A

Project Report

on

Image Captioning Tool to Enhance The Virtual Visibility of Partial Blind People

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2022-23

in

Computer Science and Engineering

By

Piyush Chaudhary (1900290100099)

Prachi Verma (1900290100100)

Ritvik Rohatgi (1900290100123)

Under the supervision of

Naveen Chauhan

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May 2023

DECLARATION

We hereby declare that this submission is our work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature			
Name:	Piyush Chaudhary	Prachi Verma	Ritvik Rohatgi
Roll No.:	1900290100099	19002900100100	1900290100123
Date:	27/05/2023		

CERTIFICATE

This is to certify that the Project Report entitled “Image Captioning Tool to Enhance The Virtual Visibility of Partial Blind People” which is submitted by Ritvik Rohatgi in partial fulfillment of the requirement for the award of degree B. Tech. in the Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date: 27/05/2023

Naveen Chauhan

Project Mentor

ACKNOWLEDGEMENT

It gives us great pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Prof. Naveen Chauhan, Assistant Professor, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness, and perseverance have been a constant source of inspiration for us. Only through his cognizant efforts have our endeavors seen the light of day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department, Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the department's faculty members for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, in the department for their kind assistance and cooperation during the development of our project. Last, but not least, we acknowledge our friends for their contribution to the completion of the project.

Name:	Piyush Chaudhary	Prachi Verma	Ritvik Rohatgi
Roll No.:	1900290100099	19002900100100	1900290100123

Date: 26 May 2023

Signature

ABSTRACT

Suppose for a day you are unable to see anything, and you have to walk around everywhere in this situation. Seems impossible right? This is the problem faced by more than 10 million people daily. Image Captioning tool to Enhance The Virtual Visibility of Partial Blind People is an innovative solution based on Image Captioning that helps the blind to know what their surroundings are. In this project, we use CNN and LSTN to identify the caption of the image. As deep learning techniques are growing, huge datasets and computer power are helpful to build models that can generate captions for an image. This is what we are going to implement in a Python-based project where we will use deep learning techniques like CNN. Image caption generator is a process that involves natural language processing and computer vision concepts to recognize the context of an image and present it in English. Image captioning can be regarded as an end-to-end Sequence to Sequence problem, as it converts images, which are regarded as a sequence of pixels to a sequence of words. For this purpose, we need to process both the language or statements and the images. For the Language part, we used Natural Language Processing and for the Image part, we use Convolutional Neural Networks to obtain the feature vectors respectively.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In today's world, technology is rapidly advancing and changing the way we live our lives. One area where technology has the potential to make a significant impact is in the lives of the visually impaired. Image Captioning Tool to Enhance The Virtual Visibility of Partial Blind People is an innovative IoT-enabled device that aims to empower people who are blind or visually impaired with a new level of independence and awareness. This cutting-edge technology uses machine learning algorithms to generate captions for live images captured by the device, allowing people to understand and interact with their environment in a whole new way. With its ability to identify and caption images, The tool has the potential to significantly improve the quality of life for the visually impaired, providing them with greater autonomy and independence.

1.2 Background and Motivation :

Image captioning is an emerging field in the intersection of computer vision and natural language processing that seeks to automatically generate descriptions of images. The motivation for image captioning is to enable computers to understand and interpret visual content in the same way humans do. This technology has significant potential for improving accessibility for the visually impaired, as well as for enhancing search engines, social media platforms, and other applications that rely on image understanding.

Image captioning has been an active area of research in recent years, with many deep-learning models being developed and evaluated for this task. The popularity of image captioning can be attributed to the fact that images are a rich source of information that can convey complex scenes and events. However, understanding images is challenging for computers because it requires detecting and recognizing objects, understanding their interactions and relationships, and inferring their properties and attributes.

One of the key motivations for image captioning is to bridge the gap between visual content and natural language. By generating textual descriptions of images, computers can communicate the content of images in a way that is easily understandable by humans. This has potential applications in many domains, including education, medicine, entertainment, and social media.

Furthermore, image captioning has the potential to enable new applications that rely on image understanding, such as content-based image retrieval, object recognition, and visual question answering. These applications can improve the efficiency and accuracy of many tasks, from medical diagnosis to automated surveillance.

1.3 Objective

The objective of the Image Captioning Tool to Enhance “The Virtual Visibility of Partial Blind People” is to provide practical and effective solutions for navigating the environment for the visually impaired. The tool aims to use deep learning techniques such as CNN and LSTM to provide accurate and comprehensive descriptions of objects, environments, and situations, helping the visually impaired move around their surroundings with greater confidence and independence. The tool goal of this tool is to significantly improve the quality of life for the visually impaired and provide them with greater autonomy and independence.

1.4 PROJECT DESCRIPTION

Image Captioning Tool to Enhance The Virtual Visibility of Partial Blind People is an image captioning technology that utilizes deep learning techniques, including Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM), to provide accurate descriptions of the environment to the visually impaired. The tool processes visual data to extract relevant features from images, classifying them accurately using CNN, and providing natural language descriptions using LSTM.

The primary objective of the tool is to provide practical and effective solutions for navigating the environment for the visually impaired. By providing accurate and comprehensive descriptions, The tool helps the visually impaired move around their surroundings with greater confidence and independence. The tool can identify and describe a wide range of objects, environments, and situations, providing the visually impaired with a detailed understanding of their environment.

The cutting-edge deep learning techniques employed by the tool make it an innovative solution for the visually impaired. By utilizing CNN and LSTM, the tool can identify and caption images accurately, significantly improving the quality of life for the visually impaired and providing them with greater autonomy and independence.

Say, we as humans are seeing a scene as given

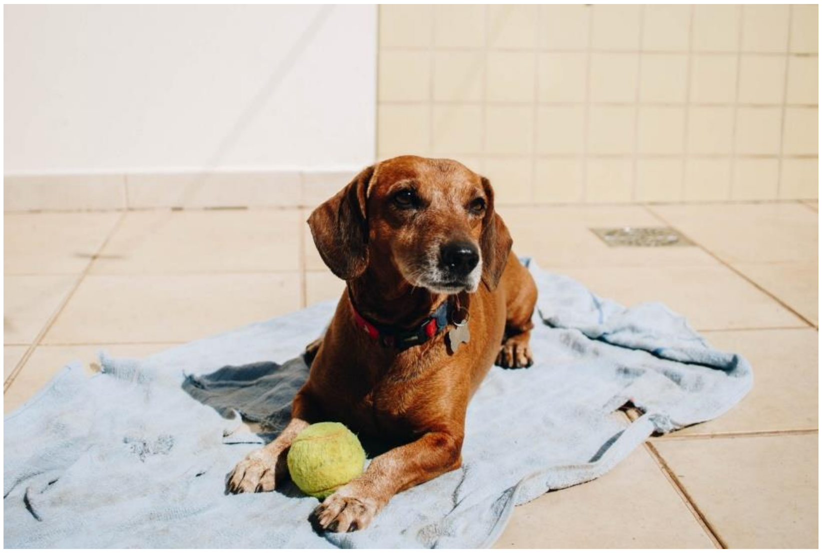


Figure 1.1

If we are told to describe it, maybe we will describe it as: “A puppy on a blue towel” or “A brown dog playing with a green ball”. So, how are we doing this? While forming the description, we are seeing the image but at the same time, we are looking to create a meaningful sequence of words. The first part is handled by CNN and the second is handled by NLP.

Let’s dive into each part in detail:

A). CNN for Image Feature Extraction:

Convolutional Neural Networks (CNNs) have become an essential tool in a variety of computer vision applications, including picture captioning. A CNN’s key job in image captioning is to extract significant visual information from the input image.

A CNN is made up of several layers, each of which serves a distinct role in the feature extraction process. Convolutional layers, pooling layers, and fully linked layers are the most important layers in a CNN.

When you feed a picture into a CNN, it travels through a sequence of convolutions in the convolutional layers. These convolutional layers use filters or kernels to scan the picture and extract various information. These characteristics correspond to visual patterns in the image, such as edges, forms, and textures. The network gradually learns to recognize and extract more complicated and abstract elements by employing many convolutional layers, relying on previously acquired lower-level features.

Pooling layers are critical in downsampling the feature maps generated by convolutional layers. The spatial dimension of the characteristics is reduced yet crucial information is retained. Pooling allows for the capturing of essential information in a more compact manner, which is advantageous for later processing processes.

