep learning is implemented using deep networks, which are basically neural networks with several hidden layers.

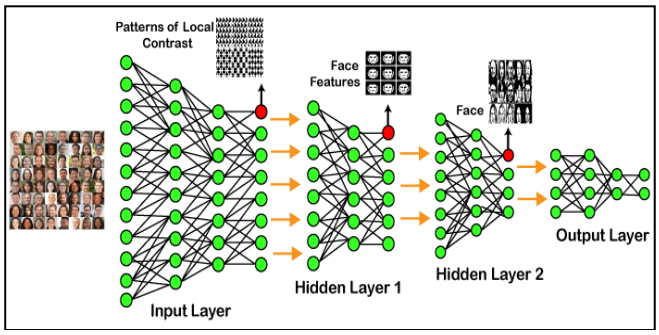


Fig. 3.2 Deep Learning Example

In the example shown in Fig. 3.2, the raw image data are sent into the first layer of the input layer, which then determines the patterns of local contrast by discriminating based on colors, luminance and so on. The final hidden layer will then choose facial features focusing on the (i) eyes (ii) nose (iii) lips and so on.

The facial traits will subsequently be fixed on the appropriateface template. Since the correct face can be seen in the above image (Fig. 3.2), it will be determined in the second hidden layer before being transmitted to the output layer. More hidden layers can be developed in a similar manner to address complex situations **[14]**.

3.2 Architectures used in the Proposed Project

In this paper, the CNN and LSTM architectures are usedin order to implement the proposed model, where LSTM is an RNN. The image's characteristics are extracted using CNN. LSTM is used to create a description of an image using CNN data. These two architectures are discussed below.

3.3 Generic CNN Architecture

Artificial deep learning neural networks are called as Convolutional Neural Networks (CNNs) used for image categorization, computer vision, image identification and object detection. CNN picture classifications take an input picture, evaluate it and then group it into subcategories such as "dog" or "cat" among others. CNN classifies images using an algorithm that scans them from top to bottom and left to right in order to extract key features **[1]**.

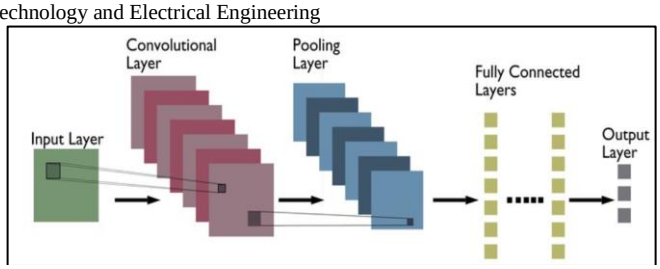


Fig. 3.3 Generic CNN Architecture

CNNs are deep neural networks that can handle data with input shapes comparable to a 2D or two-dimensional matrix. A 2D matrix is a simple way to represent images and convolutional neural networks are powerful image processing tools. Typically, CNN is used to classify photographs deciding if they show among other things a bird or Superman **[6]**.

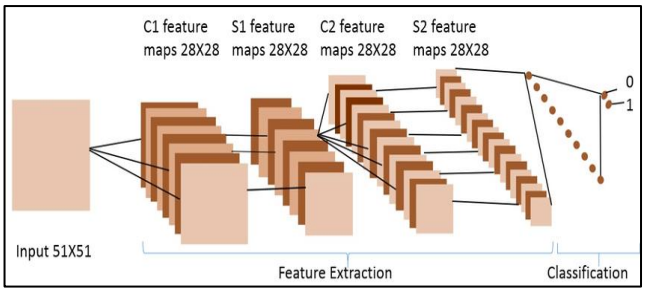


Fig. 3.4 Feature Extraction and Classification

Feature Extraction is the technique of extracting or taking out numerical features from raw data. The feature extraction includes many convolutional layers, pooling (max/min/average) and activation algorithms such as relu, softmax and others. Classification involves grouping objects into categories. Figure 3.4 depicts the feature extraction and classification.

Neurons in the convolutional layer collect information from the previous layer's set of pixels. The purpose of pooling layer is to decrease the input image size aiming to decrease the load of computation, memory requirements and the number of parameters that allow for overfitting. Accuracy is slightly lower while using pooling strategy than non pooling strategy but the execution time is faster in pooling. Even though the accuracy is high in without pooling but the execution time will increase. Thus, for faster execution we are using the polling strategy in our proposed project or model.

