Eye For Blind

By
Gouthami C

Introduction

Problem Statement:

Eye for Blind is an Image Captioning application for the Visually Impaired.
In this project, we will convert an image to text description first; then, using a simple text-to-speech API, we will extract the text description/caption and convert it to audio. So the central part of this capstone is focused on building the caption/text description using deep learning encoder-decoder model with attention layer(CNN-RNN model) whereas the second part, which is transforming the text to speech is using text to speech API.

Objective

- **Goal:** To Develop a deep learning model for image caption generation which takes image as an input and generates the description or caption for an image.
- **Target Users:** Visually impaired individuals.
- **Use of Attention Mechanism:** Using attention mechanism with encoder and decoder model to enhance caption quality.

Project Pipeline

Overview of the four major steps involved in project building:

1. Data Understanding
2. Data Preprocessing
3. Train-Test Split
4. Model Building
5. Model Evaluation

Data Understanding

- Source of Dataset: Kaggle, Flicker 8k dataset https://www.kaggle.com/datasets/adityajn105/flickr8k
- It is a Sentence based Image description dataset
- It contains 8000 images and five captions for each image

Data Preprocessing

Load image paths and captions into the project.

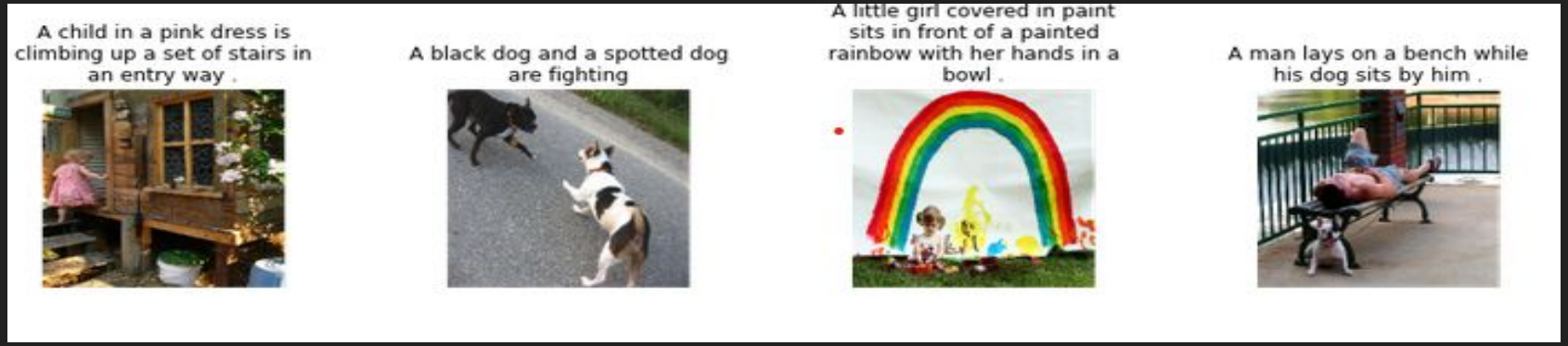


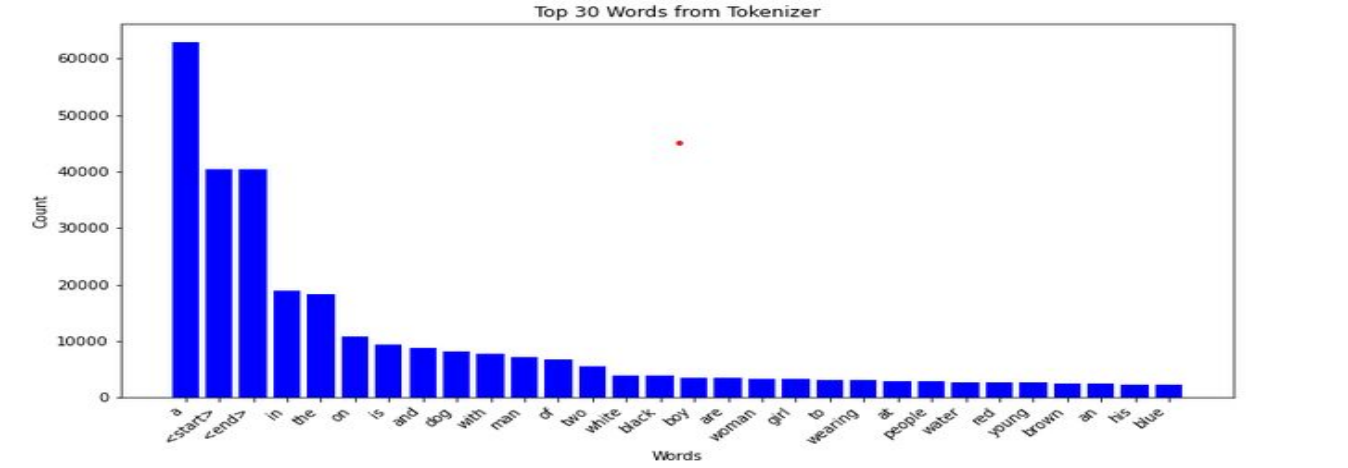
Image Preprocessing involves following steps:

- Loading images: Load images into the model using the file path
- Resizing images: Resize the image into size 299 * 299
- Normalizing pixel values: Normalize pixel values between -1 and +1.

