Eye for Blind - Audio Captioning of Images

Hardik Panchal
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070054@iitb.ac.in

Kalpit Borkar
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070029@iitb.ac.in

Durgaprasad Bhat
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
200070017@iitb.ac.in

***Abstract*—The project aims to address the daily life adversities faced by the visually impaired, with the core pipeline for the 'Eye for the Blind' system. The system uses cutting-edge computer vision and NLP techniques to generate rich audio prompts that summarize the input image. This system currently functional for static images, can be upgraded to form the basis for a real-time surrounding captioning system. Our hybrid model, which integrates inception V3 with Attention-based GRUs and Google text-to-speech, achieves significant results. We use the Flickr 8K image-cation dataset to train and evaluate our architecture.**

***Index Terms*—Inception V3, CNNs, GRU, Google text-to-speech (gTTS)**

I. Introduction

In the modern world, overloaded with visual information, persons with visual impairments face profound challenges. Most surroundings with which people interact on a day-to-day basis, be it indoors, outdoors, or in workspaces, are not visual impairment-friendly. With this project, we aim to leverage cutting-edge computer vision and natural language processing techniques to aid the lives of such persons.

In this project, we build the core pipeline for an image-to-audio system. The pipeline will be able to generate detailed and contextually rich audio prompts that will summarize the input image. This pipeline can then be upgraded to work with a live video feed, which can form the basis of a wearable surrounding awareness system for the visually impaired. A body-mounted camera can provide a live video feed of the surroundings to the pipeline. The audio prompt generated by the pipeline can be delivered to the person using Bluetooth or a wired earpiece.

This paper will delve into the technical details of the implementation and the methodologies explored by us for the various sub-tasks involved in the problem. In section II, we discuss similar work which has been done previously. We also highlight advancements in methodologies that will be used in formulating our solution. Section III gives an overview of the challenges in the problem and how we approach solving them. We take a look at the dataset we used for training and evaluating our model in section IV. The architecture of the final model used for tackling the problem statement has been detailed in section V. Section VI briefs about the nuances in the training procedure used for learning the model weights. We showcase our results in section VII. Discussions on some key observations, conclusions, some pointers to key resources can be found in sections VIII, IX, and X, respectively, followed by acknowledgment and references.

II. Background and Prior Work

A. Image to text

With the rise of LSTMs, research related to speech processing has gained a lot of momentum in the past few years. Research works such as text-to-speech, speech-to-text, image-to-text, etc., have gained traction in recent years. By loading pre-trained weights borrowed from the Inception V3 model trained on the ImageNet dataset, feature extraction from images is made easy. Combining state-of-the-art CNNs, encoders, attention models, and decoders allows us to create models that can generate text from given images. Image captioning requires us to combine both computer vision as well as natural language processing knowledge.

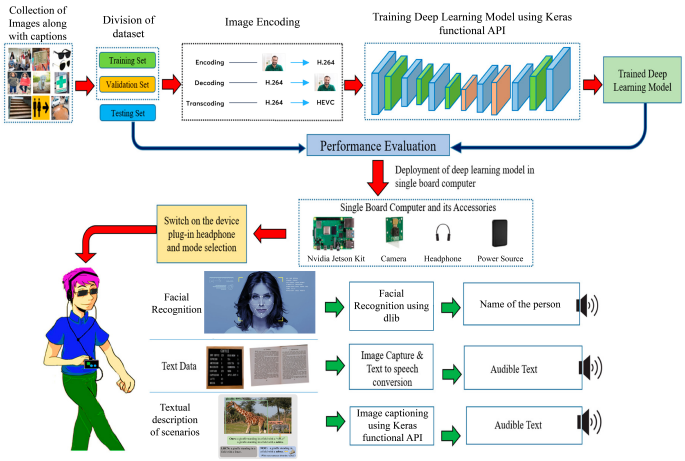

Fig. 1: Overview of the problem and it's solution [1]

B. Text to audio

Generating synthetic speech from text is achieved using autoregressive models by training them for duration prediction and knowledge distillation. Google's text-to-speech library (gTTS) provides us with ready-to-use state-of-the-art results for audio generation from text.