3.4 LSTM Architecture

LSTM, a type of RNN which is useful for solving the sequence forecast **[1]**. By overcoming the short term memory restrictions of classical RNN, Long Short Term Memory has proven to be more successful. The LSTM cell is as shown in Fig. 3.5.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

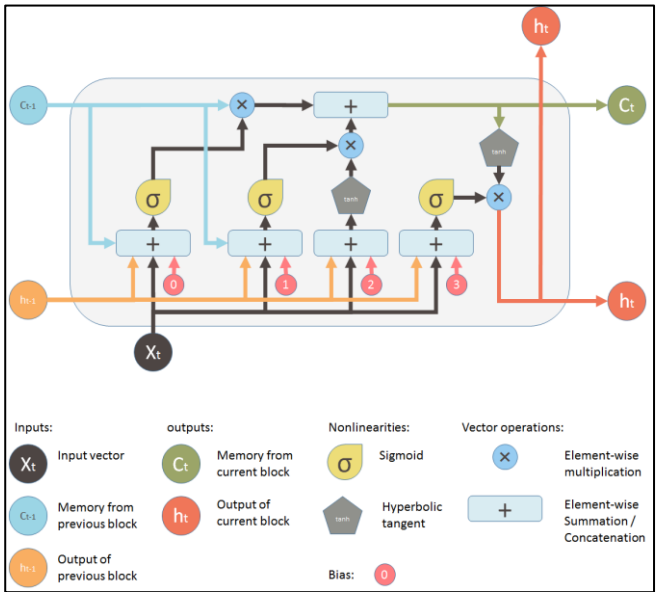


Fig. 3.5 LSTM Architecture/LSTM Cell Structure

LSTM was invented by Hochreiter and Schmidhuber. When processing the inputs, Long Short Term Memory can save useful data while discarding unnecessary data via a forget gate. In a recurrent neural network, the output from the previous phase serves as an input for the current phase.

LSTM was created to solve the issue of RNNs long-term dependence which occurs when RNN cannot foretell or predict words stored in long-term memory but based on recent input, it is competent to do so. Recurrent Neural Network does not perform effectively as the gap duration rises. Long Short Term Memory has the ability to keep the knowledge for a long time by default. The three applications of LSTM are processing, prediction and using time series data to categorize.

The LSTM design is quite straightforward; it consists of just 3 primary gates which hold the data for a longer time and assist in resolving issues that Recurrent Neural Networks (RNNs) were unable to resolve. The three primary gates of the LSTM covers are **[13]**: Forget Gate, Input Gate and Output Gate.

3.5 Working of CNN

In CNN, various layers are there such as (a) Convolution Layer (b) Stride Layer (c) Padding Layer and (d) Pooling Layer.

a) Convolution Layers

- Convolution Layers are the first layers used to extract features from a picture. Convolution Layers preserve the link between pixels by using a brief input data sequence to learn features. A Convolution Layer in mathematics is made up of three components: a kernel or filter, an image matrix, and two inputs **[11]**. The computed result is as shown in Fig. 3.6.

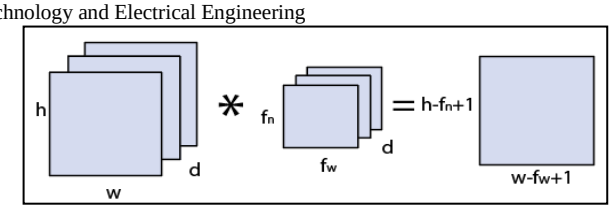


Fig. 3.6 Kernel or Filter Matrix for Image Matrix Multiplication

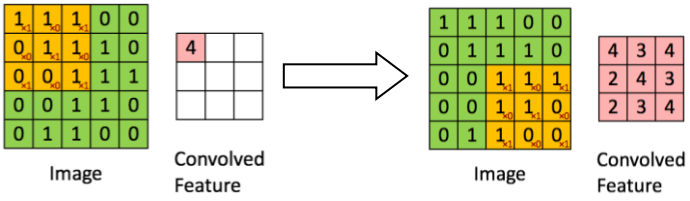
- In the Fig. 3.6;
  - The image matrix or the window is h × w × d.
  - The dimensions of the filter are $f_h \times f_w \times d$ and it may be either vertical or horizontal.
  - The output is calculated as $(h - f_h + 1)(w - f_w + 1) \times 1$ and this will be the feature map.

- Convolutional Layer is illustrated by solving an example witha 5x5 image matrix, a 3x3 filter matrix and pixel values of 0 and 1, as shown below.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$5 \times 5 -$ Image Matrix $\quad 3 \times 3 -$ Filter Matrix

- The matrix multiplication will function as follows; the first and last steps are given below.



- The final Convolution Layers output matrix of a 5x5 picture multiplied by a 3x3 filter will look like this and the feature created is known as a feature map or convolved feature.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Convolved Feature

b) Strides

- The Stride is the transition of vision from one neuron to another neuron. When the number of Strides is 1, the filter is moved to 1 pixel at a time and the pixels are moved from the output matrix to the input matrix when the array is constructed. The filters are also changed to 2 pixels when the number of Strides is 2 and so forth.
- Strides are essential because they control the convolution of the filter against the input i.e. Strides are responsible for regulating or for adjusting the features that could be missed while flattening the image. Strides denote the number of steps we are moving in each convolution. The Fig. 3.7 shows how the convolution would work.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