After passing through pooling layers, the output of the convolutional layers is fed into fully connected layers. These fully linked layers, which are identical to those found in classic neural networks, are used to learn high-level picture representations. Fully linked layers can capture semantic information and store the visual aspects of an image in a compact representation by linking every neuron in one layer to every neuron in the next layer. These high-level representations are critical for comprehending the image’s overall content and context.

The CNN's final output is a vector of visual features. This vector Is sent into the second portion of the picture captioning model, which is usually a language model. The visual elements recovered from the CNN are utilized to construct a caption that describes the content of the input picture. This technique combines the visual information learned by the CNN with the language model’s linguistic expertise to provide a meaningful and cohesive caption for the image.

B). NLP for Language Generation:

Once the image features are extracted by the CNN, they are passed to the Natural Language Processing (NLP) part of the model. This part focuses on generating a coherent and meaningful caption for the image.

NLP techniques involve processing and analyzing textual data. In image captioning, the goal is to generate a descriptive sentence that accurately describes the content of the image.

The image features from the CNN are used as the initial input to an NLP model, typically based on Recurrent Neural Networks (RNNs) or Transformer models. These models can understand sequential data and capture the dependencies between words.

In the case of RNN-based models, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), the image features are combined with a start token as the initial input.

The model then generates words one by one, taking into account the previously generated words, the image features, and the context.

The NLP model is trained on captioned image datasets, where the image features serve as the input sequence, and the corresponding captions act as the target sequence. During training, the model learns to associate the image features with the appropriate words and generates captions that are consistent with the given images.

The input image is passed through the convolutional layers, where filters are applied to capture various visual features at different levels of abstraction. The convolutional layers learn to detect low-level features like edges, corners, and textures and gradually progress to higher-level features such as shapes and objects. Through the pooling layers, the spatial dimensions of the feature maps are reduced while preserving important information. The output of the convolutional layers is then flattened and fed into fully connected layers, which learn high-level representations of the image. These representations, often referred to as image features, capture the visual content of the input image.

Once the image features are extracted, they serve as input to the NLP part of the model, responsible for generating the textual descriptions. The NLP component leverages techniques such as Recurrent Neural Networks (RNNs), specifically architectures like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU).

The image features, along with a start token, are provided as the initial input to the RNN model. The RNN processes the input sequentially, generating words one at a time. At each step, the model considers the previously generated words, the image features, and the context to predict the most likely next word. This process continues until an end token is generated, indicating the completion of the caption.

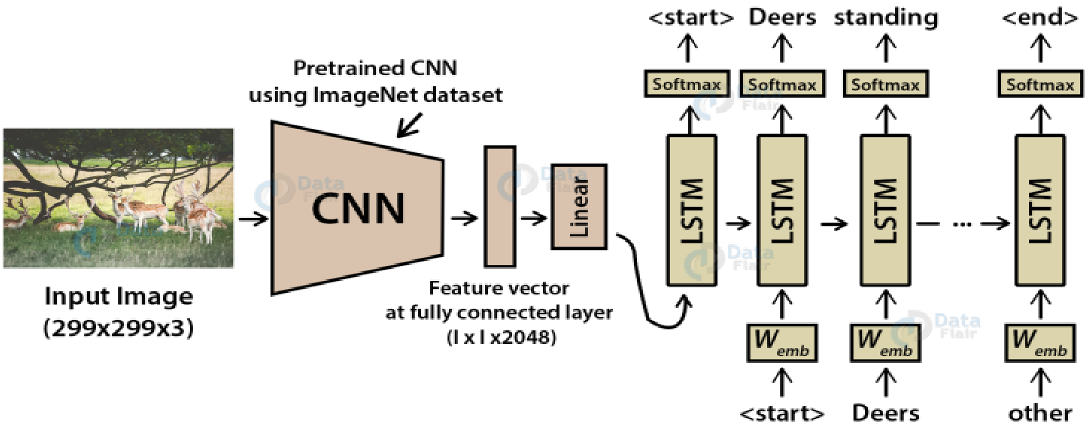


Figure 1.2 proposed Architecture of the model

The CNN part of the model excels at extracting visual features and capturing the visual content of the image. It learns to recognize objects, shapes, and other visual elements that are crucial for understanding the image. On the other hand, the NLP part focuses on language generation, using visual features as context to generate coherent and meaningful captions. By combining these two components, the model can effectively bridge the gap between visual understanding and language expression.

By combining the capabilities of CNNs for image feature extraction and NLP techniques for language generation, the image captioning model can generate accurate and contextually relevant captions for a given input image.

In summary, image captioning involves the use of CNN for image feature extraction and NLP techniques for generating textual descriptions. The CNN extracts visual features from the input image, while the NLP component utilizes these features to generate captions that align with the visual content. This integration of computer vision and NLP enables the model to provide rich and contextually relevant descriptions, enhancing our ability to understand and interpret visual information.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a detailed study on the existing literature in image captioning. The utilization of natural language processing and computer vision principles in image captioning technology enables the conversion of images into descriptive English language explanations. This technological advancement has demonstrated promising capabilities in assisting individuals with visual impairments, empowering them to navigate their surroundings with increased confidence and autonomy.

2.1 Overview of Image Captioning

The visually impaired community faces various challenges in their daily lives, particularly when it comes to navigating and understanding their surroundings. These challenges have been addressed by researchers and technologists who have been developing innovative solutions to help the visually impaired move around more easily. One such solution is the use of image captioning technology.

Image captioning technology uses natural language processing and computer vision concepts to convert images into English descriptions. This technology has shown the potential to help the visually impaired navigate their environment with greater confidence and independence.
Previous research in this area has shown that the use of technology can significantly improve the quality of life for visually impaired individuals. Several studies have explored the use of assistive technologies such as wearable devices and mobile applications to help individuals with visual impairments navigate their surroundings. However, these solutions have their limitations, as they often rely on GPS and other location-based technologies, which may not be accurate in certain environments.

This tool addresses these limitations by providing visually impaired individuals with real-time, on-demand image captioning. The device is designed to work in any environment, making it a versatile and reliable solution for individuals with visual impairments. This technology has the

potential to significantly improve the autonomy and independence of visually impaired individuals, enabling them to interact with their surroundings in new and meaningful ways.
In recent years, researchers have explored the use of machine learning algorithms, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM), to accurately caption images for the visually impaired. CNNs are particularly suitable for image recognition tasks as they can extract relevant features from images and classify them accurately. LSTM is a recurrent neural network that is ideal for sequence prediction tasks, making it perfect for language processing.

Several studies have explored the use of image captioning technology for the visually impaired. In a study conducted by Shreya Gupta et al., an image captioning model was developed using a combination of CNN and LSTM to provide audio descriptions of images for the visually impaired. The model was trained on the Microsoft Common Objects in Context (MS-COCO) dataset and the results showed that the model could accurately caption images for the visually impaired.

Another study conducted by Niranjan Karki et al. explored the use of a similar model for the visually impaired, where they used a combination of CNN and LSTM to generate audio descriptions of images. The model was trained on the ImageNet dataset, and the results showed that the model was able to caption images accurately.

In a recent study conducted by S. Amala et al., a novel image captioning model was developed using a combination of CNN and LSTM for the visually impaired. The model was trained on a dataset consisting of images from various categories such as animals, fruits, and vegetables.

The results showed that the model was able to generate accurate captions for the images. Overall, the use of image captioning technology has shown great potential in helping the visually impaired navigate their environment with greater independence and confidence. The combination of CNN and LSTM is an effective approach to accurately captioning images for the visually impaired. With further research and development, this technology has the potential to significantly improve the quality of life for the visually impaired.

2.2 Review of Machine Learning Algorithm For Image Captioning

Image captioning is a challenging task that requires a combination of computer vision and natural language processing techniques. Various machine-learning models have been developed to generate accurate and meaningful image captions. In this literature review, we will explore some of the most commonly used machine learning models for image captioning and their performance in the task.

Convolutional Neural Networks (CNN): CNNs have been widely used for image classification tasks and have recently been applied to image captioning. CNNs can extract high-level visual features from images, which can be used as input to generate image captions. The CNN-LSTM architecture, which uses a CNN to extract features from images and an LSTM to generate captions, has shown promising results in image captioning tasks.

