Image Caption Generator with Accessibility Features

Prathyusha Reddy Nandikonda
Computer Science
University of Massachusetts
Lowell, MA
prathyushareddy_nandikonda@student.uml.edu

Ranga Rohit Nallamolu
Computer Science
University of Massachusetts
Lowell, MA
rangarohit_nallamolu@student.uml.edu

Naga Sai Tejaswi Gandu
Computer Science
University of Massachusetts
Lowell, MA
nagasaitejaswi_gandu@student.uml.edu

***Abstract*—Our project aims to revolutionize accessibility for visually impaired individuals by leveraging cutting-edge computer vision and natural language processing techniques to generate accurate and detailed descriptions of visual content. By developing an innovative image caption generator, we seek to bridge the gap between the visual world and those with visual impairments, enabling them to interact more effectively with digital content. This system holds immense potential beyond aiding visually impaired users; it extends to applications in robotic vision, business, image indexing, social media, and various natural language processing domains. Utilizing deep learning methodologies, our approach strives to emulate human-like image description capabilities, ultimately enhancing user experiences by automating the generation of descriptive captions for images and environments.**

Keywords—Accessibility, Visually impaired individual, Deep learning methodologies, Image caption generator, Natural language processing.

I. Introduction

In today's visually driven digital realm, ensuring accessibility for individuals with visual impairments remains a pressing challenge. Our project is dedicated to developing a novel solution—an image caption generator—that aims to address this crucial gap by enabling visually impaired individuals to access and understand visual information effectively.

The foundation of our project is an effective approach that makes use of the Flickr 8k dataset, comprising 8000 images, each accompanied by five descriptive captions. This dataset serves as the foundational resource for our research, providing a diverse range of images and their corresponding textual descriptions.

Our approach unfolds in two fundamental stages. The initial phase involves employing Convolutional Neural Networks (CNN) to extract essential features from the images, facilitating a deeper comprehension of their visual content. Subsequently, we utilize Recurrent Neural Networks (RNN), specifically Long Short Term Memory (LSTM) architectures, to generate coherent and meaningful textual descriptions based on the extracted image features.

By merging state-of-the-art technologies in image processing and language generation, our goal is to train a sophisticated model capable of accurately describing visual content, thereby enhancing accessibility for individuals with visual impairments by converting it into speech. Our aspiration is to contribute to a more inclusive digital landscape, where everyone, regardless of visual abilities, can engage with and comprehend visual information effortlessly.

II. Motivation

Our project's motivation is to empower visually impaired individuals in the digital landscape. By harnessing computer vision and natural language processing, we aim to provide accurate image captions, fostering a more inclusive experience for those with visual impairments. We strive to break down barriers, allowing these individuals to access and comprehend visual content effortlessly, enhancing their interaction with the digital world. Our goal is to ensure equal access and participation for everyone, enabling a more meaningful and inclusive online experience for individuals with visual impairments.

III. Related Work

The related work in the field of image caption generation using CNN and RNN techniques has showcased significant advancements in making visual content accessible. [1]Panicker et al. (2021) contributed to the International Journal of Innovative Technology and Exploring Engineering with their research on an image caption generator. Their utilization of CNN and RNN methodologies demonstrated promising results in generating descriptive captions for images.

Similarly, [2] Kotak (2021) explored the realm of image caption generation in the International Journal of Engineering Research and Technology. Their study focused on implementing CNN and RNN techniques to generate captions for images, presenting further insights into the effectiveness of these methods in enabling accessibility to visual content.

Both research papers underscore the efficacy of employing Convolutional Neural Networks (CNN) for feature extraction from images and Recurrent Neural Networks (RNN) for generating coherent textual descriptions. These findings form a critical foundation for our project, inspiring and guiding our efforts to develop an image caption generator with enhanced accessibility features for individuals with visual impairments.

IV. Dataset

The Flickr_8K dataset comprises around 8,000 images designated for training, with a set of images—1,000 each—allocated for development and testing purposes, separated into Flickr_8k.trainImages.txt, Flickr_8k.devImages.txt, and Flickr_8k.testImages.txt, respectively. Additionally, the Flickr8k.token.txt file within the Flickr8k_text folder contains five captions for each image in the dataset, providing diverse textual descriptions for the images.