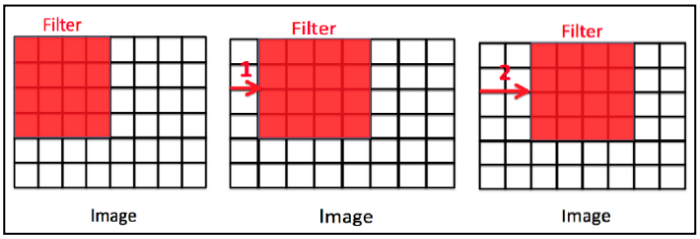


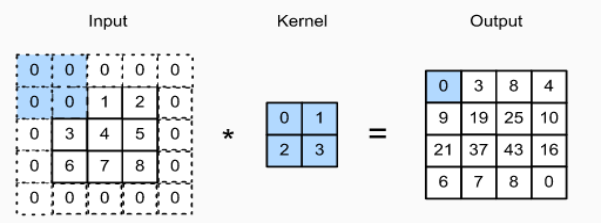
Fig. 3.7 Working of Convolution

- Stride = 0 in the first image or matrix of Fig. 3.7, Stride = 1 in the second image and Stride = 2 in the third image. The formula in equation (3.1) is used to compute the size of the output image.

$$\left[\left\{\frac{(n+2p-f+1)}{s}\right\}+1\right]\left[\left\{\frac{(n+2p-f+1)}{s}\right\}\right] \ldots\ldots\ldots\ldots\ldots\ldots \text{(3.1)}$$

c) Padding

- The Padding plays a vital role in creating Convolutional Neural Network. After the convolution technique, the image is smaller than it was before. In image classification task, after every step of multiple convolution layers, the original image is shrunk.
- Secondly, overlapping occurs when the kernel moves over the original image, it passes through the middle layer more times than that of the edge layers.
- To overcome this overlapping problem, a new concept was introduced named Padding. An additional layer known as Padding can increase an image's boundaries while maintaining or without reducing the size of the underlying image. An example is shown below:



- When an f matrix with padding p and an nxn matrix are combined, in this case equation (3.2) will display the size of the resultant picture.

$$(n+2p-f+1)\times(n+2p-f+1) \ldots\ldots\ldots \text{(3.2)}$$

d) Pooling

- Another component of a CNN is the Pooling layer, which is crucial in the pre-processing of an image.
- If an image is too huge, the pre-processing stage reduces the amount of parameters to make it smaller. The pixel density is reduced when the picture is shrunk and the downscaled image is obtained from the previous layers.
- In essence, Polling's objective is to gradually lower the spatial size of the image to lessen the computational burden and network complexity. Down sampling or sub sampling are other names for spatial pooling, which reduce the dimensionality of each map while keeping the key properties.
- Each value in the feature map is subjected to a ReLU. Rectified linear unit is a straightforward and useful nonlinearity that is present but it has no effect on the feature map's values because Pooling layers are added later.
- Pooling is introduced after applying the nonlinearity to the feature maps or the convolved features.

3.6 Working of Long Short Term Memory

At a high level, Long Short Term Memory works very much like a Recurrent Neural Network cell. It is detailed how the LSTM network operates internally. As seen in Fig. 3.8, the LSTM is composed of 3 sections, each of which serves a specific purpose.

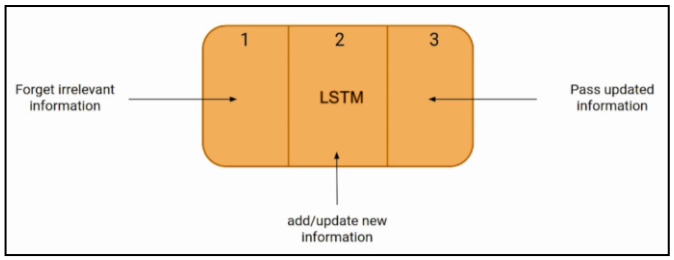


Fig. 3.8 Three Parts of LSTM

The LSTM's "Forget Irrelevant Information" part decides if the data from the preceding timestamp should be maintained i.e. do not forget or if it is irrelevant and can be forgotten. In the second half of the LSTM, the "Add or Update New Information" cell strives to obtain new information from the data entering this cell. In the third segment of the LSTM, the "Pass Updated Information" cell communicates the updated data between the current and subsequent timestamps.

In Fig. 3.9, the three components of a Long Short Term Memory cell are referred to as gates. The first, second and third components are known as the Forget Gate, Input Gate and Output Gate respectively.