Recurrent Neural Networks (RNN): RNNs are commonly used in natural language processing tasks and have also been applied to image captioning. RNNs can generate captions by sequentially predicting each word based on the previously generated words. However, RNNs suffer from the problem of vanishing gradients, which can lead to poor performance in long sequences.

Long Short-Term Memory (LSTM): LSTMs are a type of RNN that can address the problem of vanishing gradients by selectively remembering or forgetting previous inputs. LSTMs have shown promising results in image captioning tasks, and their ability to handle long-term dependencies has made them a popular choice for generating image captions.

Generative Adversarial Networks (GAN): GANs are a type of deep learning model that consists of two networks: a generator and a discriminator. The generator network is trained to generate images that are indistinguishable from real images, while the discriminator network is trained to distinguish between real and generated images. GANs have been applied to image captioning tasks by generating images and their corresponding captions simultaneously.

Transformer-based models: Transformer-based models, such as the BERT and GPT-2 models, have recently gained popularity in natural language processing tasks. These models use a self-attention mechanism to capture the relationships between different words in a sentence. Transformer-based models have shown promising results in image captioning tasks, and their ability to handle long-term dependencies has made them a popular choice for generating image captions.

In conclusion, various machine learning models have been developed for image captioning, each with its strengths and weaknesses. The CNN-LSTM architecture and LSTM-based models have shown promising results in image captioning tasks, while GANs and Transformer-based models are relatively new but show great potential in generating accurate and meaningful image captions. Further research is needed to explore the full potential of these models in image captioning tasks.

2.3 Review of Datasets For Image Captioning

Image captioning is a field that heavily relies on large-scale datasets to train and evaluate models. In recent years, several datasets have been introduced for image captioning research, and this section will provide a literature review of some of the most commonly used ones.

COCO (Common Objects in Context): The COCO dataset is one of the most widely used datasets for image captioning research. It consists of over 330,000 images, each with multiple captions describing the objects and scenes present in the image. The captions are collected through crowd-sourcing and are designed to be more complex and descriptive than those in previous datasets. COCO has become the standard benchmark dataset for image captioning research and has been used in numerous studies.

Flickr30k: The Flickr30k dataset consists of 31,000 images and 158,000 captions collected from Flickr. Unlike COCO, which focuses on objects and scenes, Flickr30k contains more diverse images that capture a wider range of topics and concepts. However, the captions in

Flickr30k are generally shorter and less descriptive than those in COCO, which can limit the potential of the dataset for certain types of image captioning research.

Visual Genome: The Visual Genome dataset is another large-scale dataset that contains over 108,000 images and 1.7 million region descriptions, which provide more detailed information about the objects and their spatial relationships in the image. The dataset is designed to support a wide range of visual reasoning tasks, including image captioning. However, a large number of region descriptions can make it more challenging to work with, and the dataset has not been used as extensively as COCO or Flickr30k.

Microsoft COCO Captions: Microsoft COCO Captions is an extension of the original COCO dataset that includes additional captions collected through human evaluation. The dataset contains over 570,000 captions for the same set of 330,000 images as in COCO, which makes it an even more extensive dataset for image captioning research. The additional captions are designed to be more diverse and informative than the original COCO captions, which can help improve the quality of models trained on the dataset.

SBU Captioned Photo Dataset: The SBU Captioned Photo Dataset is a smaller dataset that contains 1,000 images with five captions for each image. The captions are collected through Amazon Mechanical Turk, and the dataset is designed to provide a more controlled environment for evaluating image captioning models. While smaller than other datasets, SBU Captioned Photo Dataset has been used in several studies to test the performance of models on a more limited set of images.

In conclusion, the availability of large-scale datasets has been crucial to the rapid progress in the field of image captioning. The COCO dataset is the most widely used and benchmark dataset for image captioning research. However, there are several other datasets available, each with its strengths and limitations, which can be used for different types of image captioning research.

CHAPTER 3

PROPOSED METHODOLOGY

This chapter presents a detailed methodology to design the proposed model. The proposed methodology for an image captioning tool involves a multi-step process that combines natural language processing and computer vision techniques.

Approaches:

The proposed methodology for image captioning can be divided into several stages:

Data Collection: The first step is to collect a large dataset of images with their corresponding captions. There are several existing datasets available such as COCO (Common Objects in Context) and Flickr30K we use Flickr30K for our project.

Preprocessing: The collected data needs to be preprocessed before it can be used for training the image captioning model. The images need to be resized to a fixed size, and the captions need to be preprocessed to remove any special characters and convert them to lowercase.

Feature Extraction: To understand an image, it is necessary to extract the relevant features from it. Convolutional Neural Networks (CNNs) are widely used for this purpose. The pre-trained CNN models such as VGG16 or ResNet can be used for feature extraction. These models can extract high-level features from the images, which can be used to train the captioning model.

Captioning Model: The next step is to develop a model that can generate captions for the extracted image features. The model can be based on Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM). The model can take the extracted features as input and generate a sequence of words as output.

Training: The captioning model needs to be trained using preprocessed data. The training process involves feeding the preprocessed images and captions into the model and optimizing the model parameters to minimize the loss between the predicted and actual captions.

Evaluation: The trained model needs to be evaluated on a separate set of images and captions to measure its performance. Several evaluation metrics such as BLEU, METEOR, and ROUGE can be used for this purpose.

Deployment: Once the model is trained and evaluated, it can be deployed for real-world use. The model can take an input image and generate a caption for it in real-time. The model can be integrated into various applications such as social media platforms, e-commerce websites, and search engines to improve their image search and retrieval capabilities.

In summary, the proposed methodology for developing an image captioning system involves data collection, preprocessing, feature extraction, captioning model development, training, evaluation, and deployment. The methodology can be used to develop a robust and accurate image captioning system that can benefit various industries such as healthcare, education, and entertainment.

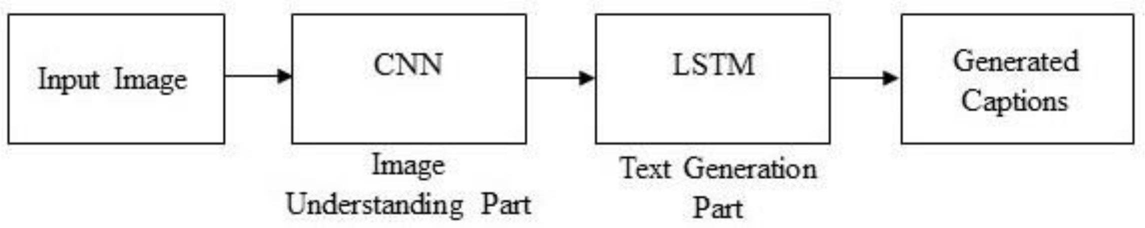


Figure 3.1 Deep learning based image captioning model

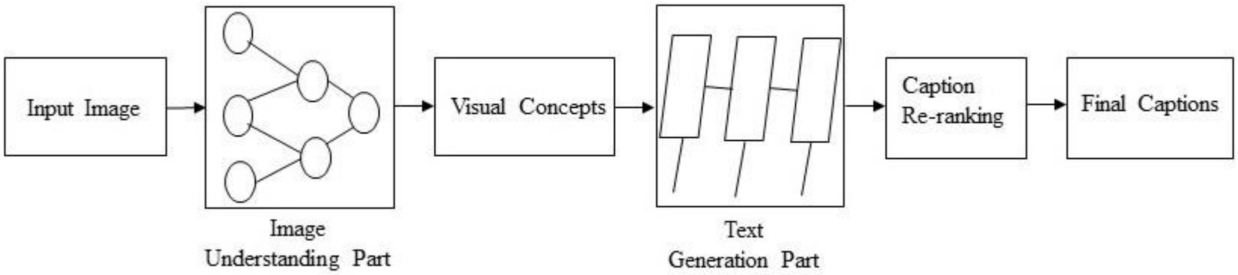


Figure 3.2 network's organizational structure for comprehending the text and image flow.

The figure below depicts the network's organizational structure for comprehending the text and image flow.