III. Approach

The aim of the project is to develop a pipeline that takes images as input and generates captions for them in the form of a short audio snippet. Models used for tasks related to images are very different than those used for tasks based on audio datasets. Pipelines that deal with images are designed to detect spatial constructs along with their spatial context and perform various tasks. E.g., CNNs, GANs. On the other hand, pipelines that work with audio are designed to learn and exploit the temporal context of the serial data and achieve their target. E.g., Transformers, RNNs, etc. The problem at hand poses a unique challenge that cannot be handled completely by either of the two classes of models. This necessitates us to come up with a hybrid model which has characteristics of both families of models. The initial stages need to be CNN-like, which can capture objects and their spatial context and convert it to some sort of a temporal structure which can then be fed to a sequential type model that generates the captions. We decided to break the problem into 2 chunks: image-to-text and text-to-audio. We chose a dataset that has images of day-to-day scenarios and multiple possible test captions for each of them.

The model consists of 3 major parts:

- Pre-trained model-based encoder: Uses a pre-trained CNN model to convert the image to an embedding
- GRU-based decoder: Uses the features generated by the encoder to iteratively generate the words of the caption, using the previous word and an attention model
- Text to speech using Google text to speech libraries.

IV. Datasets

Flickr 8k dataset is a new benchmark collection for sentence-based image description and search, consisting of 8092 images that are each paired with five different captions, which provide clear descriptions of the salient entities and events. The images were chosen from six different Flickr groups and tended not to contain any well-known people or locations but were manually selected to depict a variety of scenes and situations.

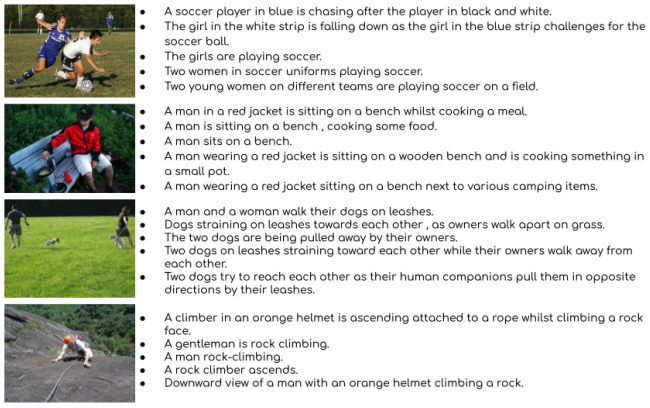

Fig. 2: Flicker8k dataset sample [6]

V. Model Architecture

Our model will be in the form of encoder-decoder architecture, which will take the image and make an embedding using CNN based encoder. This embedding will be passed to some sequential model-based decoder, which will generate the sentence for our image. Finally, that sentence will be passed to a text-to-audio conversion pipeline to get our audio output, which can describe the image accurately for a blind person.

A. Encoder Architecture

This will be composed of a CNN architecture. We will exploit transfer learning and take an ImageNet-based pre-trained model as our base model and then fine-tune it to achieve better performance on this dataset where it'll work as an encoder taking the image into some latent space which will have enough features of images encoded that it'll help decoder to convert it to sentence.

The inception v3 model was released in the year 2015, it has a total of 42 layers and a lower error rate than its predecessors. Let's look at the different optimizations that improve the inception V3 model.

The major modifications done on the Inception V3 model are,

- Factorization into Smaller Convolutions
- Spatial Factorization into Asymmetric Convolutions
- Utility of Auxiliary Classifiers
- Efficient Grid Size Reduction

The first two things are easily understood as the first one points to the use of small filter convolutions instead of large filter size, and the second one emphasizes the concatenation of many convolutional branches which are asymmetric in nature. The objective of using an Auxiliary classifier is to improve the convergence of very deep neural networks. The auxiliary classifier is mainly used to combat the vanishing gradient problem in very deep networks. Traditionally, max pooling and average pooling were used to reduce the grid size of the feature maps. In the inception V3 model, in order to reduce the grid size efficiently, the activation dimension of the network filters is expanded.

