AUDIO CAPTION GENERATION FROM IMAGES USING DEEP LEARNING

[1] Omprakash Yadav, [2] Atharva Jadhav,[3] Abdul Hannan Sunsara,[4] Idris Vohra
[1]Project guide, [2]Student, [3]Student, [4]Student
[1]Computer Engineering,
[1]Xavier Institute of Engineering, Mumbai, India.

Abstract : Visually impaired individuals face various types of difficulties as they cannot visualize the natural environment. To overcome this problem, the proposed system would automatically generate captions for an input images and convert the generated caption to an audio format so that visually impaired individuals can listen to the generated captions. Captioning is performed using Deep Learning algorithm Convolution Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory.

Index Terms – Deep Learning, RNN, LSTM, CNN.

I. INTRODUCTION

Vision deficiency, also known as loss of vision or vision impairment, is nothing but a vision disability which has no immediate remedies of any kind. It is something that any person does not wish to have. Millions of individuals are visually impaired globally, according to the World Health Organization (WHO). Without one of the most useful sensory organs in a technologically developed environment where even the smallest piece of work needs hearing, it is very hard to survive. In today's world where the technology sector is growing, many developments can be made where the technology can provide aid to the visually impaired people. One technique is to detect items in an image and have a meaningful caption that will be in the form of audio that would enable visually disabled people to associate all object to an image. Generating an image caption requires a number of tasks, such as understanding the higher level of semantics, and then explaining the semantics in a sentence that humans can understand. Communication in human beings takes place with the aid of natural language, so it becomes a challenge to create a system that creates explanations that human beings can understand. The goal of this paper is to detect, recognize and generate images using Deep Learning algorithms.

II. RELATED WORK

The generation of natural language descriptions from an image is a relatively new area, though lots of work has been done in this area. This section will give you a brief idea about the work that has been done to date.

In this [1], the system generates captions automatically for the news articles. The caption is generated from a database of various news articles along with an image embedded in them. This model consists of two stages as Content selection and surface realization. Content selection identifies what the image and accompanying article are about, whereas surface realization determines how to verbalize the chosen content.

This paper [2], presents a model that generates captions automatically for news articles with an embedded image in it. Captions for the news articles are generated by using the stemming algorithm and frequency ranking calculation.

This [3] developed system helps smartphone users to generate captions for their photos. Users have to upload a photo to cloud service where a number of parallel modules like face detection, GPS, date-time, scene recognition is applied to recognize a variety of entities and relations. The outputs of the modules are combined to generate a large set of captions.

Deep Visual-Semantic Alignments for Generating Image Descriptions model can automatically generate captions for given images using multimodal RNN [4]. This model aligns sentence snippets to the visual regions that they describe through multimodal embedding then uses this as an input to multimodal RNN that will generate snippets.

In this [5] presented model is based on the concept 'Show and Tell'. This model had used CNN and LSTM to generate a caption. This model has been implemented by using various datasets to obtain more accurate results.

In the proposed model [6], which takes input as an audio file and converts it into text. The conversion is done by using an encoder-decoder scheme with an attentional layer in between them. The method was evaluated using a commercial dataset of recordings, each of which is associated with a textual description (caption) within the dataset.

An automatic caption generation model has been implemented by using Recurrent Neural Network and

LSTM with additional Read-Only units [7]. The model is trained with the MSCOCO dataset which generates more accurate captions by using all these technologies together.

III. METHODOLOGY

The image caption generation problem can be addressed by using a standard encoder-decoder RNN architecture which involves two elements which are the encoder and the decoder. The encoder reads the input image and generates a fixed-length vector

using an internal representation and the decoder takes the encoded input and generates the textual description. In short, the generated output is expected to describe in a single sentence what is shown in the image such as the objects present, the properties, actions being performed and interaction between objects, etc. A pre-trained convolutional neural network (CNN) model can be used to encode the images and RNN, such as a Long Short-Term Memory network can be used to encode the generated text sequence or generate the next word in the sequence. Mentioned below are the technical approaches in detail which are used to build a model.

A.* Convolution *Neural Network.

A Convolution Neural Network is a deep learning algorithm that can take input in the form of the image, assign the learnable weights and biases to various objects in an image and then make one object different from another. A much lower preprocessing is required as compared to other classification algorithms. The architecture of this network is analogous to that of the connectivity pattern of neurons in the human brain. Individually, neurons can respond to the stimuli only in the restricted region of the visual field known as the Receptive field. A collection of such fields then overlap to cover the entire visual area. Also, there is an "observation window" from which the model tries to learn the weights of the filter. The weights are shared for each convolution. It ensures translational equivariance by allowing the model to recognize objects wherever they show up in the image.