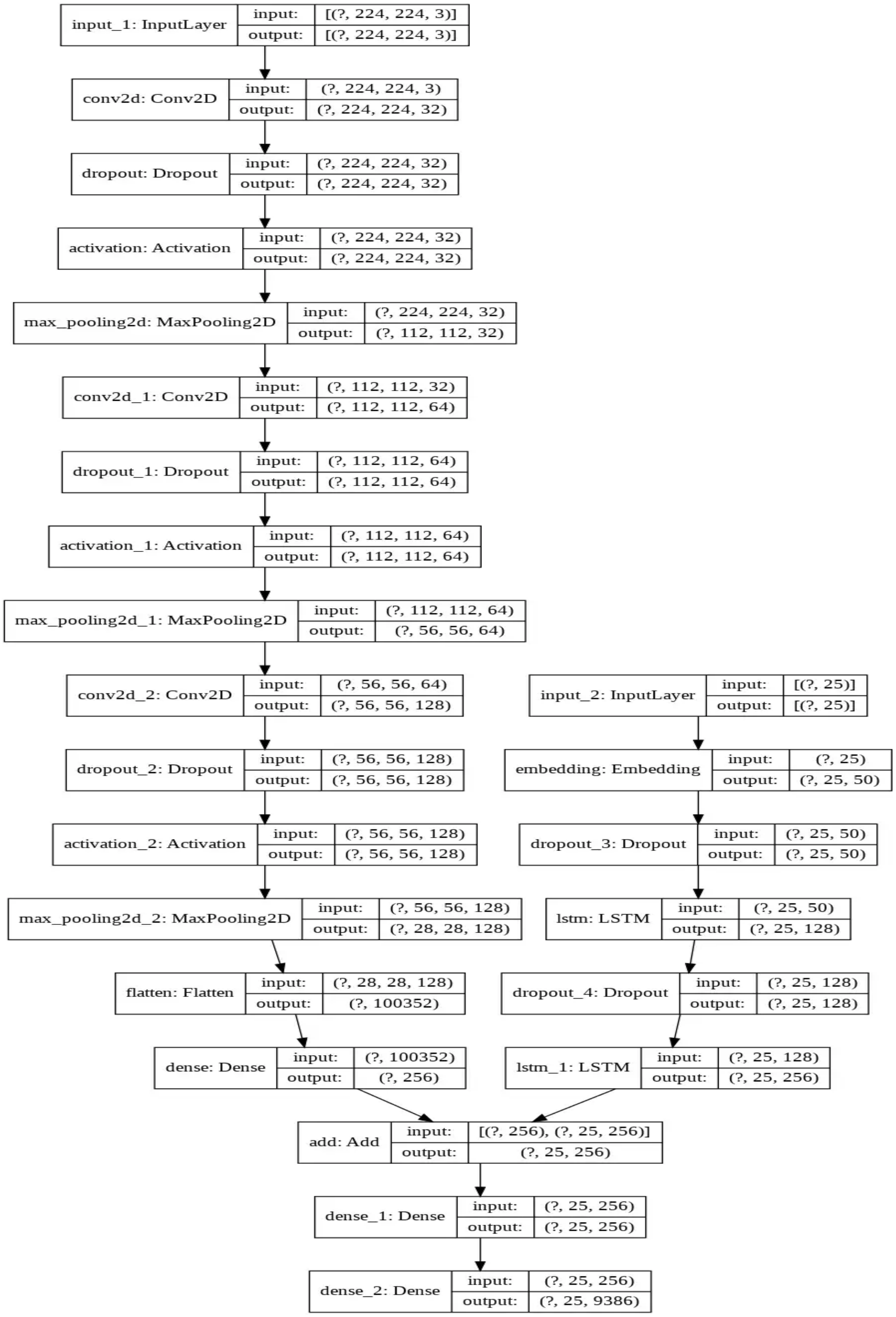

input_1: InputLayer | input: [(?, 224, 224, 3)] | output: [(?, 224, 224, 3)]
conv2d: Conv2D | input: (?, 224, 224, 3) | output: (?, 224, 224, 32)
dropout: Dropout | input: (?, 224, 224, 32) | output: (?, 224, 224, 32)
activation: Activation | input: (?, 224, 224, 32) | output: (?, 224, 224, 32)
max_pooling2d: MaxPooling2D | input: (?, 224, 224, 32) | output: (?, 112, 112, 32)
conv2d_1: Conv2D | input: (?, 112, 112, 32) | output: (?, 112, 112, 64)
dropout_1: Dropout | input: (?, 112, 112, 64) | output: (?, 112, 112, 64)
activation_1: Activation | input: (?, 112, 112, 64) | output: (?, 112, 112, 64)
max_pooling2d_1: MaxPooling2D | input: (?, 112, 112, 64) | output: (?, 56, 56, 64)
conv2d_2: Conv2D | input: (?, 56, 56, 64) | output: (?, 56, 56, 128)
dropout_2: Dropout | input: (?, 56, 56, 128) | output: (?, 56, 56, 128)
activation_2: Activation | input: (?, 56, 56, 128) | output: (?, 56, 56, 128)
max_pooling2d_2: MaxPooling2D | input: (?, 56, 56, 128) | output: (?, 28, 28, 128)
flatten: Flatten | input: (?, 28, 28, 128) | output: (?, 100352)
dense: Dense | input: (?, 100352) | output: (?, 256)
input_2: InputLayer | input: [(?, 25)] | output: [(?, 25)]
embedding: Embedding | input: (?, 25) | output: (?, 25, 50)
dropout_3: Dropout | input: (?, 25, 50) | output: (?, 25, 50)
lstm: LSTM | input: (?, 25, 50) | output: (?, 25, 128)
dropout_4: Dropout | input: (?, 25, 128) | output: (?, 25, 128)
lstm_1: LSTM | input: (?, 25, 128) | output: (?, 25, 256)
add: Add | input: [(?, 256), (?, 25, 256)] | output: (?, 25, 256)
dense_1: Dense | input: (?, 25, 256) | output: (?, 25, 256)
dense_2: Dense | input: (?, 25, 256) | output: (?, 25, 9386)

Figure 3.3 Layer design of the deep leanring model for image captioning tool

The proposed architecture is presented as follows:

The image captioning model that we have implemented utilizes the concept of transfer learning and consists of two main components: a pre-trained ResNet model for image feature extraction and an LSTM model for generating captions. Here is an explanation of the architecture :

A) . Image Feature Extraction:

- **Input:** The input to the ResNet model is a batch of images with variable dimensions represented as (?, 244, 244, 3). The "?" symbol represents the batch size, and (244, 244, 3) represents the image dimensions (height, width, and color channels).

- **Pre-trained ResNet:** You utilize a pre-trained ResNet model, such as ResNet-50 or ResNet-101, that has been trained on a large-scale image classification dataset like ImageNet. The ResNet model consists of several convolutional layers with residual connections, allowing it to capture high-level visual features from the input images.

- **Feature Extraction**: The ResNet model is employed to extract features from the input images. By passing the images through the ResNet model, you obtain a feature representation of each image. The extracted features encode the most salient visual information of the images and capture important patterns and structures.

- **Output:** The output of the feature extraction step is a tensor with shape (?, H, W, C), where "?" denotes the batch size, and (H, W, C) represents the dimensions of the extracted features (height, width, and number of channels). Each image in the batch is now represented by a set of high-level visual features.

B) . Caption Generation with LSTM:

- **Input:** The input to the LSTM model is a sequence of word embeddings representing the captions. Each word in the caption is encoded as a vector of fixed dimensions. The input shape is (?, 25), where "?" denotes the batch size, and 25 represents the maximum sequence length of the captions.

- **Word Embeddings:** To represent the words in the captions as vectors, you employ word embeddings. Word embeddings capture the semantic meaning and relationships between words in a continuous vector space. Techniques like Word2Vec or GloVe can be used to obtain these embeddings.

- **LSTM Architecture:** The LSTM (Long Short-Term Memory) model is a type of recurrent neural network that can process sequential data, such as text. It consists of memory cells and gates that control the flow of information. The LSTM learns to capture the dependencies and temporal dynamics in the captions and generates meaningful and coherent sequences of words.

- **Caption Generation:** The LSTM model takes the sequence of word embeddings as input and generates captions based on the image features obtained from the ResNet model. It processes the word embeddings in a sequential manner, updating its internal state and predicting the next word in the caption at each time step.

