

Abstract

Recent advances in advanced reading and computer vision-based machine translation have resulted in superior graphic models using advanced techniques such as deep learning. Although these models are very accurate, they tend to rely on the use of expensive computing hardware, making it difficult to use them in real-time situations where you can run real-world applications. This model uses a hybrid CN-NRNN model, where the CNN part of the model system uses the Xception model for transfer learning, and RNNs are widely used in language modeling. The Flickr8k dataset is used for real-time training and testing. RNN's LSTM model is used to avoid problems with extinction or gradient explosion during the training phase.

Key Words: CNN, LSTM(Long Short Term Memory), RNN, Xception

1.1 PROJECT IDEA

- For the machine to automatically interpret the objects in a picture and their relationships or actions performed using a learned language model is a challenging task, but with a huge impact in many areas. As a precautionary measure it can help people with visual impairments to better understand visual inputs, thus acting as an assistant or guide.
- Its purpose is to mimic the human ability to understand and process large amounts of visual information in a descriptive language, which is a fascinating challenge in the field of AI.

1.2 MOTIVATION OF THE PROJECT

- **Aid to the blind** - We can create a product for the blind that will guide them on the streets without the support of anyone else. We can do this by first translating an article into text and then the text into a word. Both are now popular Deep Learning applications.
- **Self driving cars** - Automatic driving is one of the biggest challenges and if we do not say the exact location of the incident near the car, it can improve the self-driving system.
- Automatic captions are useful and can make **Google Image Search** similar to Google Search. That way, all images can be converted to captions first and then searched based on the captions.
- In **Web development**, it is a good practice to give a description of any image from the page so that the image is readable or audible rather than just plain visible. This makes web content accessible.

- **CCTV cameras (closed TV cameras)** are everywhere today, but World Viewing allows you to sound an alarm if you can recreate the relevant captions every time something bad happens. This may help reduce some crimes and accidents.

2.1 LITERATURE SURVEY

In Literature review, various references of the existing projects are taken into consideration.

1. The article "Understanding Convolutional Neural Networks" describes one of the deepest emotional networks known as Convolutional Neural Networks (CNNs). There are many changes in CNN. Convolutional layer, non-linear layer, integrated layer, fully integrated layer, etc. CNNs perform well on machine learning problems and one of the most common algorithms.
2. In the paper "The Generation Awareness Business Image", Modern photo captions produce descriptive definitions, such as businesses with words involved in photography. Here Di Lu, Spencer Whitehead had proposed a very new project that produces descriptive caption captions, rendered images as embedded. A simple solution to the problem we propose is to train a CNN-LSTM model to produce image-based captions.
3. In the article on automatic neural image caption generators for the visually impaired, automatically defining image content in well-organized English sentences can be a daunting task, but it can be very helpful in helping the visually impaired. .. Modern mobile phones can take pictures and help people with poor eyesight. Here, the image as input can generate captions high enough for the visually impaired to hear to see the surrounding objects better. Here, Christopher Elamri uses the CNN model to extract the image elements. These elements are then integrated into the RNN or LSTM model to generate an image definition with grammatically correct English sentences that describe the environment.
4. In the paper "Neural Image Caption Generator", the basic problem of artificial intelligence that combines computer vision and natural language processing

automatically defines image content. In this article, A.L systematically analyzes the process of generating labels for neural networks. Here, the photo is presented as an insert and as a way to get out in the form of English sentences that explain the content of the photo. They analyzed three components of this method: convolutional neural networks (CNNs), recurrent neural networks (RNNs), and sentence generation. This model analyzes the image and generates keywords and important words in the image.

5. In his work "LONG TERM MEMORY", Sepp Hochreiter describes the Short Term Neural Memory Network algorithm of the Short Term Team (LSTM). LSTMs are both spatially and temporally local. Computational complexity is each step in the process and represents a weighting pattern. Compared to other algorithms, LSTMs run better and learn much faster. It also solves complex, long-term tasks that previous iterative networks could not.

3.1 PROBLEM STATEMENT

To capture image and to recognize the context of an image and describe them in a language like English.

