International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

IMAGE CAPTIONING FOR VISUALLY IMPAIRED

B. Uthkarsh Jaiswal -uthkarshj@gmail.com, K. Devashish Singh -devashishsingh02@gmail.com, Shaikh Asif Ahmed -yaseenshaik46512@gmail.com, Unnati Khanapurkar- unnati.khanapurkar@gmail.com

ABSTRACT: -
Image captioning has always been a great source of help for visually impaired by generating captions for the given image. But limiting it to the captions won't be that helpful for the visually challenged. In this project we tried to give voice to our generated captions by using the concept for TTS that is text-to-speech which is more impactful and practical.To accomplish caption generation and to implement Deep learning architecture we have used Tensorflow and Keras. it is challenging to generate captions that have right linguistic properties because it requires sophisticated level of image understanding. In this project, we used VGG16 deep learning architecture for the purpose of feature extraction for the images. The generated caption's quality and accuracy are evaluated using BLEU score
KEYWORDS:Imagecaptioning,Deep Learning,Neural Networks, Text-to-speech, BLEU Score.

INTRODUCTION:-

Caption generation for the given image has always been a fascinating model in the era of computer vision and machine learning. Understanding images, extracting features and translating visual scenes in the images into plain text and converting the plain text into speech are all elements of this project. The goal of this project is to generate appropriate captions for a given image and converting them into speech which enhances the experience of visually impaired. To generate accurate captions for the given input image, our project uses convolution neural networks(CNNs) and recurrent neural networks(RNNs) or their branched models. CNNs like VGGs provide architecture for extracting features from the images and RNNs like LSTM model is used for the purpose of caption generation. For the purpose of extracting features we use CNN model like VGG16 and removed the last two layers as those layers are used for the purpose of classification. RNN model like Long-short-term memories (LSTM) are especially helpful in sequential prediction. Images and captions from large, labelled data sets, such those in Microsoft COCO and Flickr, offer details about the events and objects. The generated captions' appropriateness and relevance to the data set captions can be evaluated using metrics such as BLUE score. Lastly we can use TTS libraries like gTTS,pyttsx3 can be used for the purpose of text-to-speech conversion. Over the past few decades, a number of pertinent research articles have sought to conduct this task, but they have encountered a number of difficulties, including linguistic problems, cognitive absurdity, and irrelevant content. In order to get over those problems, we developed this method, which uses computer vision and natural language processing techniques to extract pertinent content and properly structure sentences, making the model usable by visually impaired individuals.

PROBLEM IDENTIFICATION &OBJECTIVES:-

Through the use of image captioning, recent technology breakthroughs have made it possible for many visually impaired persons to use their capacity to hear what is around them. Natural language processing (NLP) and deep learning technologies like CNN are where the idea of picture captioning originated. When we are unable to use our eyes, it is extremely difficult to comprehend what is going on around us. However, we are still able to hear what is going on. It is quite difficult to help visually challenged to provide information about the surroundings, but the idea of image captioning and converting the generated caption into speech is the best possible way we can help them. This can be accomplished with the help of training the machine on 2D images and process it to predict the objects that are present in the image and able to form sophisticated sentences and autoplay text-to-speech model. To create such a model, the neural networks suggested are convolutional neural networks(CNNs) and recurrent neural networks(RNNs). Convolutional Neural Networks

like VGG or YOLO can be implemented for feature extraction of the images. Recurrent Neural Network like Long-Short term memory(LSTM) can be impactful in sequential prediction. This RNN us used for generating captions. For this project, we also used the Flickr 8k instructive collection, where each picture includes five captions. Any nature photo can have a caption created for it by this application based on what's in the photo. With the help of such a tool, people with vision impairments could view the vast majority of photographs on the internet. This method automatically generates natural language captions for social media tagging, image indexing and searching, helping the visually impaired, and other uses. In this research, we will create an interface using Long Short-Term Memory (LSTM) for caption generation, Convolution Neural Networks (CNN) for feature extraction and TTS model for converting the generated caption into speech. For the highest level of success, GPU-based registration must be used for the Deep Learning assignments.