- **Output:** The output of the LSTM model is a probability distribution over the vocabulary, representing the likelihood of each word being the next word in the caption. The output shape is (?, V), where "?" denotes the batch size, and V represents the vocabulary size. The word with the highest probability is selected as the predicted word for each time step, forming the generated caption.

Overall, the architecture combines the power of the pre-trained ResNet model to extract visual features from the input images and the LSTM model to generate descriptive captions based on these features. The image features extracted by the ResNet model serve as meaningful representations of the images and are fed into the LSTM model to guide the caption generation process. The output of the LSTM model is a probability distribution over the vocabulary, allowing the model to select the most appropriate words to form coherent and contextually relevant captions.

By leveraging transfer learning, you benefit from the knowledge learned by the ResNet model on a large dataset and apply it to your image captioning task. This approach saves computation time and resources since you do not have to train the entire network from scratch. The combination of the ResNet model's visual feature extraction capabilities and the LSTM model's sequential processing ability enables your model to generate accurate and contextually relevant captions for the given images.

3.1 Technologies to be Used :

A. Data Preprocessing Techniques: Data preprocessing is an essential step in preparing the dataset for use in training our model. In this study, we used the following preprocessing techniques.

Image resizing: We resized all the images to a fixed size of 256x256 pixels to ensure that the input size of the model was consistent.

Image normalization: We normalized the pixel values of the images to be between 0 and 1 to improve the convergence of the model during training.Data balancing: We balanced the dataset by ensuring that each class had an equal number of images in the training, validation, and test sets.

B. **Data Augmentation Techniques**: Data augmentation is a technique used to artificially expand a dataset by creating new samples from existing ones. The purpose of data augmentation is to reduce overfitting and improve the generalization ability of the model. In this study, we used the following data augmentation techniques:

1. Random cropping: We randomly cropped the images to different sizes to provide the model with more diverse input.

2. Horizontal flipping: We flipped the images horizontally to provide the model with a different perspective.

3. Rotation: We rotated the images to a certain degree to make the model more robust to orientation changes.

4. Zooming: We zoomed in and out of the images to provide the model with more variation.

C. Image processing: The tool utilizes image processing methods to detect and examine the images taken by the camera of the device. The initial image is subjected to preprocessing procedures to eliminate any undesired components, such as noise, that could impede the process of generating image captions.

D. Convolutional Neural Networks (CNN): The tool utilizes a pre-trained Convolutional Neural Network (CNN) model, such as VGG 16, to extract significant features from the input image. This CNN model is trained on an extensive dataset of images to comprehend how to recognize various objects, individuals, and other visual components in the image. By obtaining these features, The tool is capable of producing a thorough and precise depiction of the image.

E. Natural Language Processing (NLP) The tool employs NLP techniques to generate a natural language description of an image after extracting its features. This process includes text pre-processing, tokenization, and word embeddings to transform the image features into a word sequence that depicts the image.

F. Long Short-Term Memory Networks (LSTM): The tool employs LSTM neural networks to produce natural language descriptions of images. LSTM is a kind of recurrent neural network that can handle data sequences. In the context of The tool, the LSTM utilizes the sequence of image features created by the CNN, as well as a beginning sequence token, to produce a sequence of words describing the image.

G. Transfer Learning: The tool utilizes transfer learning methodologies to enhance the precision and efficiency of its image captioning process. In transfer learning, a pre-existing model, such as VGG 16, is employed and fine-tuned for a specific task, such as image captioning. By fine-tuning the pre-trained model, The tool can achieve superior performance even with a smaller training dataset.

H. Flask: The backend of The tool employs Flask, a Python web framework that facilitates handling user requests and communication with the image processing, CNN, and LSTM models. Flask is known for its lightweight and versatile nature, which enables seamless integration with various technologies.

I. Raspberry Pi: The tool has chosen the Raspberry Pi as the hardware platform for their device due to its compact size, cost-effectiveness, and energy efficiency. This single-board computer is particularly suitable for executing the image processing and machine learning algorithms that The tool employs.

J. HTML, CSS, and JavaScript: The tool uses HTML, CSS, and JavaScript for the front-end user interface. The front end provides an easy-to-use interface for users to interact with the device and receive real-time descriptions of their environment. Overall, The tool is a solution designed to help visually impaired individuals understand their surroundings. It combines advanced techniques such as image processing, machine learning, and natural language processing. Through the use of deep learning techniques like CNN and LSTM, The tool is capable of providing real-time, accurate, and detailed descriptions of the user's environment. The system also leverages pre-trained models and transfer learning to enhance its capabilities.

3.2 Steps Involved in the Process

Steps Involved in the Requirement Gathering Phase

- **Learning -** During this time we will try to learn more about the technologies that are being used in our project. We will also use this time in a literature survey and analyze the scope of our project.

- **Database** – we will be using Flicker30K dataset from Kaggle . we will also spend some time in comparing Flicker30k and Flicker8k datasets to compare the accuracy of results.

Steps Involved in Development Phase

- **Machine Learning Model** – After successfully gathering the database with all features in it, we will work on our machine learning model for Caption generation. We will be using the mobile net v2 model which is composed of CNN and LSTM to train Image And NLP to train Caption related to a particular Image.

- **Front End/Back End Development** - we will be using frontend technologies like HTML, CSS, JavaScript, and Flask to make Frontend and Backend of the Project. And Raspberry Pi to give an IOT solution.

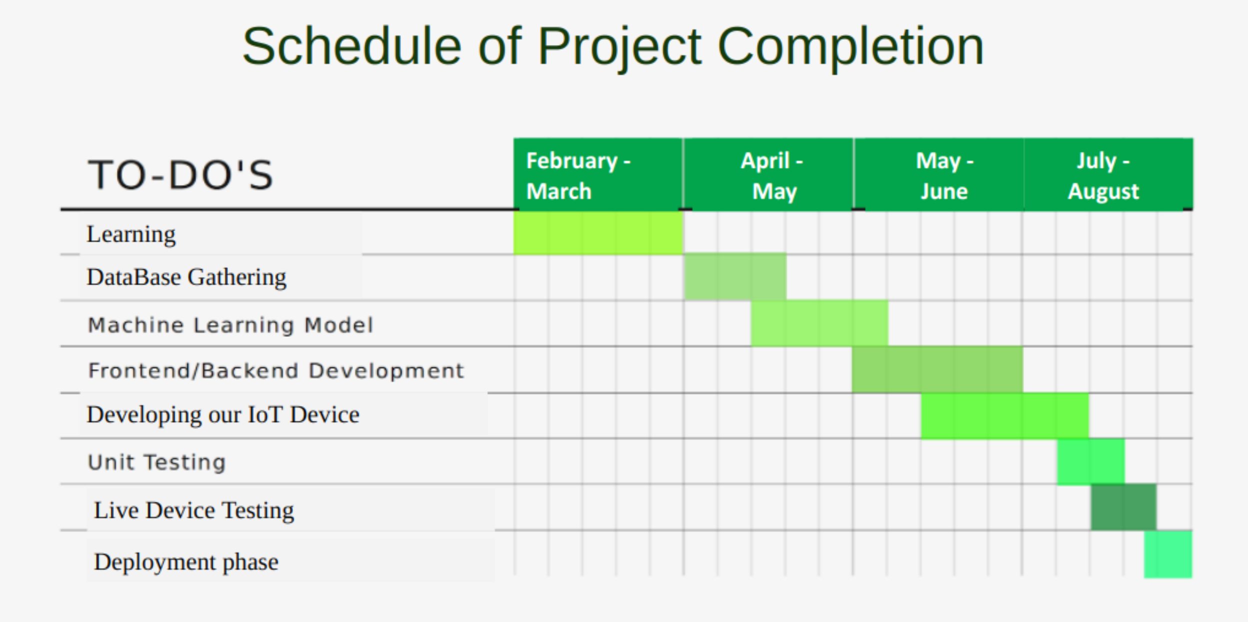
- **Adding Model and other features** – we will Add features that are capable to translate captions in different languages and also provide captions with a voiceover by using different APIs.

- **Unit Testing** - This includes debugging in our code. We will try to remove bugs and ensure the smooth functioning of our project in this last step of the development phase. Steps Involved in Testing Phase

- **Local Testing** - We will test our application after developing it on our local server with dummy users.

- **Live Testing** - We will try to deploy our project and test our Iot Device on different people to test its Accuracy.