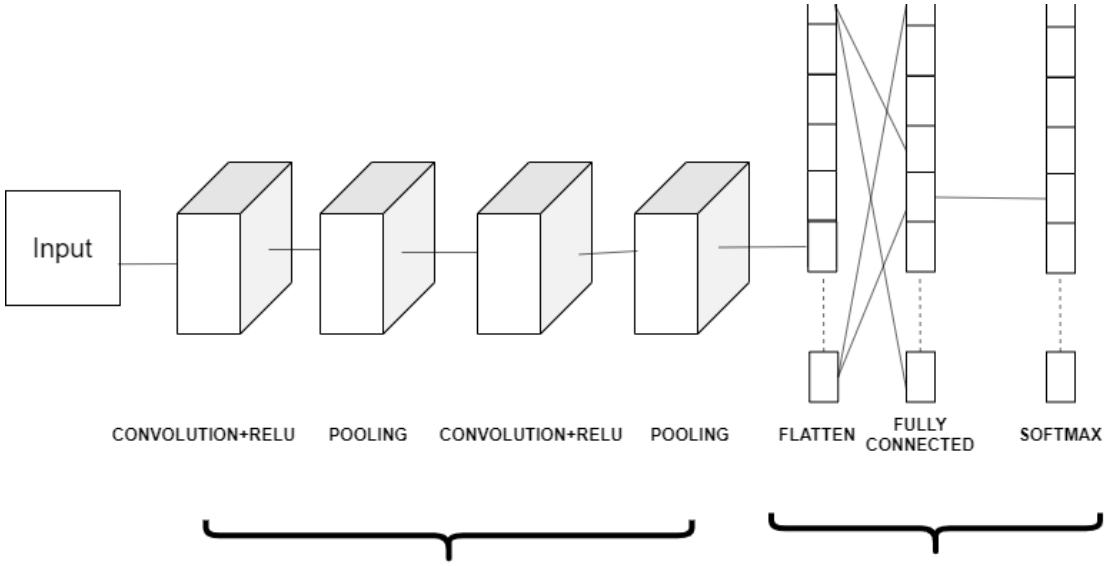

Fig. 1 Sequence of transformations involved in the convolutional neural network [11].

B. Recurrent Neural Network

Recurrent neural network architecture succeeds in being able to capture information about previous states to inform the current prediction through its memory cell state [1]. Generally, in traditional neural networks, all the inputs and outputs are independent of each other. The problem with feed-forward neural networks is that they need a fixed-sized input and give a fixed-sized output. They do not capture sequences or information about time series nor do the account for memory. But in some cases like when it is required to predict the next word of a sentence, the previous words are required and therefore there is a need to remember the previous words. Thus RNN was introduced which solved the issue. It can take variable-sized input and give variable-sized output and it works well with time series data as well. One of the most important features of RNN is the hidden state which remembers some information about a sequence. RNN has a 'memory' because of which it remembers all the information about what has been calculated. For each input, it uses the same parameters as it performs the same task on all the inputs or hidden layers to produce the output. And doing this, it reduces the complexity of parameters.

The intuitive representation of RNN is shown in Fig. 2. The basic formula of RNN is the recursive formula which is as follows:

$$S_t = F_w(S_{t-1}, X_t)$$

Where S_t is the new state at time t, F_w is the recursive function, S_t is the state at time t-1 and X_t is the input at time t.

To deal with the vanishing gradient problem faced by RNN, Long Short Term Memory (LSTM) was introduced.

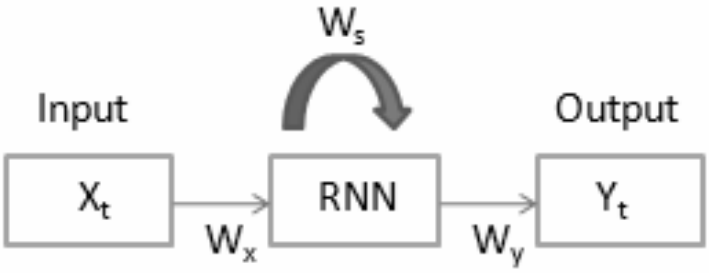

Fig. 2 Representation of RNN

Suppose if there is a deeper network with one input layer, more than one hidden layer and one output layer then RNN basically performs the following actions:

1. Firstly, the conversion of independent activation into dependent activation takes place by providing the same weights and biases to all the layers. It will help in reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer.
2. Therefore all the hidden layers can be joined together such that the weights and biases of all the hidden layers are the same into a single recurrent layer. Fig. 3 shows a simplified version of RNN model.