SYSTEM METHODOLOGY:-

In order to achieve our objective that is image captioning is divided into five main categories. Loading the data, pre-processing the data are the initial stages. Afterwards, features are extracted from the pre-processed images. After these stages appropriate models are put to work for training and testing purpose. Finally, the captions generated are given speech. Google's Tensorflow library and Keras were used throughout the project. We used Google Colab to train our model as it gives cloud based platform with GPU support which is quite helpful for such models.

ARCHITECTURE:-

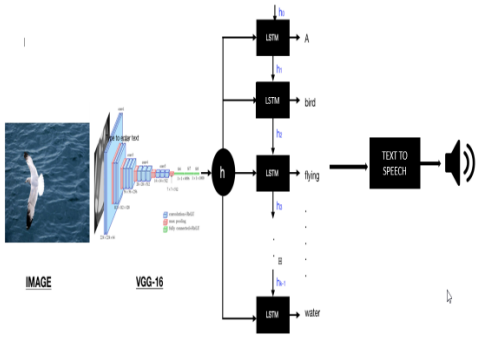


Fig 1:-Conversion of image to text and TTS

In the initial stages, the pre-processing of the image which is in the format of .jpg takes place and the CNN model like VGG16 assists in this task. Now, using LSTM captions are being generated which provides the information of the given image and finally, the generated image caption is given a speech.

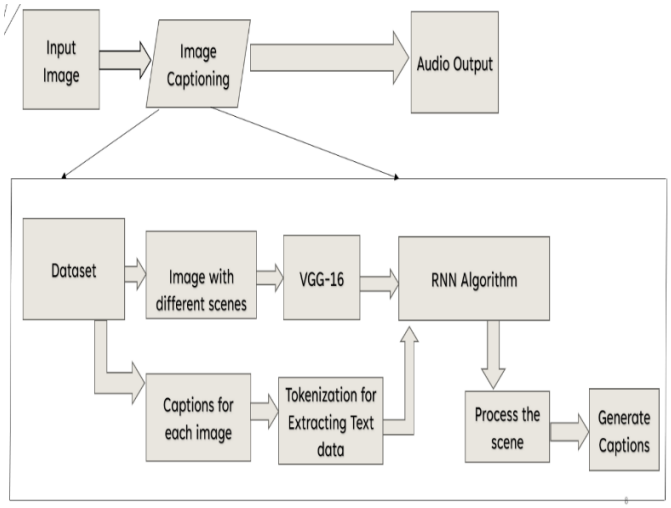
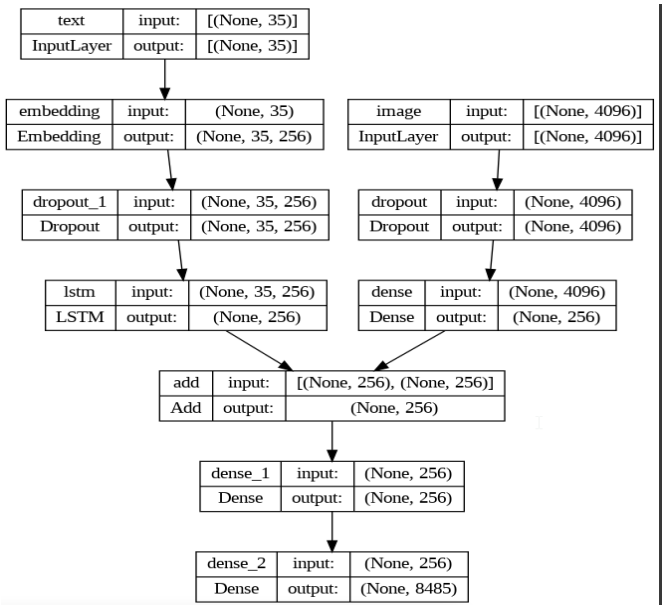


Fig 2:-Block diagram

In the initial stages, for every image to be captioned the input images are in the form of .jpg and there will be list of words associated with it. After this, the input image is fed to CNN like VGG16 for the purpose of feature extraction. The extracted features of the image are stored in the form of vector and using the Dense we reduce the nodes to 256. For future use we are saving these extracted features in a file named "features.pkl". Conversely, the list of words is provided to the sequence processor, which receives the list of words linked to the image as input. The embedding layer reduces the size of the list to 256 nodes, after which the LSTM layer is applied to the image to create a more precise and meaningful sentence. Now, the Dense layer is utilized to combine the outputs from the feature extractor and sequence processor. The SoftMax layer is used as an activation function prior to the output layer, providing an effective and understandable caption for each image.