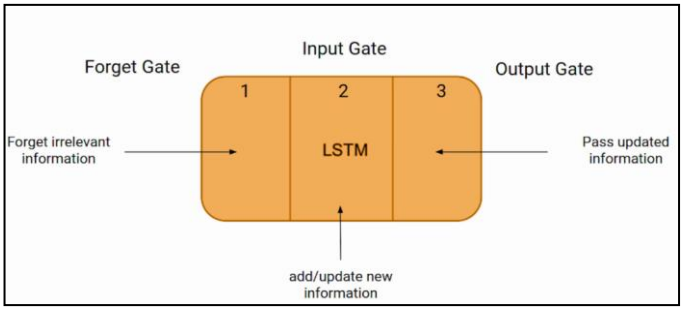


Fig. 3.9 Three Gates of LSTM

Just like a basic RNN, a LSTM is also having a hidden state, where H_{t-1} denotes the prior timestamp's concealed state or hidden state and H_t denotes the present timestamp's concealed state or hidden state.

Furthermore, the LSTM cell state is represented by the timestamp's C_{t-1} for the past and C_t for the present. In this instance Long Term Memory is the cell state while Short Term Memory is the concealed state or hidden state as seen in Fig. 3.10.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

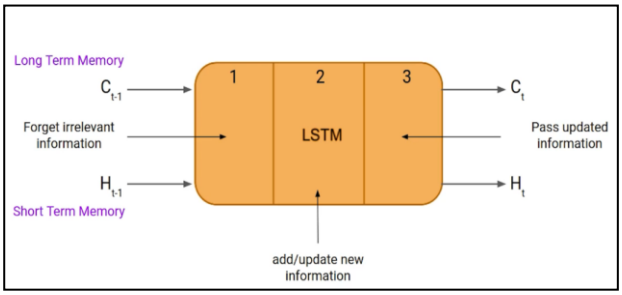


Fig. 3.10 States of LSTM

The fact that all of the timestamps are stored with the information in the cell state or long term memory is interesting to notice.

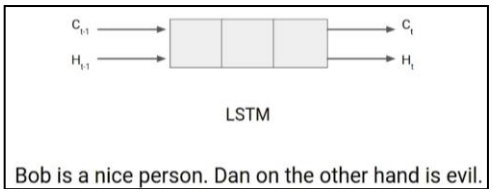


Fig. 3.11 Example of LSTM

As seen in Fig. 3.11, an example is used to demonstrate how the LSTM functions. In this instance, a full stop separates two sentences. In the example, the first sentence reads, "Bob is a good person" whereas the second, "Dan on the other hand is evil." It is obvious that the topic in the first phrase is about Bob and that the discussion switched to Dan as soon as we reached the full stop (.).

Network should understand that we are no longer talking about Bob when we switch from 1st to 2nd phrase and we are instead discussing Dan. As a result, the LSTM network's Forget Gate enables it to forget about it.

3.7 Models

a) Model for an Generating Image Caption

- CNN and LSTM architectures are combined to develop model for an Image Caption Generator, commonly known as a CNN-RNN model.
- From the image in order to extract features, the pre-trained Xception model and Convolutional Neural Network (CNN) are used.
- CNN data will be utilized by Long Short Term Memory to provide a picture description.
- In this paper the proposed project involves combining these two models into a single model known as a CNN-RNN model.
- While the semantic characteristics are extracted using a semantic tagging model, the visual picture features are retrieved using a Deep Convolutional Neural Network.
- The caption is produced by an LSTM network using the concatenated visual characteristics from CNN and semantic information from the tagging model **[1]**.
- The Image Caption Generator Model is as shown in Fig. 3.12.

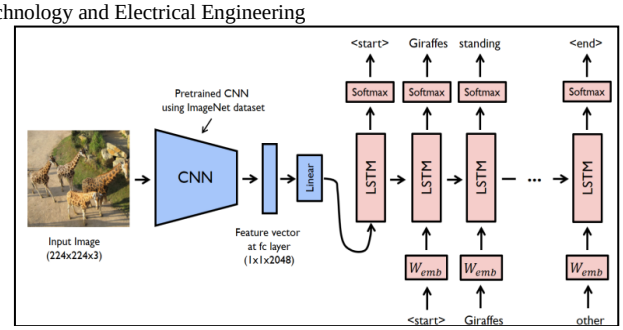


Fig. 3.12 Image Caption Generator Model

b) Xception Model

The Xception model is divided into three phases:

- **Image Feature Extraction:** Since the Xception model performs best when used to recognize objects, thus it is used to extract Image attributes from the Flickr8K dataset. Xception model is employed since it is more accurate than VGG16. The Convolutional Neural Network (CNN) used by the Xception has a very quick learning curve. These are then sent to the LSTM layer, which processes them into a 2048 vector element image representation.