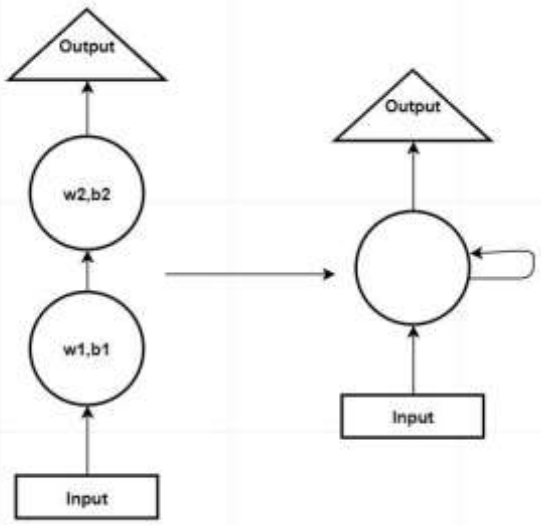

Fig. 3 Simplified form of RNN model

IV. **Dataset**

We have used the Flicker8k dataset to train our model. This dataset contains two types of data, image, and caption (text). This dataset contains 8092 images in JPEG format with different shapes and sizes. Out of which 6000 images are used for training, 1000 for tests and 1000 for development. Fliker8k dataset also contains text data. In that dataset, Fliker8k.token.txt file contains 5 captions for each image i.e. 40460 captions.

V. **TEXT TO SPEECH CONVERSION**

To convert generated captions into audio we have used Android TextToSpeech class. Android allows us not only to convert the text into audio but also we can convert it into different languages. In order to use this class, we need to first initiate an object of this class and specify the Init Listener.

VI. IMPLEMENTATION

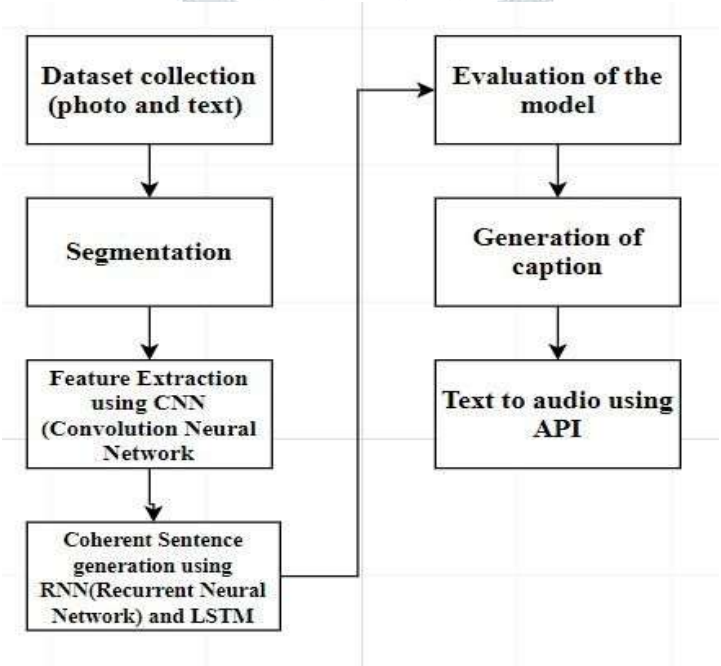

Fig 4. Block Diagram of the System

The very first step of implementation is importing all the necessary packages. The implementation of the project as shown in Fig. 4 is as follows.

1. ***Getting and performing data cleaning***:

The Flickr8k text dataset is used for the purpose of training. The document is loaded and the contents inside the file are read into a string. A description dictionary is created that maps the images in the Flickr8k dataset with a dataset of captions. Then data cleaning is performed on all the descriptions which involve removing punctuation, converting all text to lowercase and removing words that contain numbers. After this, all the unique words are separated and the vocabulary is created from all the descriptions. Finally, a preprocessed list of descriptions is created which is stored in a file.

2. ***Extracting the feature vector from all images***:

This method is often called transfer learning in which pre-trained model which has already been trained on large datasets is used and features are derived from it. We use the VGG16 imaging model for the same thing

3. ***Lading dataset for Training the model***:

This module contains various functions which:

a. Load a text file containing a list of 6000 image names that are used for string training and return a list of image names.

b. Build a dictionary with the captions for each photo from the list of images. For each caption, the <start> and <end> Identifiers are appended. This is required for the LSTM model to define the start and end of the caption.

c. Gives the dictionary for image names and their feature vectors which were previously extracted. The max_length is calculated to decide the model structure parameters. Max_length of description is 34.