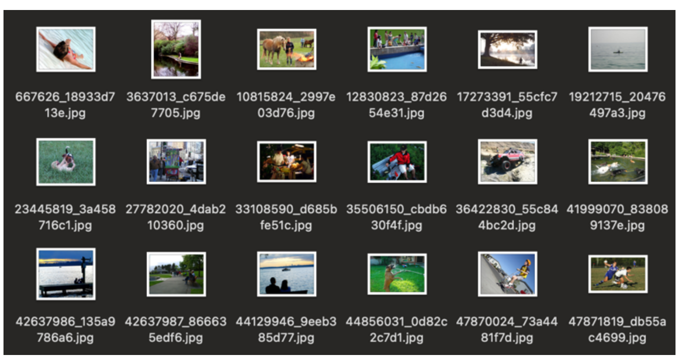


Fig 1: Flickr_8k Data Set

```
1000268201_693b08cb0e.jpg#0	A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1	A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2	A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3	A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4	A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0	A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1	A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2	A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70.jpg#3	Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4	Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg#0	A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
1002674143_1b742ab4b8.jpg#1	A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2	A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
1002674143_1b742ab4b8.jpg#3	There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4	Young girl with pigtails painting outside in the grass .
```

Fig 2: Flickr8k.token.txt

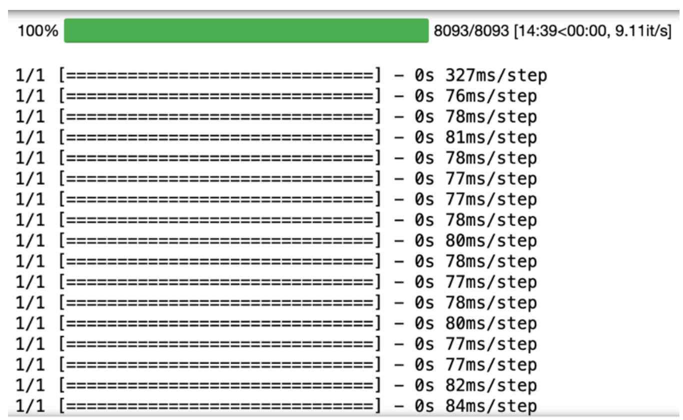
V. Proposed Approach

The overall workflow can be divided into these main steps:

1. Introduction and Dataset Description: The project aims to develop an Image Caption Generator utilizing CNN and LSTM techniques. The dataset, Flickr8k.token, contains images paired with multiple captions, enabling diverse descriptions for each image.

***2. Data Preprocessing*:** Initial functions involve loading the document file, organizing image captions into a dictionary structure, and cleaning textual data. Text cleaning includes removing punctuation, converting text to lowercase, and eliminating words with numbers.

3. Feature Extraction using Transfer Learning: Our project implements transfer learning by utilizing the Xception model pre-trained on the ImageNet dataset. Extracted features from images are obtained by removing the classification layer, generating 2048 feature vectors for each image. These features are stored in a pickle file for future use.



```
100% 8093/8093 [14:39<00:00, 9.11it/s]

1/1 [==============================] - 0s 327ms/step
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 81ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 82ms/step
1/1 [==============================] - 0s 84ms/step
```

Fig 3: Feature Extraction

4. Training Dataset Preparation: The training dataset, sourced from Flickr_8k.trainImages.txt, comprises 6000 image names for model training. We defined functions to load photos, clean descriptions, and extract image features for training purposes.

5. Vocabulary Tokenization: The vocabulary, comprising 7577 unique words, is tokenized to represent words numerically for the model's comprehension. A tokenizer function maps words to unique index values, facilitating the model's language understanding.

6. Data Generator Creation: Due to memory constraints, we employed a generator method to yield input and output sequences, managing the large dataset efficiently. This generator method helps in training the model with batches of data.

7. Model Architecture Definition: The model architecture comprises three main parts:

1. ***Feature Extractor:*** Initially, the image's extracted features (with a size of 2048) are processed through a dense layer to reduce dimensions to 256 nodes.
2. ***Sequence Processor:*** Textual inputs go through an embedding layer followed by an LSTM layer to handle the sequence of words in the captions.
3. ***Decoder:*** The outputs from the above layers are combined and processed by a dense layer to make the final prediction. The final layer contains nodes equal to the size of our vocabulary, facilitating word predictions.