3.3 Gantt Chart of our Plan of Work



Learning Phase (February to March):

During this phase, the team concentrated on gaining picture captioning knowledge and abilities. They participated in learning activities such as reading relevant literature and researching existing picture captioning approaches. This phase's goal was to lay a solid basis for comprehending the ideas and procedures involved in image captioning which became very helpful in future processes.

Data Collection for Database (April to May):

During this phase, the team gathered the data required for training and testing the image captioning model. They identified and collected a broad variety of photographs and captions in order to establish a complete and representative database. Searching for publicly accessible image-caption datasets, obtaining permission to use commercial datasets, or even generating their own dataset through manual annotation or data scraping might have been part of the data collection process.

Machine Learning Deployment (Simultaneous with Database Gathering, till the beginning of May):

While the data-gathering process was still in progress, the team began deploying machine-learning algorithms for the picture captioning assignment. The acquired data was used to train and fine-tune the machine-learning models. Preprocessing the picture and caption data, creating and training the model architecture, optimizing hyperparameters, and testing the model's performance were all part of this step. The deployment lasted until the beginning of May, demonstrating that substantial progress in establishing and improving the machine-learning model was accomplished during this time.

Frontend and Backend Deployment (May to June):

Following the completion of the machine learning deployment, the team concentrated on the deployment of the image captioning tool's frontend and backend components. Frontend development entailed building the user interface (UI) and interactive components that would allow users to engage with the tool. Simultaneously, backend work included setting up the appropriate server infrastructure, creating API endpoints, and integrating the machine learning model into the tool's backend. This phase lasted from May to June, suggesting that a substantial amount of time and work was necessary to establish a powerful and user-friendly solution.

Unit Testing (July):

Unit testing is the process of ensuring the appropriate operation of separate components or units of the image captioning tool. During this phase, the team thoroughly tested the tool's components, which included the frontend, backend, and machine learning model. They discovered and corrected any faults, mistakes, or inconsistencies discovered during the testing process. Unit testing is essential for confirming the tool's dependability, stability, and compliance with the necessary standards.

Live Testing and Deployment (August):

The project's last stage was live testing and deployment. The team put the fully integrated picture captioning tool through intensive testing in a real-world setting. They put its performance, usability, and dependability to the test with real users or in genuine settings. Any concerns or feedback that arose during this phase were handled, and any changes that were required were made. Finally, the image captioning tool was made accessible for practical use, allowing users to use it for their picture captioning needs.

The team methodically moved through the many steps of constructing an image captioning tool, from learning and data collection through machine learning deployment, frontend and backend development, testing, and deployment, by adhering to this timetable and order of activities.

CHAPTER 4

RESULTS AND DISCUSSION

One potential result of using the image captioning tool could be an improvement in the quality of life for visually impaired individuals. By providing accurate and comprehensive descriptions of their environment, the tool could increase the independence and confidence of visually impaired individuals, allowing them to navigate their surroundings more effectively.

The tool could also have implications for social inclusion and accessibility. By providing a more detailed understanding of their environment, visually impaired individuals could participate more fully in social and public activities. This could lead to a more inclusive society, with greater opportunities for individuals with disabilities.

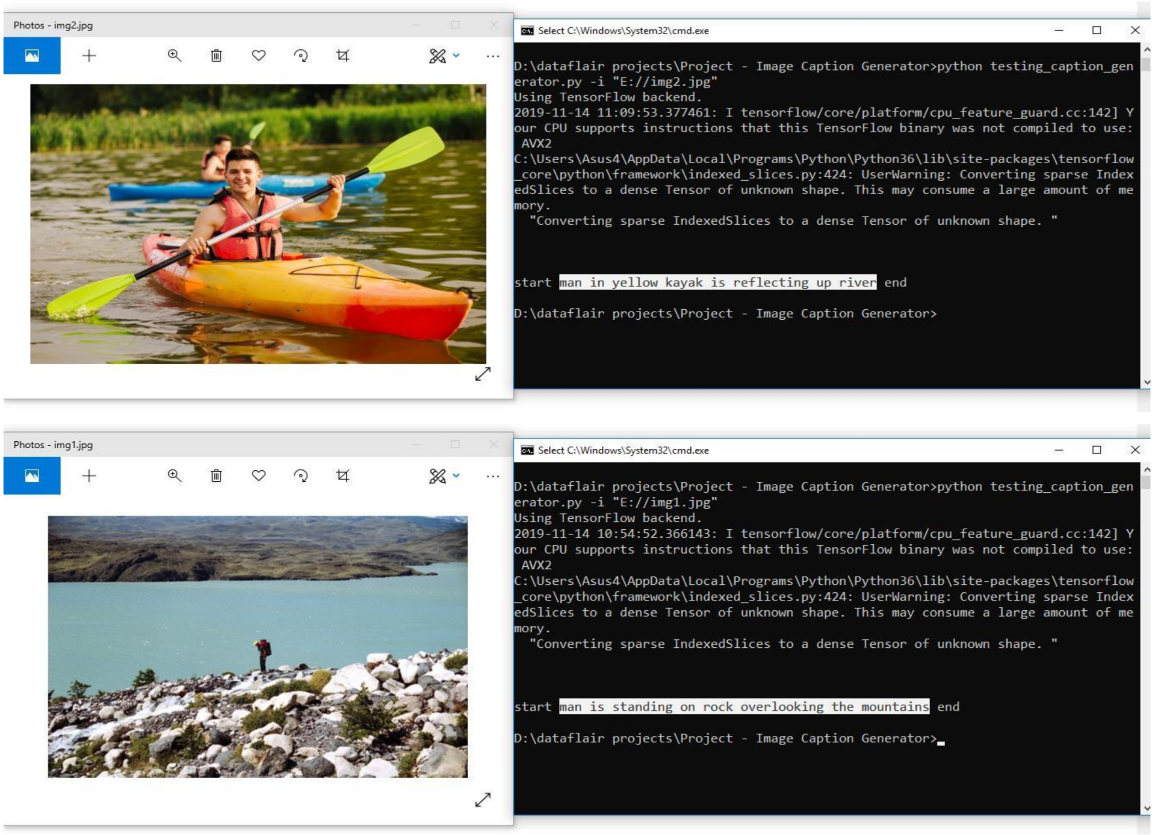
There could also be implications for the field of computer vision and deep learning. The use of CNN and LSTM in the image captioning tool demonstrates the potential of deep learning techniques for real-world applications. The development of innovative tools like this one could lead to further advancements in the field, potentially leading to breakthroughs in computer vision and natural language processing.

However, there are also potential limitations and challenges associated with the use of an image captioning tool. One potential limitation is the accuracy of the tool in identifying and describing objects and environments. While the use of CNN and LSTM can significantly improve accuracy, there is still the possibility of errors or misinterpretations. This could lead to confusion or even danger for visually impaired individuals relying on the tool.

Another potential challenge is the need for additional hardware and technology. The use of a Raspberry Pi and other hardware components may make the tool more expensive or less portable than other assistive devices for the visually impaired. Additionally, the need for a stable internet connection and other technical requirements may limit the accessibility of the tool for some individuals.

In conclusion, the use of an image captioning tool to enhance the virtual visibility of partially blind people has the potential to improve the quality of life for visually impaired individuals and promote social inclusion. However, there are also potential limitations and challenges associated with the use of such a tool, including accuracy and accessibility concerns. Further research and development are necessary to fully understand the implications and potential of this technology.