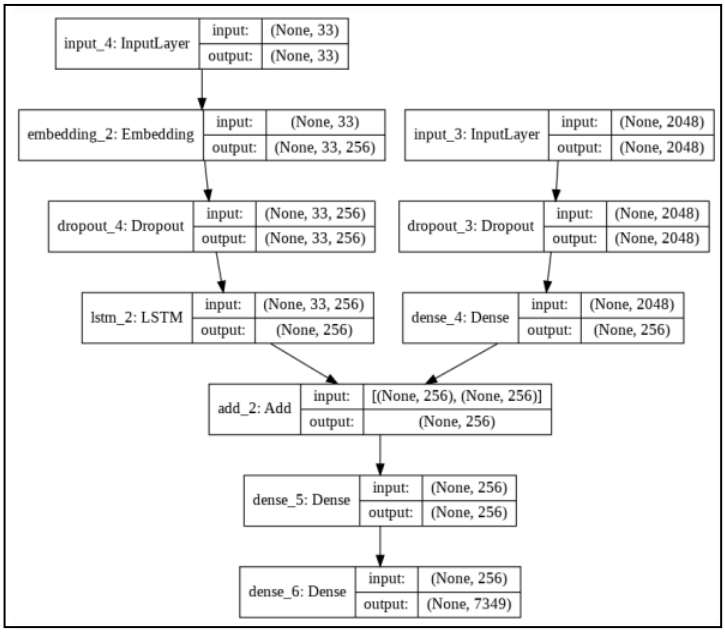


Fig. 3.13 Xception Model

- **Sequence Processor:** A Sequence Processor manages the input text by serving as a word embedding layer. The embedded layer contains rules for extracting the required text attributes or features from the picture and comprises of mask to reject or ignore padding values. The network is linked to a LSTM for the last stage of captioning images.

- **Decoder:** Using an additional operation, the model's last step merges the input from the Image Extractor phase and Sequence Processor phase. This input is then sent to a 256 neuron layer, which is then fed into a final output dense layer, which generates a softmax prediction of the next word i.e. at the output layer in

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

the caption over the whole vocabulary. The text data processed in the 2nd phase (i.e. in Sequence Processor phase) yields the next word. The organizational layout of the network for analyzing the text and picture flow is depicted in Fig. 3.13.

3.8 Project Dataset and File Structure

a) Project Dataset

- Flickr8K dataset is used to generate image captions. A smaller dataset, Flickr8k is employed in the proposed project instead of larger datasets like Flickr30K and MSCOCO, which require weeks to train the network.
- Large datasets are used to build better models.
- Knowledge of DL, Python, Keras Library, Jupyter Notebooks, NumPy and Natural Language Processing are all essential for the proposed project.
- For functioning, the following libraries must be installed: install TensorFlow, Keras, Tqdm, Pillow and NumPy using pip.

b) Project File Structure

The dataset's following folders are downloaded:

- **Flicker8k_Dataset:** A folder containing the 8091 pictures that makes up the dataset.
- **Flickr8k_text:** This text folder contains text files as well as image captions.

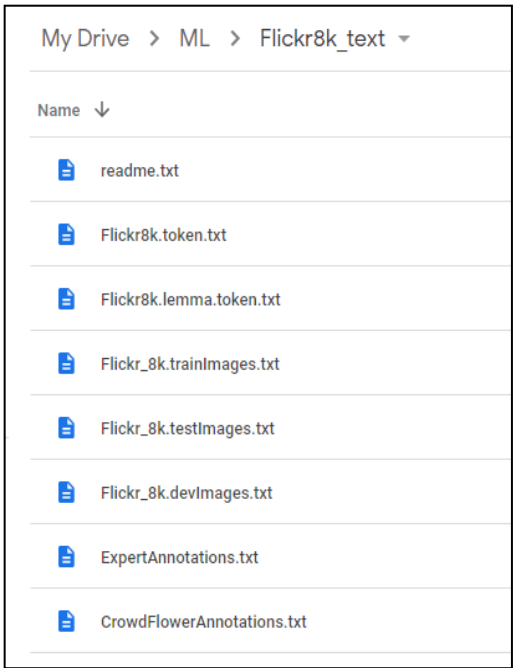


Fig. 3.14 Flickr8k_text Dataset Folder

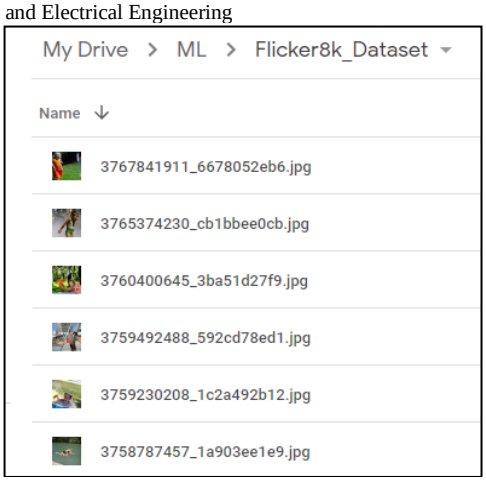


Fig. 3.15 Images in Flicker8k_Dataset Folder

- While making the project the files created by us are as shown in Fig. 3.16.