For example, if we have a d×d grid with k filters after reduction, it results in a d/2 × d/2 grid with 2k filters.

This is done using two parallel blocks of convolution and a pooling layer concatenated.

B. Decoder Architecture

Our decoder architecture will be a sequential model comprising a Gated Recurrent Unit(GRU) with an attention mechanism. We'll pass our image encoding at every step of decoding.

1) Attention Model: The attention model will take the feature vector and hidden state and return the context vector and attention weights. The hidden state will be initialized with a zeros vector of appropriate shape.

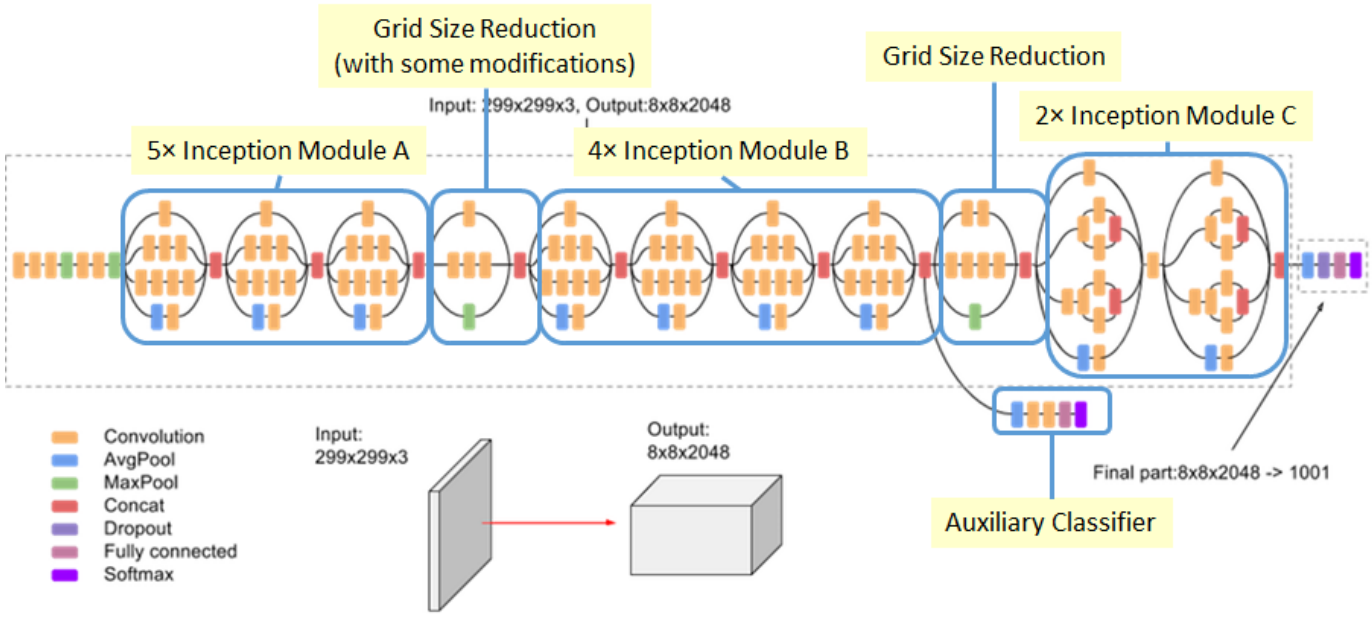

Fig. 3: Architecture of Inception V3 model used as encoder [4]

2) Decoder model: The decoder model will also take the feature vector and hidden state, and it will call the attention model with it to get the context vector and attention weights. Our input from CNN(embedding vector) will also be given to the decoder model, which will pass to the Keras built-in embedding layer, which will give an embedding vector that will be concatenated with the context vector. This concatenated embedding vector will be passed to the GRU unit. The output of the GRU unit will go through a dense layer and, again, one more dense layer to match the shape of the vocabulary size. This prediction will be passed to the loss function, which is categorical cross-entropy loss from keras, and we'll apply a reduce_mean on that to get the final loss output.