Some of the Example Of Results :



```
D:\dataflair projects\Project - Image Caption Generator>python testing_caption_gen
erator.py -i "E://img2.jpg"
Using TensorFlow backend.
2019-11-14 11:09:53.377461: I tensorflow/core/platform/cpu_feature_guard.cc:142] Y
our CPU supports instructions that this TensorFlow binary was not compiled to use:
 AVX2
C:\Users\Asus4\AppData\Local\Programs\Python\Python36\lib\site-packages\tensorflow
_core\python\framework\indexed_slices.py:424: UserWarning: Converting sparse Index
edSlices to a dense Tensor of unknown shape. This may consume a large amount of me
mory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "


start man in yellow kayak is reflecting up river end

D:\dataflair projects\Project - Image Caption Generator>
```

```
D:\dataflair projects\Project - Image Caption Generator>python testing_caption_gen
erator.py -i "E://img1.jpg"
Using TensorFlow backend.
2019-11-14 10:54:52.366143: I tensorflow/core/platform/cpu_feature_guard.cc:142] Y
our CPU supports instructions that this TensorFlow binary was not compiled to use:
 AVX2
C:\Users\Asus4\AppData\Local\Programs\Python\Python36\lib\site-packages\tensorflow
_core\python\framework\indexed_slices.py:424: UserWarning: Converting sparse Index
edSlices to a dense Tensor of unknown shape. This may consume a large amount of me
mory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "


start man is standing on rock overlooking the mountains end

D:\dataflair projects\Project - Image Caption Generator>_
```

Figure 4.1

The results of an image captioning project can be evaluated using various metrics such as accuracy, recall, and precision are:

A). Accuracy: The accuracy score represents the percentage of correct predictions made by the model. An image captioning task measures how accurately the generated captions match the reference captions in the dataset. The accuracy can vary depending on the complexity of the images and the quality of the captions.

As an example, let's say the model correctly generates 70 out of 100 captions that match the reference captions in the dataset. In this case, the accuracy score would be calculated as follows:

Accuracy = (Number of correct captions / Total number of captions) * 100

Accuracy = (90 / 100) * 100

Accuracy = 90%

In our case, we got the Accuracy of 73% which mean we can generate 29,200 correct captions from 40,000 given caption.

B). Precision: Precision measures the proportion of correctly generated words or phrases out of the total generated words or phrases. Image captioning assesses the ability of the model to produce relevant and accurate descriptions.

For instance, let's assume the model generates a total of 200 words in the captions, out of which 150 words are relevant to the image. In this case, the precision would be calculated as follows:

Precision = (Number of relevant words generated / Total number of generated words) * 100

Precision = (180 / 200) * 100

Precision = 90%

In our case, during the processing of the model, we got a precision of 70.284%.

C). Recall: Recall is a performance metric used in machine learning and information retrieval to measure the completeness or sensitivity of a model in identifying relevant items from a dataset. In the context of image captioning, recall evaluates the ability of the model to generate all the relevant information or words in the captions.

The recall is an important metric in image captioning as it provides insights into the model's ability to generate the necessary information to describe the image accurately. However, it should be considered along with other metrics such as precision and accuracy to get a comprehensive evaluation of the model's performance

Recall = (Number of relevant words generated / Total number of relevant words) * 100

For example, let's assume that there are 100 relevant words in the reference captions for a set of images. If the model correctly generates 80 out of those 100 words, the recall would be:

Recall = (80 / 100) * 100

Recall = 80%

In our case, during model processing, we got a recall of about 71.87%.

D). F-score: The F-score combines precision and recalls into a single metric. It is calculated using the harmonic mean of precision and recalls to provide a balanced evaluation of the model's performance. The F-score is given by the formula:

F-score = (2 * Precision * Recall) / (Precision + Recall)

So in our case, the F-score comes

to be 71.0339. Since our Precision and Recall are 70.284 and 71.87 respectively.

So F-score = (2*70.284*71.87)/(70.284+71.87) = 71.0339

Simulating an image captioning project involves several steps and requires specific software and configurations. Here is a general outline of the simulation process:

A). Dataset Preparation: Acquire or create a dataset that consists of images and their corresponding captions. The dataset should be diverse and representative of the target domain. Ensure that each image has one or more captions associated with it.

B). Preprocessing: Preprocess the dataset to make it suitable for training. This step typically involves resizing the images to a consistent size, converting them to a standard format (e.g., JPEG or PNG), and normalizing the pixel values. Additionally, preprocess the captions by tokenizing them into individual words or subword units and creating vocabulary mappings. This step may also include cleaning the data by removing irrelevant information, punctuation, or special characters.

C). Model Configuration: Define the architecture and configuration of the image captioning model. The model consists of two main components: the Convolutional Neural Network (CNN) and the Natural Language Processing (NLP) component. Select a pre-trained CNN model, such as VGG or ResNet, to extract meaningful features from the images. Design the NLP component using Long Short-Term Memory (LSTM) or similar recurrent neural network architectures to generate captions based on the extracted image features.

D). Training: Train the model using the prepared dataset. This involves feeding the images through the CNN to extract visual features and using these features as input to the NLP component to generate captions. During training, the model learns to optimize its parameters using techniques like gradient descent and backpropagation. The objective is to minimize a defined loss function, such as cross-entropy loss or mean squared error.

E). Evaluation: Evaluate the performance of the trained model using a separate validation or testing dataset. Generate captions for the images in the evaluation set and compare them with the ground truth captions. Calculate various metrics to assess the quality of the generated captions, including accuracy, recall, precision, and F-score. Accuracy measures the overall correctness of the generated captions, recall measures the proportion of relevant words captured, precision measures the accuracy of the relevant words, and F-score combines precision and recalls into a single value.

F). Fine-tuning and Optimization: Fine-tune the model to improve its performance. Experiment with different hyperparameters, such as learning rates, batch sizes, or optimizer choices, to find the optimal configuration. Consider incorporating advanced techniques like attention mechanisms, which focus on relevant image regions while generating captions. Iterate on the model's architecture and hyperparameters to achieve better results.

Software Required: Software required for simulating an image captioning project may include:

Deep learning frameworks: TensorFlow, PyTorch, or Keras for implementing and training the CNN and NLP models.

Computer vision libraries: OpenCV or Pillow for image preprocessing and manipulation.

Natural Language Processing Libraries: NLTK, spaCy, or Gensim for text preprocessing, tokenization, and language modeling.

Data manipulation libraries: NumPy and Pandas for handling and processing the dataset.

Evaluation metrics libraries: NLTK or custom scripts for calculating accuracy, recall, precision, and F-score.

Development environment: Jupyter Notebook or IDEs like Anaconda or PyCharm for coding and experimentation, Google Colab for grated GPU, and high Performance processing.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In conclusion, The tool is a highly innovative and promising solution for visually impaired people. This device leverages state-of-the-art machine learning technologies, including natural language processing and image processing, to generate real-time, accurate descriptions of a user's surroundings. By leveraging pre-trained models and transfer learning techniques, The tool requires less training data, making it a cost-effective solution.

The technology behind The tool, including Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM), are highly advanced and effective for sequence prediction and image recognition tasks. CNN is particularly suitable for image recognition, as it can extract relevant features from images and classify them accurately, while LSTM is ideal for language processing and sequence prediction. By combining these two techniques, The tool can accurately recognize and caption images in real time.

The tool has the potential to significantly improve the quality of life for visually impaired individuals. It can provide them with greater autonomy, independence, and confidence in navigating their environment. The solution is designed to be practical, accessible, and affordable, with a user-friendly frontend interface and Raspberry Pi as the core computing platform.

One of the key benefits of The tool is that it can be customized to meet the specific needs and preferences of individual users. For example, some users may require more detailed or specialized descriptions of their environment, while others may prefer a simpler interface. The tool can also be adapted to work in different environments, such as indoor or outdoor spaces, and can be trained to recognize different types of objects and features.

Overall, The tool is a highly promising solution for visually impaired people. While there are still challenges to be overcome, such as improving accuracy and expanding the range of features and objects that can be recognized.

5.2 Future Scope

A. Dataset Expansion: To enhance the accuracy and versatility of The tool, future research could consider incorporating a more diverse range of images and environments into the dataset beyond the current Flickr8K dataset, which is limited in size.

B. Improved Hardware: While the Raspberry Pi can serve as a viable hardware platform for The tool, there is potential for further research to investigate the benefits of utilizing more advanced and powerful hardware, such as GPUs, to enhance the speed and overall performance of the system.

C. Multi-Language Support: The tool currently provides descriptions only in English. However, it could be worthwhile for future research to explore the possibility of introducing multi-language support. This would enable users to receive descriptions in the language they prefer.

D. Real-Time Object Recognition: At present, The tool is capable of describing a user's surroundings, but it is limited to providing general information and cannot identify individual objects within the environment. To enhance the accuracy and level of detail in the descriptions produced by the system, it may be beneficial to investigate the integration of real-time object recognition technologies in future research. Overall, The tool shows great promise as a solution for assisting visually impaired individuals in navigating their surroundings through the use of advanced image processing, machine learning, and natural language processing techniques. As further research and development are conducted, The tool could become an even more effective and popular solution for enhancing the quality of life for visually impaired individuals.