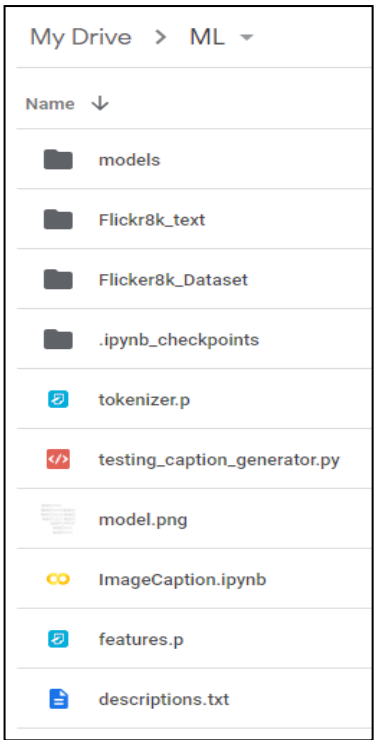


Fig. 3.16 Files Created while Making the Project

4. IMPLEMENTATION

Implementing this project is a remarkable achievement in our present system so it is observed that many of the utilities to implement this one, standard APIs are available.

Before implementing this proposed project, following are the activities and very important requirements;

- **Dataset**: Flicker8k_Dataset is used and is downloaded from the Kaggle or GitHub.
- **Text File**: Flickr8k_text file is also downloaded from Kaggle or GitHub.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

First, download and extract the required files and then upload the file (testing_caption_generator.py) and folders (Flickr8k_text, Flicker8k_Dataset) in ML or any newly created folder in drive. Create an ImageCaption.ipynb notebook in Google Colab where the code is written starting from mounting the drive to training the model. This is depicted in Fig. 4.1.

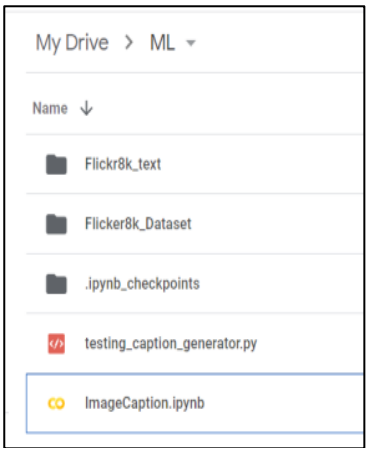


Fig. 4.1 ML Folder Containing Python File, Text and Dataset Folders

1. Mounting Drive

- Mounting the drive allows us to access or read files and folders or directories from our Google Drive.

2. Installing Necessary Libraries

- The required libraries are installed with the pip command, as shown in Fig. 4.2.
- The libraries are:
  - **TensorFlow** is an open source platform which is used to develop and train Machine Learning (ML) models.
  - **Keras** is an API which is used for faster experimentation.
  - **Pillow** is a library which is used to save different types of image file formats.
  - **NumPy** is a library which is used for working with the arrays.
  - **Tqdm** is a library which is used for creating progress bars.

```
!pip install tensorflow keras pillow numpy tqdm
```

Fig. 4.2 Installing Necessary Libraries

3. Import all the Necessary Libraries

- Using the import function, import the relevant libraries.

4. Gathering and Cleaning of Data

- All of the captions for the images in Fig. 4.3 may be found in the primary text file called Flickr8k.token.txt, which is located in the Flickr8k_text folder.
- As shown in Fig. 4.3, there are five captions for each image, each of which is given a number between 0 and 5.

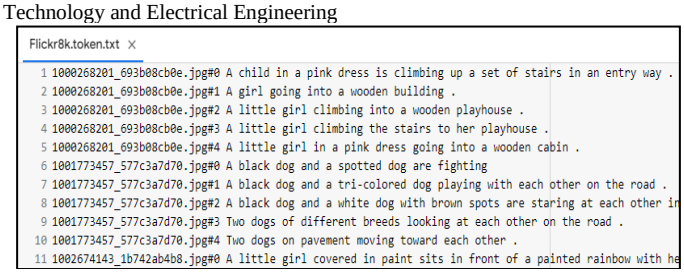


Fig. 4.3 Flickr8k Token File Containing all Image Captions

- All descriptions are taken into account while cleaning the data with the **cleaning_text (descriptions)** function. This is a crucial step when working with text data. The type of cleaning conducted on the text is determined by the purpose. In our example or case, punctuations are deleted, all text is converted to lowercase and digits are eliminated from words. With coding, a caption such as “A man mounting a horse” will be turned into "man mounting horse"
- The function **save_descriptions (descriptions, filename)** compiles a list of all the preprocessed descriptions and stores them in a file. All of the captions are saved in a file called **descriptions.txt**. It will resemble the scene in Fig. 4.4 in several ways.

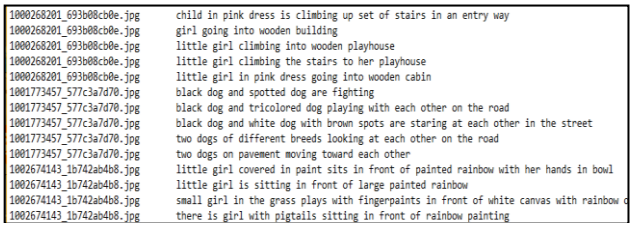


Fig. 4.4 Preprocessed Descriptions

- The fundamental **text_vocabulary (descriptions)** function collects all the special terms and builds a vocabulary from all the descriptions.