VI. Model Training

The optimizer used for training the model is the standard Adam optimizer. Sparse Categorical Cross-entropy loss is used here to compute the cross-entropy loss between the labels and predictions. We are training the model on the T4 GPU of colab for 20 epochs. Our Encoder-Decoder model will be like a next-word prediction model, which will take an embedding vector and attention weights to provide the next-word prediction.

VII. Model Evaluation

Now, how to build a logical sentence using this next-word prediction model? At the start, we'll pass the ¡start¿ token to this model, and it will give some probability distribution. Now, we have used two algorithms to make the best prediction.

A. Greedy Search

In the greedy method, we'll just look for the next one-word probability distribution and use it to choose the best possible next word till the sentence length limit reaches, or the end

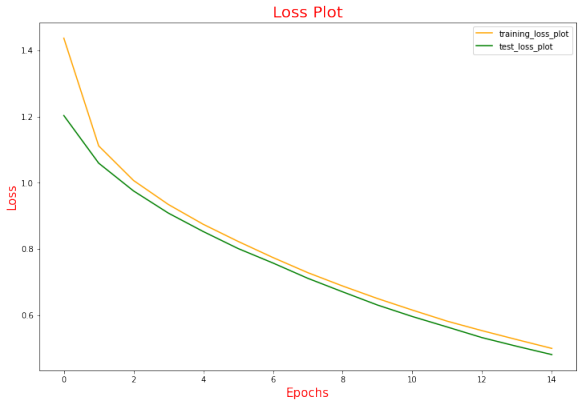

Fig. 4: Training and Test loss

of sentence token ¡end¿ appears. The sentence length limit is decided by dataset inspection, looking for max possible sentence length or average length of sentences. For example, in our dataset, the max sentence length is 39, so we will allow the evaluation model to generate a sentence of a maximum of 39 words in length.

B. Beam Search

With a beam width of 3, we frame sentences using beam search. During decoding, the algorithm investigates several pathways parallelly with a beam width of 3. According to our model, the algorithm generates several candidate sequences at each step based on the probabilities of future tokens. Because of the beam width of 3, the algorithm keeps only the top three candidate sequences with the highest probabilities at each step and discards all other sequences. This narrowing

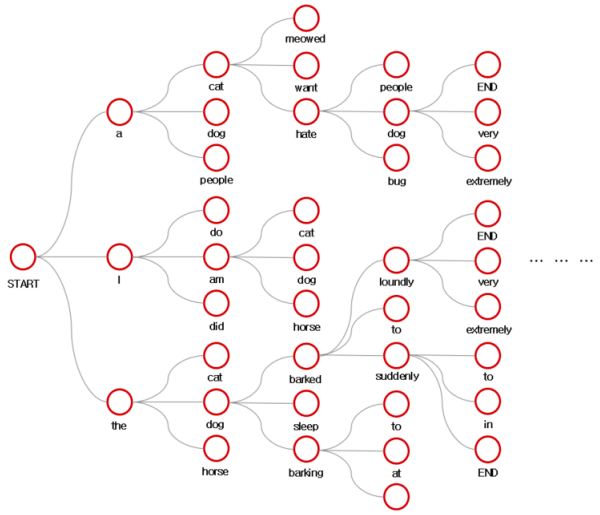

Fig. 5: Beam search with width = 3 [3]

helps strike a balance between exploration and exploitation, enabling the algorithm to choose the most probable paths while considering other options. The sequence with the highest overall probability is selected as the final output by the algorithm after the loop is completed. We might have an end-of-sentence ¡end¿ token early on, and sentences can be of various lengths due to that.

VIII. Results

The architecture trained, as elaborated above, gave decent results on the test images; the following are some examples.

Fig. 6: Example 1
Real Caption: a young girl standing in front of a fountain
Prediction Caption: a boy sit into the fountains

It can be observed that the caption is a little off, but it still manages to recognize the salient features of the images well. The Attention map in Fig. 7 clearly depicts the contribution of each part of the image in generating a particular word in the caption.