In the above diagram we can see how different layers are being merged to accomplish the goal of image captioning.

IMPLEMENTATION:-

A. About Dataset

A key component of our new system for caption-based image description and search is the Flickr8k dataset, which we obtained from Kaggle. With eight thousand well chosen photos and five distinct captions for each, it provides a wide variety of situations and activities. Interestingly, the absence of notable people or items in these pictures ensures that different situations are the main emphasis. Selected from six different Flickr groups, they portray a variety of events and circumstances. Flickr8k and Flickr30k are two of the most well-known image caption training datasets in computer vision research. Flickr8k is currently our favorite because of its efficient storage requirements and manageable size, even though both allow comprehensive annotation. Furthermore, we use a cleaned-up version of the Flickr8k dataset, which has been divided into train and test subsets and has been selected by Andrej Karpathy. This guarantees uniformity and promotes easy interaction with our framework.

B. Importing Libraries/Models Required

We will load a few libraries first, including those needed for data representation, display, and setup, as well as those that work with active models. These modules are all used in various stages of a project. Every module in this project is crucial for providing models and creating the graphs that we use with tqdm.

1.Numpy: For Mathematical Calculations

2.Tqdm: To display progress indicators for the loops

3.Tensorflow: This python package is used for building machinelearning

4.Keras: This provides highly-productive interface for solving ML problems.

5.gTTS: This is Google Translate Text-to-speech library for TTS conversion.

C. Image's Feature Extraction:

This section involves creating an image dictionary, loading a picture from a file, and resizing it to 224,224 pixels. We turn the image into a numpy array after resizing it. In order to extract the characteristics and build a model, this data needs to be reshaped. Since the image is RGB, there will be three samples. We The image was preprocessed for the VGG model, and the features were extracted. Each image was assigned an image ID so that the features could be stored. This process is applied to each image in the dataset, and it takes some time to finish. In order to process multiple photos at once rather than just one, we must employ GPUs for this.

After the features are extracted, we save them in a pickle file so that we can access the dictionary we built. We can now load these features whenever we need them, which saves a ton of time.

This stage makes use of a pre-established CNN network model, which is excellent for designing computer vision applications.

D. Preprocessing Of Text Data:

We split the captions line by line after reading the text data found in the captions.txt file. Every image has five captions once the captions are divided, thus we create a mapping dictionary with the image id as the key and the image name removed.This image's collection of captions is denoted by the jpg extension and value.

Therefore, we have 40455 captions that we process for 8091 photos with 5 captions apiece. The mapping

dictionary currently comprises of pre-processed captions, each of which is captured one at a time.

To avoid the requirement for special characters when speaking, we first translate the captions into lowercase before replacing every one of them. Additionally, we eliminated the extra spaces inserted between each word. In order to make it simple for the model to determine where to begin and stop a sentence, we employed start and end tags.

E. Creating Data Generator Function:

Our model for image captioning tasks is trained using the data generator function. It repeats over image keys and the captions that go with them in an endless cycle. It tokenizes the text for every caption and splits the series to produce input-output pairs. Token sequences and picture characteristics that have been structured for model training make up these pairs. By processing data in manageable batches, the function ensures effective memory usage and prevents session crashes. To arrange and format the data for training, NumPy arrays are used. This generator function makes it possible to analyze huge datasets with ease by facilitating constant data flow during model training.

F. Encoder-Decoder Model creation:

To create this assistive tool, we are creating a model which is of type encoder-decoder.

Encoder: In order to provide visual attributes to RNNs, the encoder must extract features of various sizes and encode them into vector space. VGG-16 and ResNet are two commonly suggested image encoders. We made the decision to modify the pre-trained VGG-16 model from the PyTorch package. CNN is used in this work to encode the features rather than classify the photographs.