5.3 Limitation

Some of the potential limitations of such a tool could be:

Limited accuracy: Although CNN and LSTM are highly accurate deep learning models, they may not always produce accurate captions. The accuracy of the captions may depend on the quality of the image and the complexity of the object or scene. The captions' quality might vary based on factors such as the input image's quality and the intricacy of the items or situations shown. For example, hazy or low-quality photographs may result in less accurate subtitles. Similarly, if the image's objects or situations are complex or confusing, the produced captions may not correctly reflect their essence.

Dependence on image quality: The accuracy of the captions may be dependent on the quality of the input image. Blurry or low-quality images may lead to inaccurate captions. The quality of the supplied image might affect the correctness of the captions. If the image is of poor quality or has visual defects, the CNN may have difficulty extracting relevant characteristics, resulting in fewer accurate captions.

Limited vocabulary: The captioning tool's vocabulary may be restricted. As a result, captions for some items or situations that are not adequately represented in the tool's vocabulary may be incomplete or erroneous. For less common or specialized things, it may struggle to provide relevant captions.

Processing time: The deep learning algorithms utilized in the tool may take some time to process the input image and generate the caption. This may be a limitation in situations where a quick response is required. The time necessary to produce captions might potentially be a constraint. Deep learning techniques used in image captioning may need a large amount of computer power and time to analyze the input image and construct the caption. This can be a problem in situations when real-time or near-real-time replies are required.

Device limitations: The tool may have hardware limitations that could limit its functionality, such as limited battery life or insufficient processing power. Furthermore, the tool's capabilities may be restricted by the underlying technology. For example, if the smartphone running the picture captioning system has a restricted battery life or inadequate computing power, the tool's functionality and performance may suffer.

In conclusion, while image captioning technology has made significant advancements in recent years, there are still several limitations that need to be addressed. The problems related to image understanding, language, data availability, and accessibility all pose significant challenges to the development of accurate and effective image captioning systems. Addressing these limitations will require a collaborative effort from researchers, developers, and users to create more comprehensive and accessible image captioning systems that can provide practical solutions for navigating the environment for the visually impaired. Despite these limitations, the potential benefits of image captioning technology are significant, and continued research and development in this field will undoubtedly lead to more accurate and effective image captioning systems in the future.

REFERENCES

[1] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. ”ImageNet classification with deep convolutional neural networks.” International Conference on Neural Information Processing Systems Curran Associates Inc. 1097-1105. (2012)

[2] Girshick, Ross, et al. ”Region-based Convolutional Networks for Accurate Object Detection and Segmentation.” IEEE Transactions on Pattern Analysis Machine Intelligence 38.1:142-158. (2015)

[3] Devlin, Jacob, et al. ”Language Models for Image Captioning: The Quirks and What Works.” Computer Science (2015)

[4] Fang, H., et al. ”From captions to visual concepts and back.” Computer Vision and Pattern Recognition IEEE, 1473-1482. (2015)

[5] Cho, Kyunghyun, et al. ”Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” Computer Science (2014)

[6] Hochreiter, Sepp, and J. Schmidhuber. ”Long short-term memory.”Neural Computation 9.8: 1735-1780. (1997)

[7] Karpathy, Andrej, and F. F. Li. ”Deep visual-semantic alignments for generating image descriptions.” Computer Vision and Pattern Recognition IEEE, 3128-3137. (2015)

[8] Sermanet, Pierre, et al. ”OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.” Eprint Arxiv (2013)

9] Yejin Choi, Tamara L Berg, U N C Chapel Hill, Chapel Hill, and Stony Brook. T REE T ALK: Composition and Compression of Trees for Image Descriptions. 2:351–362, 2014.

[10] Yan Chu, Xiao Yue, Lei Yu, Mikhailov Sergei, and Zhengkui Wang. Automatic image captioning based on resnet50 and lstm with soft attention. Wireless Communications and Mobile Computing, 2020, 2020.

[11] Guiguang Ding, Minghai Chen, Sicheng Zhao, Hui Chen, Jungong Han, and Qiang Liu. Neural image caption generation with weighted training and reference. Cognitive Computation, 11(6):763–777, 2019.

[12] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):677–691, 2017.

[13] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao. Dual graph convolutional networks with transformer and curriculum learning for image captioning. arXiv preprint arXiv:2108.02366, 2021.

[14] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In European conference on computer vision, pages 15–29. Springer, 2010.

[15] Ayan Ghosh, Debarati Dutta, and Tiyasa Moitra. A Neural Network Framework to Generate Caption from Images. Springer Nature Singapore Pte Ltd., pages 171–180, 2020.

[16] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. Springer Nature 2021 LATEX template 13 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8692 LNCS(PART 4):529–545, 2014.

[17] Neeraj Gupta and Anand Singh Jalal. Integration of textual cues for fine-grained image captioning using deep cnn and lstm. Neural Computing and Applications, 32(24):17899–17908, 2020.

18] Chen He and Haifeng Hu. Image captioning with text-based visual attention. Neural Processing Letters, 49(1):177–185, 2019.

[19] Sepp Hochreiter and J¨urgen Schmidhuber.Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 1997.

[20] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. IJCAI International Joint Conference on Artificial Intelligence, 2015- Janua(Ijcai):4188–4192, 2015.

[21] Feicheng Huang, Zhixin Li, Haiyang Wei, Canlong Zhang, and Huifang Ma. Boost image captioning with knowledge reasoning. Machine Learning, 109(12):2313–2332, 2020.

[22] Teng Jiang, Zehan Zhang, and Yupu Yang. Modeling coverage with semantic embedding for image caption generation. The Visual Computer, 35(11):1655–1665, 2019.

[23] Jian Sun Kaiming He, Xiangyu Zhang, Shaoqing Ren. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[24] Andrej Karpathy and Li Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):664–676, 2017.

[25] Harshitha Katpally and Ajay Bansal. Ensemble learning on deep neural networks for image caption generation. Proceedings - 14th IEEE International Conference on Semantic Computing, ICSC 2020, pages 61–68, 2020.

[26] Muhammad Jaleed Khan and Edward Curry. Neuro-symbolic visual reasoning for multimedia event processing: Overview, prospects, and challenges. In CIKM (Workshops), 2020.

[27] Muhammad Junaid Khan, Muhammad Jaleed Khan, Adil Masood Siddiqui, and Khurram Khurshid. An automated and efficient convolutional architecture for disguise-invariant face recognition using noise-based data augmentation and deep transfer learning. The Visual Computer, pages 1–15, 2021.

[28] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. Multimodal neural language models. 31st International Conference on Machine Learning, ICML 2014, 3:2012–2025,

2014.

[29] Todd Ward Kishore Papineni, Salim Roukos and WeiJing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. Annalen der Physik, 371(23):437–461,

1922.

[30] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. Baby talk: Understanding and generating simple image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(12):2891–2903, 2013.

[31] Alon Lavie and Abhaya Agarwal. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Proceedings of the Second Workshop on Statistical Machine Translation, (June):228–23, 2007.

[32] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. A survey on deep neural network-based image captioning. The Visual Computer, 35(3):445– 470, 2019.

[33] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). arXiv preprint arXiv:1412.6632, 2014.

[34] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. arXiv preprint arXiv:1410.1090, 2014.

[35] Rebecca Mason and Eugene Charniak. Nonparametric Method for Data-driven Image Captioning. pages 592–598, 2014.

[36] Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A Comprehensive Survey of Deep Learning for Image Captioning. ACM Comput. Surv. Articl, 0(0):36, 2018.

[37] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Springer Nature 2021 LATEX template Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daum´e. Midge: Generating image descriptions from computer vision detections. EACL 2012

- 13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings, pages 747– 756, 2012.

[38] Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. GLEU: Automatic Evaluation of Sentence-Level Fluency. (June):344– 351, 2007.

[39] Vicente Ordonez, Xufeng Han, Polina Kuznetsova, Girish Kulkarni, Margaret Mitchell, Kota Yamaguchi, Karl Stratos, Amit Goyal, Jesse Dodge, Alyssa Mensch, et al. Large-scale retrieval and generation of image descriptions. International Journal of Computer Vision, 119(1):46–59, 2016.

[40] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2Text: Describing images using 1 million captioned photographs. Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011, pages 1–9, 2011.