8. Model Training: Training involves utilizing the 6000 images in batches, where input and output sequences are fed into the model using the model.fit_generator() method. The trained model is saved for future use.

```
Dataset:  6000
Descriptions: train= 6000
Photos: train= 6000
Vocabulary Size: 7577
Description Length:  32
Model: "model_1"
_____________________________________________________________________________________
 Layer (type)                Output Shape         Param #   Connected to
=====================================================================================
 input_6 (InputLayer)        [(None, 32)]         0         []

 input_5 (InputLayer)        [(None, 2048)]       0         []

 embedding_1 (Embedding)     (None, 32, 256)      1939712   ['input_6[0][0]']

 dropout_2 (Dropout)         (None, 2048)         0         ['input_5[0][0]']

 dropout_3 (Dropout)         (None, 32, 256)      0         ['embedding_1[0][0]']

 dense_3 (Dense)             (None, 256)          524544    ['dropout_2[0][0]']

 lstm_1 (LSTM)               (None, 256)          525312    ['dropout_3[0][0]']

 add_25 (Add)                (None, 256)          0         ['dense_3[0][0]',
                                                             'lstm_1[0][0]']

 dense_4 (Dense)             (None, 256)          65792     ['add_25[0][0]']

 dense_5 (Dense)             (None, 7577)         1947289   ['dense_4[0][0]']

=====================================================================================
Total params: 5002649 (19.08 MB)
Trainable params: 5002649 (19.08 MB)
Non-trainable params: 0 (0.00 Byte)

6000/6000 [==============================] - 454s 76ms/step - loss: 4.5037
   1/6000 [..............................] - ETA: 8:16 - loss: 4.2338

/Users/prathyushareddynandikonda/anaconda3/lib/python3.11/site-packages/kera
ing: You are saving your model as an HDF5 file via `model.save()`. This fil
end using instead the native Keras format, e.g. `model.save('my_model.keras
  saving_api.save_model(

6000/6000 [==============================] - 443s 74ms/step - loss: 3.6593
6000/6000 [==============================] - 447s 74ms/step - loss: 3.3756
6000/6000 [==============================] - 544s 91ms/step - loss: 3.2055
6000/6000 [==============================] - 443s 74ms/step - loss: 3.0855
6000/6000 [==============================] - 472s 79ms/step - loss: 2.9949
6000/6000 [==============================] - 451s 75ms/step - loss: 2.9266
6000/6000 [==============================] - 430s 72ms/step - loss: 2.8747
6000/6000 [==============================] - 437s 73ms/step - loss: 2.8305
6000/6000 [==============================] - 432s 72ms/step - loss: 2.7944
```

Fig 4: Training the Model

9. Testing the Model: We defined a te which loads the trained model to generate predictions. Predicted index values are transformed back to words using the tokenizer.p file, allowing for human-readable captions. It also predicts the BLEU scores showing the accuracy of the model.

10. Integrating accessibility features: This step involves implementing Google Text-to-Speech (TTS) functionality to convert generated textual captions into audio descriptions for enhanced accessibility.

By incorporating the Google TTS API, our project enables the conversion of text-based image captions into spoken audio files. This process entails sending the generated caption text to Google's TTS service through API requests, which in turn produces an audio rendition of the description.

The resulting audio file serves as an additional accessibility feature, providing visually impaired users with the option to listen to spoken descriptions of the images.

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to evaluate our model, BLEU scores were taken into consideration which demonstrates reasonable performance, indicating a moderate level of linguistic similarity between the generated captions and the ground truth annotations. The generated caption "A man in a red shirt standing on top of a snowy mountain" showcases the model's ability to capture basic elements in the image. The generated caption is then converted to speech and stored in a .mp3 file which ensures accessibility for individuals with visual impairments.

The BLEU-1 score, indicating the similarity of unigrams between generated and reference captions, stands at 0.3549, showcasing a moderate level of overlap. Meanwhile, the BLEU-2 score, measuring bigram similarity, registers at 0.1884, indicating a lesser degree of match between generated and reference text pairs. Furthermore, the BLEU-3 and BLEU-4 scores, assessing trigram and 4-gram overlap, respectively, demonstrate even lower similarity levels, recording values of 0.1272 and 0.0536.

```
100%  1000/1000 [03:58<00:00, 4.53it/s]

BLEU-1: 0.354917
BLEU-2: 0.188429
BLEU-3: 0.127171
BLEU-4: 0.053565
```

Fig 5: BLEU scores

```
start man in red shirt is standing on top of snowy mountain end
<matplotlib.image.AxesImage at 0x38895b010>
```

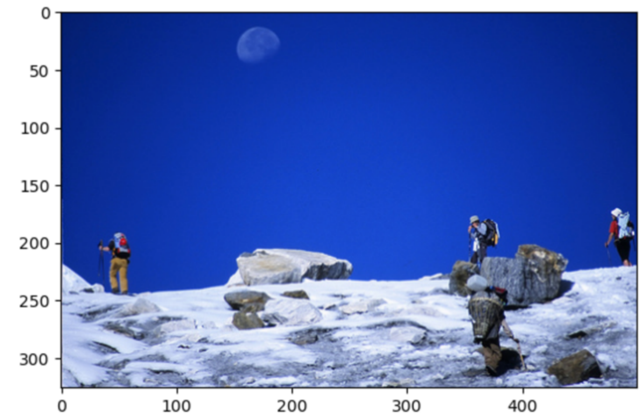


Fig 6: Output of testing the model with an image present in the dataset.

```
start little boy is jumping on the swing end
<matplotlib.image.AxesImage at 0x4031d3010>
```