Consequently, we removed the fully connected layers and max pool from the network's end. The dimensions of the input picture matrix are N X 3 X 256 X 256, and the output of this new structure is N X 14 X 14 X 512. In order to accommodate input photos of various sizes, we also added an adaptive 2D layer to the CNN design.

Decoder: Using recurrent neural networks (LSTMs), which have the ability to output words sequentially, the decoder must provide word-by-word visual descriptions. The encoded image feature vectors from CNN and the encoded picture captions generated during the data pre-processing phase make up the decoder's input. The decoder is made up of four fully connected layers: an attention module that we created and built an LSTM cell module. Upon receiving encrypted images along with captions, we initially classify the files based on the length of the encryption key. To optimize output and save training time, we only want to analyze the encoded images whose caption lengths match or exceed the number of repetitions. Before the LSTM network iterates, we first feed the encoded images and the network's historical state into the attention module to create the attention-masked images with a specific area highlighted. The subsequent state of the LSTM is created by combining all of the captivating visuals with the embedded captions of the previous words. Based on the current conditions, it is therefore possible to forecast and increase the likelihood of the current word embedding. Then, using the current state as a basis, the probability of the current word embedding may be estimated and contributed via fully linked layers to the word embedding prediction matrix.

G. Implementing Encoder-Decoder model:

Let us look into the implementation for the encoder-decoder model. In order to extract pertinent features from images, the encoder passes the input features through a dense layer and a dropout layer. In order to obtain sequence information, the decoder simultaneously processes textual input sequences via an LSTM layer, a dropout layer, and an embedding layer. The decoder processes the features through a dense layer after combining the encoder's features with the sequence features. Lastly, it uses a SoftMax activation to create a probability distribution throughout the language. The Adam optimizer and categorical cross-entropy loss are used in the compilation of the model. It makes it easier to jointly learn text and image attributes in order to create captions. We used the plot_model() function to depict the architecture and display the relationships between the levels. With this method, the model may produce insightful descriptions for images by considering their context and visual content.

H. Caption Generation for Image:

The processes involved in creating a function for creating a new caption are as follows:

1. Give the generating process a start tag.

2. Input sequence encoding.

3. Inflate the order.

4. Assume the following word.

5. Obtain the index with a high likelihood.

6. Convert a term to an index.

7. If not found, stop.

8. To create the next word, append the word as input.

9. When we get to an end tag, stop.

EVALUATION:-

Although we have built the model according to our requirements, it is not yet evident if it is quite accurate and capable of providing the desired learning. The next step is to train the model in such a way that it gives the accurate and acceptable outputs. To get the knowledge about the accuracy we use few performanceevaluation metrics

A. Evaluation Metrics:

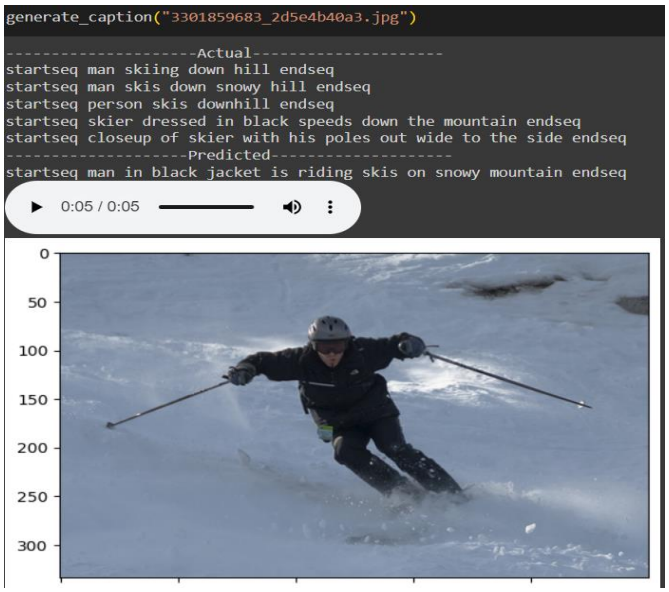
For evaluating our model, we are using BLEU score. BLEU (Bilingual Evaluation Understudy) score helps in evaluating the machine generated text and gives the quality of the text generated. Here, we used BLEU-1 and BLEU-2 score