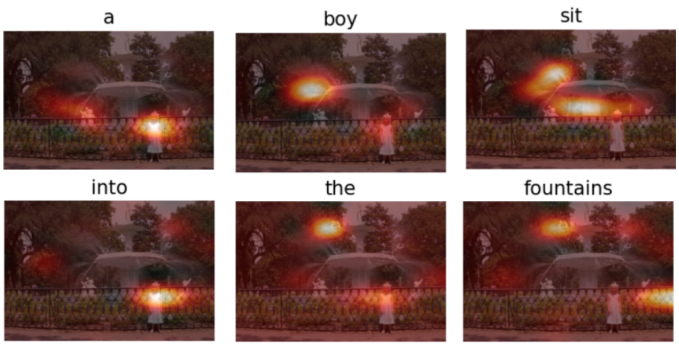

Fig. 7: Attention map for each word in E.g. 1

Fig. 8: Example 2
Real Caption: a dog jumps in the air with a lady standing near
Prediction Caption: a woman shoots a stick

The caption here is also quite relevant; if not perfect, the attention map in Fig. 9 also follows human intuition. Overall, the results seem to be fairly reasonable, given the complexity of the model and the amount of data used for training. This work can be used as a starting point for the development of a system that can generate more accurate and rich captions and can work with real-time video feeds.

IX. Conclusion

In conclusion, the 'Eye for the Blind' project presented in this report addresses the crucial challenges visually impaired individuals face by leveraging computer vision and NLP techniques. The pipeline is currently functional on static images

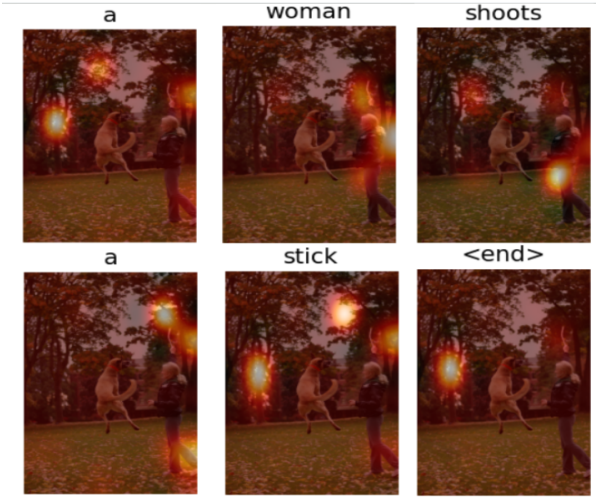

Fig. 9: Attention map for each word in E.g. 2

only, but with some modification and faster inference, we can make it work on video captioning, too. The model uses concepts like integrating Inception V3, Attention-based GRUs, and Google Text-to-Speech modules. Based on Inception V3, the encoder architecture exploits transfer learning and fine-tuning. The decoder, employing GRU with an attention mechanism, seamlessly converts image features into text sentences. The system is more adaptable to varying search lengths by evaluating through greedy and beam search methods. Finally, we can try new architectures for the encoder and decoder to further improve performance.

X. Key Links

Link to the GitHub repository containing all code files: link
Link to the demo video of our project: link

Acknowledgment

We would like to express our sincere gratitude to Professor Amit Sethi for their invaluable guidance and support throughout the duration of this project. Their extensive knowledge, constructive feedback, and unwavering encouragement have been instrumental in the successful completion of this project. Their insightful suggestions and recommendations have been critical in improving the quality of this project.

References

[1] https://github.com/jashan20/Vision-For-Blind
[2] https://www.kaggle.com/datasets/adityajn105/flickr8k
[3] https://sooftware.io/beamsearch/
[4] https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c
[5] https://tinyurl.com/bdz8e4ny
[6] https://forms.illinois.edu/sec/1713398
[7] https://www.kaggle.com/code/tripsankur/eye-for-blind-image-to-text
[8] https://www.kaggle.com/code/anushkaml/eye-for-blind-nn-rnn-attentionmodel/notebook