3.1.1 Goals and objectives

Goal and Objectives:

- To recognize objects in the images and then generates caption for that image.
- To learn the concepts of a CNN and LSTM model and build a working model of Image caption generator by implementing CNN with LSTM.

3.1.2 Assumption and Scope

- System accepts images as input extensions: .png , .jpg , .jpeg

3.2 METHODOLOGY

1. **Import all the required packages data cleaning** - The main text file containing all subtitles is Flickr8k.token in the Flickr8k text folder. Each image has 5 captions (0-4) with a number associated with each caption.
2. **Extract features vector** The system uses a pre-trained model that has already been trained on a large dataset and removes features from the model. Apply them to our business. The system uses an Xception model trained in an image database. This model has 1000 different classes to divide. Implement this model directly in your camera settings. Note that the Xception model uses an image size of $299 * 299 * 3$ as input. The system removes the last partition layer and captures 2048 vector features.

3. **Loading Flickr8k dataset** - The Flickr 8k test folder contains a Flickr8k.trainImages.txt file that contains a list of 8000 image names used for training..
4. **Tokenizing the vocabulary** - The computer does not understand English words. In computers, the system needs to represent them numerically. Therefore, place a map of each vocabulary with a unique index number. The Keras library provides tokens that are used to create tokens from your vocabulary.
5. **Defining the CNN-RNN model** - To define model design, the system will be using the Keras Model from the Functional API. It will have three main parts:
 - **Feature Extractor** – The feature extracted from the image has a size of 2048, with a thick layer, it will reduce the size to 256 nodes.
 - **Sequence Processor** – The embedding layer will handle the text input, followed by the LSTM layer.
 - **Decoder** – By combining the output from the two upper layers, the system will be processed in a compact layer to make a final prediction. The final layer will contain a number of nodes equal to our vocabulary size.
6. **Training the model** - To train the model, the system uses 8000 training images by performing input and output sequences and grouping them into the model using the model.fit generator () method. The model is also saved in the model folder. It may take some time depending on the system capacity.
7. **Testing the model** - We will create a separate caption test file that generator.py will download the model and make predictions. Predictions contain the maximum length of the index values so the system will use the same tokenizer file to get names from their index values.

3.3 OUTCOME

- Generates a caption for the specified image.

3.4 TYPE OF PROJECT

- It is a project-based project in which the neural image caption model integrates the image caption model into the CNN-RNN framework. It will have a positive effect on the real world, for example by helping visually impaired people better understand the content of the images.
- The domain involved is CNN using LSTM, Machine Learning.

5.1 FUNCTIONAL REQUIREMENTS

- To get real time image as input.
- To identify the objects.
- To provide caption for that image.

5.2 NON FUNCTIONAL REQUIREMENTS

- Availability :- There are no system requirements, but it works fast with a high GPU.
- Maintainability :- Updates to the system database (currently holding 8000 images) can be added to get accurate captions.
- Accessibility :- The system will be accessible to people who want to get photo captions.
- Speed :- Around 5 mins on Nvidia 1660 Ti GPU/ around 1-2 hrs on CPU.
- Usability :- People with eye problems will be able to use the program.

5.3 CONSTRAINTS

- Operational Constraints:-
 - Input image format must be .jpg, .jpeg, .png
- Hardware Constraints:-
 - The system meets the minimum requirements.
- Software Constraints:-
 - item The target image for the system must have a resolution higher than 500x375.

- Assumptions ;-
 - CNN is correctly able to detect object and using LSTM caption id formed.
 - Images subjected are in preferably higher resolution than 500x375.