B. Model Performance

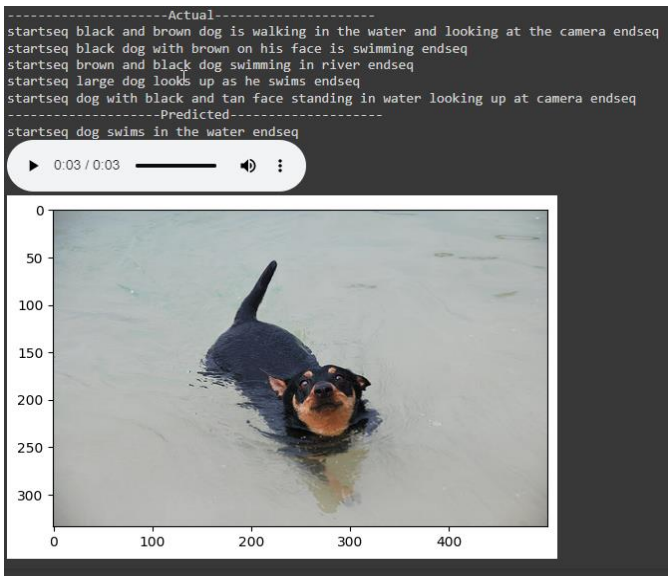
The BLEU score ranges from 0 to 1 and the highest BLEU-1 score we managed to get is 0.5284 in 15 epochs. As of BLEU-2 score for this we got 0.3014 and our average BLEU score comes out to be 0.4149. Our model is performing quite average.

RESULTS:-

As mentioned before, data is converted to numbers using a vocabulary of words. Following the consumption of the data, the output from the It will also be in an encoded format that requires conversion back into English sentences that can be understood by humans. Another important factor is that the RNN network's output is a sequence of word likelihoods (likelihoods). In an RNN, selecting the word with the highest likelihood at each decode stage typically yields less-than-ideal outcomes. LSTM, a popular technique for choosing the optimum path for word interpretation in natural language, has been utilized instead. A few test image captions that were created are shown below. It appears that some of the network's automatically generated captions miss important details from the image, while others misidentify specific visual aspects.

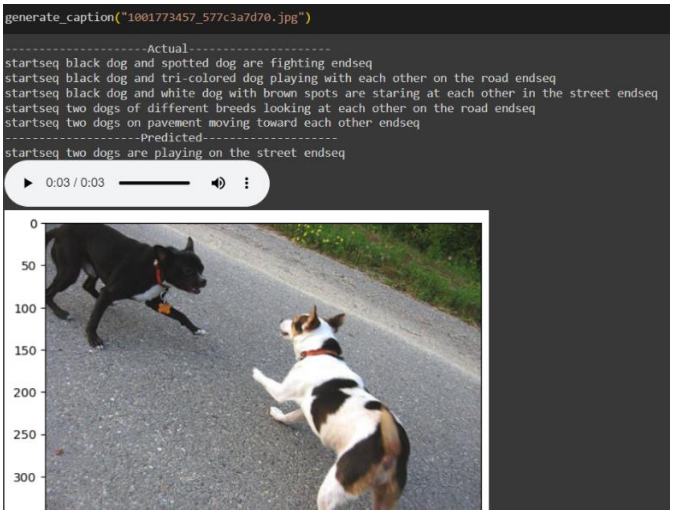


At this stage, testing results for object detection and image captioning when the model was trained on a GPU host are displayed. In the image captioning step, an input image is first given to VGG16-no-FC, which is used to extract image features. An attention mechanism that extracts a relative range of the objects' targeted region uses these qualities as inputs. Finally, descriptive sentences can be generated using the LSTM network. Our text-to-speech converters can also be used to turn the created texts into speech. Bling people can listen to the speech produced by the converter through headphones or any audio device available.