Data Preprocessing

Captions or Text preprocessing:

- Clean the text by removing punctuations, convert text to lower case
- Add the <start> and <end> to each statement
- Limit the vocabulary to 5000 and Tokenize the keywords
- Convert words to index and index to token tensors.
- Padding the sequences to max length



Train Test Split

- Divide image and captions data as test and train datasets.
- 80% of data as a train data and 20% as a test data with random state 42.
- Using this data create a tensorflow dataset with shuffle and buffer size.

Model Building

Encoder- decoder architecture with attention mechanism is used.

1. Extract features from images using Imagenet Inception V3 model by loading pre-trained model whose output is in the form 8*8*2000
2. Create Encoder model where it takes extracted image features and creates a context vector.
3. Attention Mechanism: Which helps to give importance to different parts of input image for each features and decoder hidden state.
4. Decoder model: Which consists of GRU units to decode the captions

Model Building

Training process:

- Selecting Loss Function: Such as SparseCategoricalCrossentropy
- Selecting Optimizer: Adam optimizer

HyperParameters:

- Batch Size: 64
- Epochs: 15
- embedding_dim = 256
- GRU units = 512
- vocab_size = 5001

Model Evaluation

Model Evaluation is using BLEU score. Which tells number of n-grams or sequence of words overlap between real captions and predicted captions.

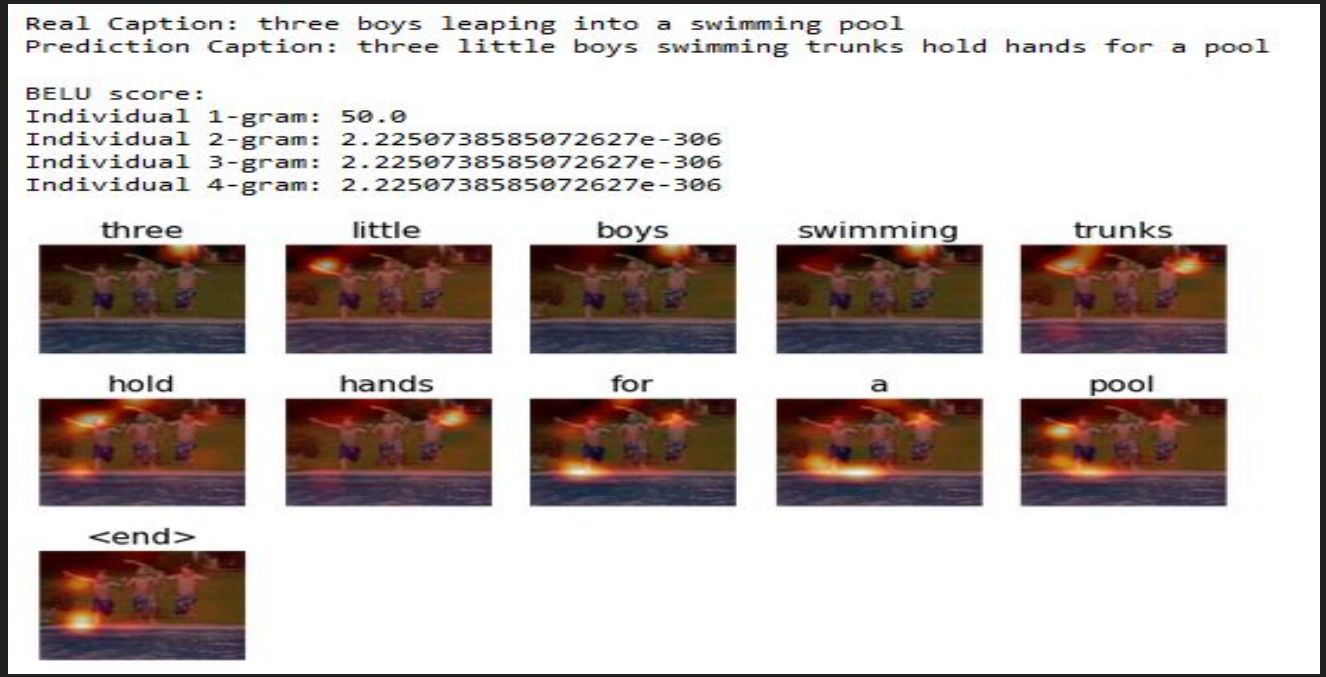
Captions are generated two ways 1. Greedy approach 2. Beam Search

Greedy search is straight word approach on each time step word with highest probability is selected.

Beam search selects k values at each time step which is time consuming based on the k-values and gives enhances output quality.

Caption Generation On Unknown Data

1. Using Greedy Search: Snapshot below shows caption generation using greedy search with attention mechanism highlighting different parts of input at each timestamp



Caption Generation On Unknown Data

2. Using Beam Search: Enhances the quality of output generated

Conclusion

Caption generated are converted to Audio using google text to speech API using python gTTS package.

Conclusion:

- Model is trained only with 15 epochs.
- Model can be fined tuned using hyperparameter tuning for more accurate predictions.

Thank You