5.4 HARDWARE REQUIREMENTS

- CPU with 64-bit support (Recommended High-end GPU)

5.5 SOFTWARE REQUIREMENTS

- Operating System: Windows
- IDE: Jupyter Notebook
- Programming Languages: Python3
- Libraries and Packages: Numpy, Keras, Tensorflow

6.1 ARCHITECTURAL DESIGN(BLOCK DIAGRAM)

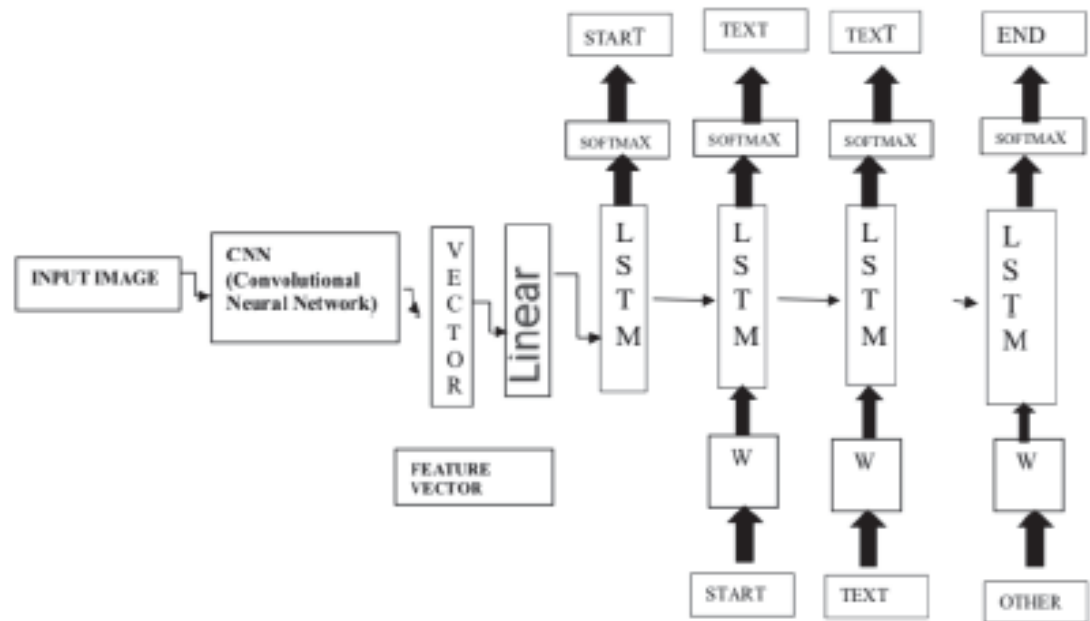


Figure 6.1: Proposed model of Image Caption Generator.

The proposed caption generator model is shown in Figure 6.1. This dense vector, also known as embedding, can be used as an integration with other algorithms to produce the appropriate captions for a particular image as output.

Captions With caption products, this embedding is a representation of an image and the first LSTM format used to create meaningful image-based subtitles.

System Architecture of our system is shown below in Figure 6.2

This is our proposed system architecture will look like.