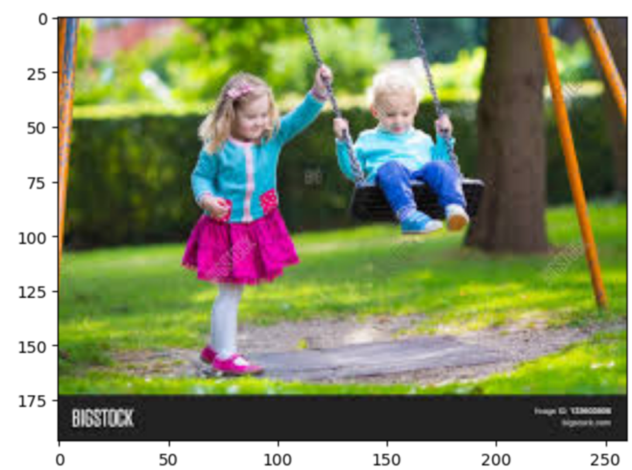


Fig 7: Output of testing the model with an image not present in the dataset.

[https://www.google.com/imgres?imgurl=https%3A%2F%2Fstatic3.bigstockphoto.com%2F3%2F3%2F1%2Flarge1500%2F133603808.jpg&tbnid=OAN2Xm8bbCRMYM&vet=12ahUKEwjLgZL7-YyDAxX4FFkFHRofCFcQMygregUIARCkAQ..i&imgrefurl=https%3A%2F%2Fwww.bigstockphoto.com%2Fimage-133603808%2Fstock-photo-little-boy-and-girl-on-a-playground-child-playing-outdoors-in-summer-kids-play-on-school-yard-happy-kid-in-kindergarten-or-preschool-children-having-fun-at-daycare-play-ground-toddler-on-a-swing&docid=QgRTz4D9c3XjQM&w=1500&h=1120&q=two%20girls%20playing%20in%20a%20ground%20picture&ved=2ahUKEwjLgZL7-YyDAxX4FFkFHRofCFcQMygregUIARCkAQ]

VII. Development Environment and Hardware Requirements

Our Project was developed on a MacBook Air M1 using Jupyter Notebook for ease of experimentation and visualization. We utilized Python 3.x with libraries such as Keras, NumPy, PIL, and NLTK for deep learning, data manipulation, and text processing. Minimum hardware recommendations include an M1 processor or equivalent, 8GB+ RAM, and a dedicated GPU.

VIII. Conclusion and Future work

In conclusion, the presented deep learning model represents a significant step towards automatically generating descriptive image captions, particularly aiming to assist visually impaired individuals in comprehending their surroundings. However, there is a clear scope for refinement in capturing nuanced details and contextual understanding, emphasizing the importance of continued advancements for more accurate and comprehensive image descriptions in aiding the visually impaired community. Integrating Transformer-based architectures into the Image Caption Generator represents a promising avenue for advancing the quality, contextual understanding, and usability of generated descriptions, catering to the needs of visually impaired individuals and broader image description applications.

References

[1] Panicker, M. J., Upadhayay, V., Sethi, G., & Mathur, V. (2021, January). *International Journal of Innovative Technology and exploring ... - IJITEE*. Image Caption Generator. https://www.ijitee.org/wp-content/uploads/papers/v10i3/C83830110321.pdf

[2] Kotak, P. (2021, November 11). *Image caption generator - international journal of engineering research ...* Image Caption Generator. https://www.ijert.org/research/image-caption-generator-IJERTV10IS110121.pdf

[3] V. Kesavan, V. Muley and M. Kolhekar, "Deep Learning based Automatic Image Caption Generation," *2019 Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2019, pp. 1-6, doi: 10.1109/GCAT47503.2019.8978293.

[4] M. Sailaja, K. Harika, B. Sridhar, R. Singh, V. Charitha and K. S. Rao, "Image Caption Generator using Deep Learning," 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2022, pp. 1-5, doi: 10.1109/ASSIC55218.2022.10088345.

[5] Sharma, Grishma and Kalena, Priyanka and Malde, Nishi and Nair, Aromal and Parkar, Saurabh, Visual Image Caption Generator Using Deep Learning (April 8, 2019). 2nd International Conference on Advances in Science & Technology (ICAST) 2019 on 8th, 9th April 2019 by K J Somaiya Institute of Engineering & Information Technology, Mumbai, India, Available at SSRN: https://ssrn.com/abstract=3368837 or http://dx.doi.org/10.2139/ssrn.3368837