5. Extracting Characteristic Vectors from all the Pictures

- Extracting characteristic vectors from all the pictures is also referred as Transfer Learning. When performing tasks, features are extracted from the pre-trained model i.e. from Xception model that has already been trained on huge datasets. The imagenet dataset with 1K (1 Thousand) different classifications is used to train an Xception model.
- Xception model is directly imported from the keras.applications; however the key is to make sure the internet is accessible because the weights are downloaded automatically.

- The function **extract_features (directory)** is used for extracting features from all photos and to map picture names to the appropriate feature array. Further, the features dictionary is copied into the pickle file called **"features.p"**.

6. Loading Dataset for Model’s Training

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

- For training, a file called **Flickr_8k.trainImages.txt** is used this file can be found in the Flickr8k_text folder and has a list of 6K (6 Thousand) picture names.
- In order to load the training dataset additional functions are required, including:
  - The function called **load_photos (filename)** is used to read the text file as a string and return a list of picture names.
  - The function **load_clean_descriptions (filename, photos)** is used to extract captions for each image from the list of images and store them in a dictionary. For each caption, the "start" and the "end" identifier is appended. This is required by the LSTM model in order to recognize the beginning and end of the caption.
- The function **load_features (photos)** is used to obtain the dictionary for picture names and their characteristic vectors which were previously taken from the Xception model.

7. Vocabulary Tokenization

- Since computers cannot understand English words, they must be represented by numbers. Thus, each word in the lexicon is mapped using a different index value.
- Tokens from the specified vocabulary will be generated by the Keras library's Tokenizer function and saved in the pickle file **"tokenizer.p"**.
- The tokenizer.p file is automatically added to the folder after running the appropriate code. The proposed project's suggested vocabulary contains 7577 terms.

8. Creating Data Generator

- In order to convert the task into supervised learning, the input and output are fed into the model during training. Each of the 6000 photographs used to train the model will have a 2048-length feature vector as well as a numerical representation of the caption. Since it is hard to store the data from 6000 photographs in memory, a generator approach is used to generate batches.
- Thinking about a scenario where [X1, X2] as the model's input and Y as the output. In this example, X1 represents the image's 2048 feature vector, X2 represents the input text string and Y represents the text string that the model needs to foretell and is depicted in Fig. 4.5.

x1(feature vector)	x2(Text sequence)	y(word to predict)
feature	start,	two
feature	start, two	dogs
feature	start, two, dogs	drink
feature	start, two, dogs, drink	water
feature	start, two, dogs, drink, water	end

Fig. 4.5 Word Prediction

9. CNN-RNN Model: A Definition

- The Keras Model of the Functional API is used to specify the structure of the model.
- The model will consist of three major components:
  - **Feature Extractor:** From the photo, the 2048-by-2048 pixel dense layer of the feature was extracted or image will be decreased (shrunk) to 256 nodes.
  - **Sequence Processor:** First the textual input is handled by an embedding layer and then it is passed to the LSTM layer.
  - **Decoder:** In order to make the final prediction, primarily the outputs from the first two layers are merged and then we will process by the dense layer. The final layer will have the same number of nodes as the lexicon.
- The model is saved in the folder as a png file.

10. The Model's Training

- To create the batches of input sequence and output sequence and fitting those sequences to the model, the function **model.fit_generator ( )** is used, 6000 training pictures are utilized to train the model. Additionally, the model is saved to the models folder which depends on our system's capacity.

```
None
<keras.engine.functional.Functional object at 0x7f89313f5d50> model
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
  app.launch_new_instance()
6000/6000 [==============================] - 1496s 249ms/step - loss: 4.5191
6000/6000 [==============================] - 1483s 247ms/step - loss: 3.6714
6000/6000 [==============================] - 1474s 246ms/step - loss: 3.3861
6000/6000 [==============================] - 1479s 247ms/step - loss: 3.2153
6000/6000 [==============================] - 1484s 247ms/step - loss: 3.0982
6000/6000 [==============================] - 1493s 249ms/step - loss: 3.0098
6000/6000 [==============================] - 1497s 249ms/step - loss: 2.9395
6000/6000 [==============================] - 1497s 249ms/step - loss: 2.8870
6000/6000 [==============================] - 1484s 247ms/step - loss: 2.8411
6000/6000 [==============================] - 1489s 248ms/step - loss: 2.8052
```

Fig. 4.6 Training the Model for 10 Epochs

11. Model Testing

- Following the training of the model, a different file called **"testing_caption_generator.py"** is used to activate the model and then produce forecasts. The same **tokenizer.p** pickle file is utilised in this situation to extract the words from their index values because the predictions have an index value that is the longest possible.