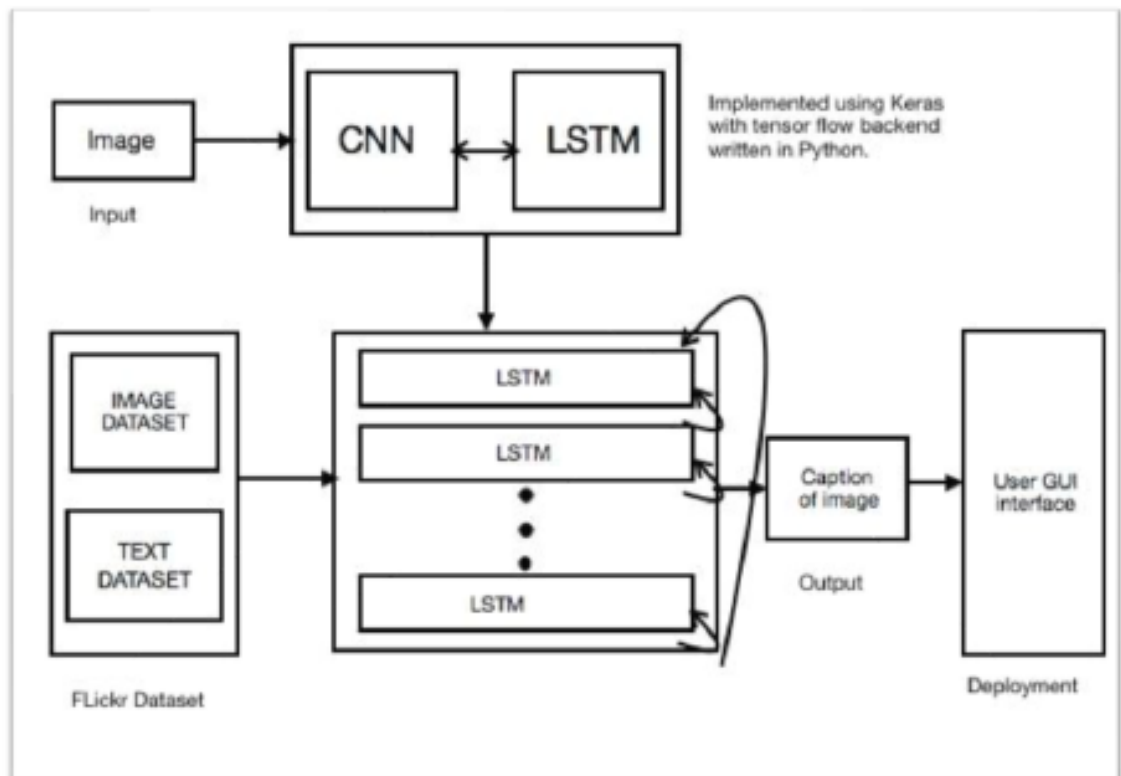


Figure 6.2: System Architecture of Image Caption Generator.

LSTM stands for short-term memory, a type of RNN (Continuous Neural Network) suitable for sequence prediction problems. Based on the previous text, you can predict what the next name will be. Overcoming RNN limitations with temporary storage has proven to be effective with traditional RNNs. The LSTM can generate relevant information during input processing and through the forgetting gateway and discard non-essential information.

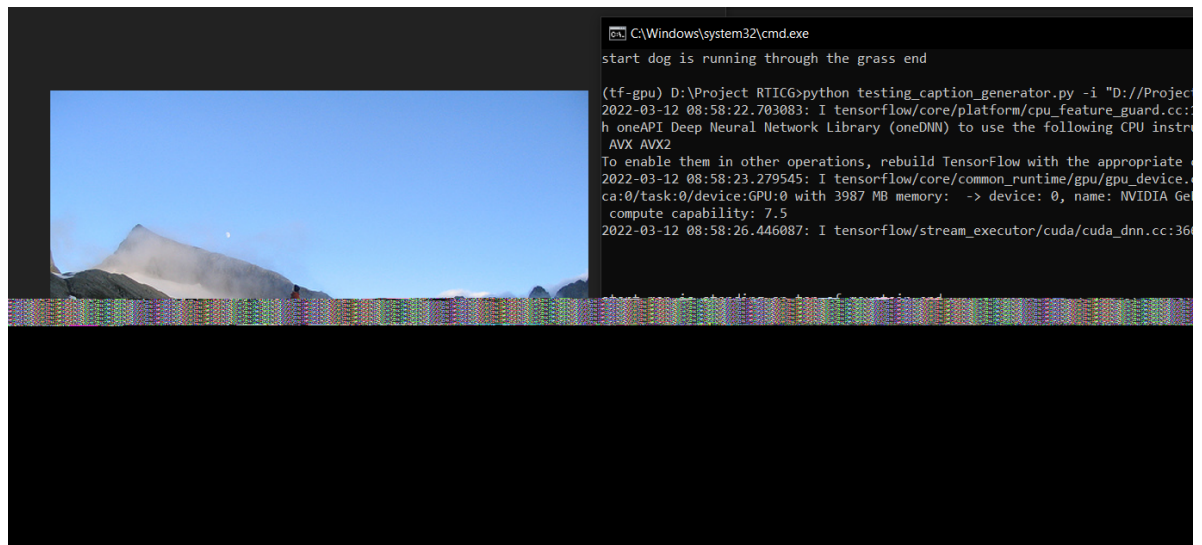


Figure 6.5: User Interface Example 2.



Figure 6.6: User Interface Example 3.

6.3 DATA DESIGN

- Generate captions on rendered images using caption generator using CNN (Convolutional Neural Networks) and LSTM (Short-term memory).