4. *Tokenizing the vocabulary and creating data generator*:

Each word of the vocabulary with a unique index value and saved to a pickle file. The model is trained on 6000 images. The sum of data for 6000 images cannot be stored in memory such that a generator process is used to produce batches.

Table 1.
Example of Input And Output To The Model

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

Table 1 shows an example in which the input to the model is [x1, x2] and the output will be y, where x1 is the 2048 feature vector of that image, x2 is the input text sequence and y is the output text sequence that the model has to predict.

5. *Defining the CNN-RNN model*:

The structure of the model consists of three major parts:

1. Feature Extractor – The attribute obtained from the image has a size of 2048, with a thick layer, which restricts the dimensions to 256 nodes.
2. Sequence Processor – The embedding layer manages the text input, followed by the LSTM layer.
3. Decoder –By combining the output from the two layers above, we're going to process the thick layer to create the final estimate. The final layer would have the same number of nodes as the scale of the vocabulary.

6. *Training the model and evaluation*:

To train the model, 6000 training images have been used by generating the input and output sequences in batches and fitting them to the model using model.fit_generator() method. The BLEU metric is used for evaluating and testing the performance of the caption generator.

VII. FUTURE SCOPE

The accuracy of the model can be increased by training it on a larger dataset such as Flickr30k or MSCOCO dataset. An attention mechanism can also be used.

Our work can be extended to the next higher level by enhancing our model to generate captions even for the live video frame. Our present model generates captions only for the image, which itself a complex task and captioning live video frames is much complex to create. This is completely GPU based and captioning live video frames cannot be possible with the general CPUs. Video captioning is a popular research area in which it is going to change the lifestyle of the people with the use of cases being widely used in almost every domain. It automates the major tasks like video surveillance and other security tasks.

The system can generate the caption in multiple languages and not only English. This will enable users from various regions to get an idea of their surroundings in their local language. Also, the scope of usage of the app will not be limited only to the users who know English.

VIII. CONCLUSION

The proposed system generates captions for photos taken on a smartphone. The system is specifically built for helping blind people in knowing what is happening in their surroundings. The system operates by sending an image to a server where all the processing is done by using deep learning algorithms like CNN, RNN, and LSTM. The blind people will be benefited from this app as the app will describe the surroundings in the form of audio. Improvisations would be made in order to expand the recognition ability of the system. These include the ability of the conversion of continuous captions into the video and many more. Another feature that could be added is to learn more from the feedback provided by the user after using the application.

IX. ACKNOWLEDGMENT

Mr. Omprakash Yadav, directed the creation of this project. Her extensive knowledge of Machine Learning and Computer Vision has given us a greater understanding of the application development process.

X. REFERENCES

[1] Feng, Y., & Lapata, M. (2013). Automatic Caption Generation for News Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(4), 797–812.
[2] Vijay, K., & Ramya, D. (2015). Generation of caption selection for news images using stemming algorithm. 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC).
[3] Ramnath, K., Baker, S., Vanderwende, L.,
El-Saban, M., Sinha, S. N., Kannan, A., Torresani,
L. (2014). AutoCaption: Automatic caption generation for personal photos. IEEE Winter Conference on Applications of Computer Vision.
[4] Karpathy, A., & Fei-Fei, L. (2017). Deep Visual-Semantic Alignments for Generating
Image Descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4)664–676.
[5] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[6] Drossos, K., Adavanne, S., & Virtanen, T. (2017). Automated audio captioning with recurrent neural networks. 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.
[7] Poghosyan, A., & Sarukhanyan, H. (2017). Short-term memory with read-only unit in neural image caption generator. 2017 Computer Science and Information Technologies (CSIT).
[8] Vishwash Batra, Yulan He, Neural Caption Generation for News Images (2018) George Vogiatzis School of Engineering and Applied Science, Aston University
[9] Galvez, R. L., Bandala, A. A., Dadios, E. P., Vicerra, R. R. P., & Maningo, J. M. Z. (2018). Object Detection Using Convolutional Neural Networks. TENCON 2018 - 2018 IEEE Region 10 Conference.
[10] Xu, N., Liu, A.-A., Wong, Y., Zhang, Y., Nie, Y., & Kankanhalli, M. (2018). Dual-Stream Recurrent Neural Network for Video Captioning. IEEE Transactions on Circuits and Systems for Video Technology, 1–1.
[11] Saha Sumit, (2015, Dec). A Comprehensive Guide to Convolutional Neural Networks the ELI5 way.towardsdatascience. Retrieved from https://towardsdatascience.com/
.