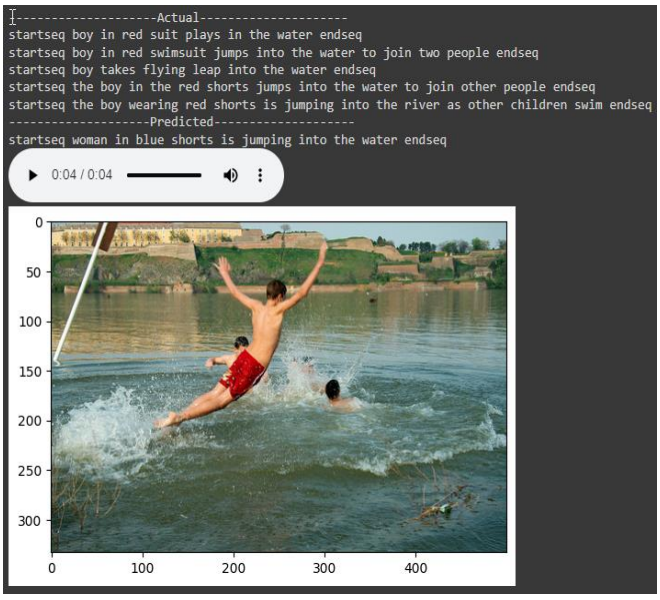


#Image of output (correct one)

There were some eye-catching observations in our project that is the test pictures that contained images of dog were captioned very accurately. This is probably because there are many images of dogs and their related activities that are present in the training part of dataset. Due to many dog pictures being present in the dataset the pretrained CNN is also very knowledgeable on how the dog looks like.



There was one more thing that was noticed that is, our model occasionally fails to separate objects and identify colours. For an example, in the below picture we can see a man is jumping in the lake but it predicts something of the chart.



CONCLUSION:-

From this research project we can conclude that we were able to generate captions to the given image using the encoder-decoder technology and further we could also give speech to the generated captions with the help of gTTS library which provides help to the visually impaired in much practical way. This model may explain the scenario in the given image and also clearly describing the colours that are present in the image which give more detailed view of the image and in-turn helping even more to the visually challenged. To evaluate the performance of our model we used BLEU (Bilingual Evaluation Understudy) score whose value ranges between 0 to 1. Overall, we were successful in achieving our goal that was generating detailed captions and converting the generated caption to speech.

Since we worked using a smaller dataset (Flickr8k), we could only fulfill the objectives of this paper. We can obtain better results for this model if we used improved and variety of datasets like Flick30k and MS COCO. The accuracy of the output will be even more if we used such datasets and also using generative adversarial network (GAN) to fill in the recovered item image's backdrop. As of the future works we can also conclude that this model can be even trained to generate outputs in multiple languages rather than just in English. In order for deep learning to be beneficial to everyone, there needs to be an increased emphasis on improving the quality of life for those who are blind or visually impaired.

References:-

- Ponnaganti Rama Devi, Mannam Thrushanth Deepak, Morampudi Lohitha, M. Surya Chandra Raju , K. Venkata Ramana, "Image Caption Generator Using VGG and LSTM For Visually Impaired" International Journal of Advances in Engineering and Management (IJAEM)Volume 5, Issue 4 April 2023

- Muhammad Abdelhadie Al-Malla, Assef Jafar and Nada Ghneim, "Imagecaptioning model using attention and object features to mimic human imageunderstanding" Al-Malla et al. Journal of Big Data (2022).

- Thivaharan S, Vasanthakumar A, Vishal K, Vishnudarshan S, "Deep Learning Based Image Captioning In Regional Language UsingCNNANDLSTM"International Research Journal of Engineering and Technology (IRJET Volume: 10 Issue: 05 | May 2023

- Christopher Elamri, Teun de Planque, "Automated Neural Image Caption Generator for Visually Impaired People"2016

- R. Kavitha , S. Shree Sandhya , Praveena Betes , P. Rajalakshmi , and E. Sarubala, "Deep learning-based image captioning for visually impaired people"E3S Web of Conferences 399, 04005 (2023)

- Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares, "Image Captioning: Transforming Objects into Words", In proceedings of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019),2019.

- L. Zhou, C. Xu, P. Koch, and J. J. Corso, "Image caption generation with text conditional semantic attention," arXiv preprint arXiv:1606.04621, 2016.