5. RESULTS AND DISCUSSIONS

5.1 Image 1 Input and Output

Input with Code

```
from PIL import Image
img = Image.open('/content/drive/MyDrive/ML/Flicker8k_Dataset/1007320043_627395c3d8.jpg')
img
```

Output with Code

```
/e/ML/testing_caption_generator.py' -i '/content/drive/MyDrive/ML/Flicker8k_Dataset/1007320043_627395c3d8.jpg'

2022-06-22 08:13:16.589164: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERRO
```

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X


```
start young boy is swinging on swing end
```

Complete Input and Output in One Image

Fig. 5.1 The above snapshot displays the input and output i.e. **image and the generated caption for second image** respectively. The caption generated for second image is **"start young boy is swinging on swing end"**

5.2 Image 2 Input and Output

Fig. 5.2 The above snapshot displays the input and output i.e. **image and the generated caption for third image** respectively. The caption generated for third image is **"start dog is running through the grass end"**

5.3 Image 3 Input and Output

Fig. 5.3 The above snapshot displays the input and output i.e. **image and the generated caption for fourth image** respectively. The caption generated for fourth image is **"start two people are sitting on bench in front of crowd end"**

5.4 Image 4 Input and Output

Fig. 5.4 The above snapshot displays the input and output i.e. **image and the generated caption for fifth image** respectively. The caption generated for fifth image is **"start two people are walking through the snow end"**

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this paper, we discussed CNN-RNN model which is used to generate caption for image. For implementation purpose, Google Colaboratory is used which provides the cloud service with free GPU. The important thing to remember is that, since the suggested model depends on data, our proposed model is unable to predict words that are not in its lexicon. A small dataset made up of 8K (8 Thousand) photos from the Flickr dataset is used in the proposed model. In this paper, further we discussed a CNN model called the Xception model which can be trained on the imagenet dataset in order to extract image features. By feeding the extracted features into the LSTM model, image captions are then produced. Sometimes the output generated is wrongly predicted because of less datasets taken. Thus for production-level models and for better accuracy training datasets must be larger than 100,000 images.

6.2 Future Scope or Future Enhancement

- After implementing and extracting image captions in English using CNN and LSTM, a model can be built to show languages such as Kannada, Urdu and Hindi among others.
- Better or more accurate findings than the suggested model can be acquired by using larger datasets.

REFERENCES

[1] Ali Ashraf Mohamed, "Image Caption Using CNN and LSTM", researchgate.net, publication/342407897, Article May 2022.

[2] Swarnim Tripathi, Ravi Sharma, **"**Image Caption Generator Using CNN and LSTM", IJCRT, IJCRT_196552.

[3] Chetan Amritkar, Vaishali Jabade, **"**Caption Generation for image using DL Technique", JCT Journal.

[4] Inaam Ilahi, H.M. Abdullah Zia, M. Ahtazaz Ahsan, M. Rauf Tabassam and Armaghan Ahmad, "Caption Generation in Urdu using Attention based LSTM", 19 June 2021.

[5] Source: Internet/Online, zendesk.com, machine-learning-and-deep-learning.

[6] Source: Internet/Online, data-flair.

[7] Source: Internet/Online, tutorialspoint.com, Google Colab.

[8] Source: Internet/Online**,** geeksforgeeks.org, Jupyter Notebook, Python.

[9] Source: Internet/Online**,** techtarget.com, TensorFlow.

[10] Source: Internet/Online, tutorialspoint.com, Keras.

[11] Source: Internet/Online, codingninjas.com, Convolutional Layers.

[12] Source: Internet/Online, geeksforgeeks.org, deep-learning-introduction-to-long-short-term-memory.

[13] Source: Internet/Online, analyticsvidhya.com, introduction-to-long-short-term memory-lstm.

[14] Source: Online/Internet, javatpoint.com, deep-learning.

[15] Source: Online/Internet, netapp.com, artificial-intelligence and deep-learning.

[16] Source: Online/Internet, analyticsvidhya.com, deep-learning-activation-functions.

AUTHORS PROFILE

Afeefa Nazneen N Z D/0 Zakir Hussain T K grew up in a Hobali Nayakanahatty, Chitradurga, Karnataka, India, where she completed her primary education tutoring by Shri Allama Prabu and Shrimati Thippamma as her close and lovely tutors. She completed her secondary education in Dr. Thimmareddy Foundation PU College for Girls. She received her B.E [CS&E] degree from BIET, Davangere in 2019. She is pursuing her M.Tech [CS&E] degree at University BDT College of Engineering, Davangere, Karnataka, India. Her area of interests includes Mathematics, Machine Learning and Deep Learning.

Dr. Shreedhara K S received his PhD degree from Manipal University, Manipal in 2008. His area of interests includes Image Processing, Deep Learning and Compiler Design. Currently he is working as professor in the Department of CS&E, University BDT College of Engineering, Davangere, Karnataka, India.

Volume 11, Issue 4
August 2022

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X