6.3.1 Data Structure

- Image assets are exposed in Xception, a CNN model trained in an image database, and a function responsible for creating captions is added to the LSTM model.

6.3.2 Database description

- Models - Will contain our trained models.
- Descriptions.txt - This text file contains all the images' names and captions after pre-processing.
- Features.p - Embellishment containing an image and its vector feature extracted from the previously trained CNN model Xception.
- Tokenizer.p - Contains a token map with a reference value. Model.png - Visual representation of the size of our project.
- Test caption generator.py - Python file to generate captions for any image.
- Training generator captions.ipynb - Jupyter notebook where we train and build a caption generator for our image.

7.1 OVERVIEW OF PROJECT MODULES

- **Feature Extractor** – The feature extracted from the image has a size of 2048, with a thick layer, it will reduce the size to 256 nodes.
- **Sequence Processor** – The embedding layer will handle the text input, followed by the LSTM layer.
- **Decoder** – By combining the output from the two upper layers, the system will be processed in a compact layer to make a final prediction. The final layer will contain a number of nodes equal to our vocabulary size.

7.2 TOOLS AND TECHNOLOGY USED

- CNN
- LSTM
- Xception

7.3 ALGORITHM DETAILS

- CNN Convolutional Neural Network (CNN) is a deep learning algorithm that can take a captured image, assign values (readable weights and biases) to various items / items in the image, and distinguish them from each other. The initial processing required for a CNN is much less than for other partitioning algorithms. Filters are handcrafted in the old fashioned way, but with sufficient training, CNN can read these filters / symbols.
- Xception model Xception is a convolutional neural network with 71 deep layers. Download a pre-trained version of the network trained with over 1 million photos from the site. Pre-trained networks can categorize images into 1000

object categories such as keyboards, mice, pens, and many animals. As a result, the network has learned to represent the rich features of various images. The image input size of the network is 299x299.

- LSTM long-term memory is a kind of continuous neural network (RNN). The RNN provides the output of the last step as input to the current step. It faces the problem of long-term RNN dependencies, where RNNs can predict names stored in long-term memory, but can provide more accurate predictions from the latest information. If the gap length is long, the RNN will not provide optimal performance. LSTMs can automatically store information longer. Used for processing, forecasting and classification based on time series data.

8.1 EXPERIMENTAL SETUP

8.1.1 Data Set

- Flickr8k Dataset :- Dataset folder which contains 8000 images.
- Flickr8k text :-Dataset folder which contains text files and captions of images.

8.1.2 Performance Parameters

- Model Accuracy
- Timing Required to train the model (Around 5 mins for Nvidia 1660 Ti GPU and around 1-2 hours for CPU)
- Timing Required to test the model

8.1.3 Efficiency Issues

- This is a database trained model, so you can predict the names of its members.
- I used a small database of 8000 images. Production-level models require training on datasets containing over 100,000 images. This allows you to create a more accurate model.
- Depending on your system, this process may take some time. When using the Nvidia 1660Ti GPU for training purposes, it can take up to 5 minutes to complete the task. However, it is CPU-intensive and this process can take up to 12 hours.

9.1 CONCLUSION

Although computer vision has made great strides, it is a relatively new task to allow computers to define transmitted images, although features such as visual acuity, behavioral separation, image separation, feature separation, and scene recognition are possible. A human-like sentence format.

This principle requires real-time capture of image semantically based captions to be expressed in a natural language such as English in the desired way. It has a huge impact on the real world, for example, to help visually impaired people better understand the content of images on the web.

Therefore, use CNN to create your own image caption model. Image features are removed via CNN. I used a previously trained exception model. Information obtained from the CNN and used by the LSTM to create image captions.

In this project, I created a caption generator to create a CNNRNN model. Another important point is that the model is data dependent and cannot predict words other than itself. I used a small database of 8000 images. Production-level models require training on datasets containing over 100,000 images. This allows you to create a more faithful model.

9.2 FUTURE SCOPE

- Extend our system by taking large datasets to improve accuracy.

9.3 APPLICATION

- Aid to the blind
- CCTV cameras (Closed-circuit television cameras)