2nd National Level Students' Research Conference on
"Innovative Ideas and Invention with Sustainability in Computer Science and IT-2021"
In association withInternational Journal of Scientific Research in Computer Science, Engineering and Information Technology| ISSN : 2456-3307 (www.ijsrcseit.com)

Audio Assistance for Visually Impaired Using Image Captioning

Krunal Tule, Krishna Patil, Manas Yeole, Shrenik Shingi, Dr. Rashmi Phalnikar*
Dept. of Computer Science and Engineering, MIT-WPU, Pune, Maharashtra, India

ABSTRACT

Blind people navigate safely through a familiar room based on a strong judgement about the location of objects. If something has been moved, added or removed, it can present difficulty and potentially a danger. Human eyes are one of the most important body parts that help humans to understand and interact with their surroundings. Most learning and recognition of objects around us is accomplished using the eyes and their biological capabilities. Given the recent advancement of imaging systems and the ever-increasing processing power of microprocessors, developing audio assistance systems for the visually impaired using image captioning is possible. In the initial system, we propose a system consisting of a camera-equipped microprocessor to capture the images and generate descriptive text out of them. This will ultimately help the visually impaired to perform their day-to-day activity independently.

Keywords : CNN, RNN, Image Captioning, Text-To-Speech, Raspberry Pi

I. INTRODUCTION

Over 40 million people in the world are blind, and over 120 million people have significant low Vision conditions that cannot be corrected, cured or treated by conventional refraction, medicine or surgery. This number is expected to double by the year 2020(World Health Organization, 2004). Most common and current mobility devices for the blind provide information in either tactile or auditory form. Of these the most used are the long cane and the guide dog, the limitations of which include effective range and cost, respectively. Human eyes are one of the most important body parts that help humans to understand and interact with their surroundings. Most learning and recognition of objects around us is accomplished using the eyes. Given the recent advancement of imaging systems and the ever-increasing processing power of microprocessors, a machine vision aiding system for the blind can be a reality. Blind and visually impaired people navigate safely through.

People can extract data from almost everything that surrounds them. For example, when you hear a sound, it can describe it using natural language. When it comes to machines, they cannot do that readily. Humans can make use in that way of every sense they possess. Now imagine if one of those senses is missing. Machine learning applications can help, for instance, deaf people when it comes to written information by using a text-to-speech algorithm to "read" to them.

There is a great need to integrate visually impaired persons into society. This work proposes one method that we believe can go a long way in achieving this objective. If blind people can recognize their surroundings, especially people around them, then we think that their quality of life will improve greatly.

II. LITERATURE REVIEW

Adela Puscasiu in [1] this paper presents a composite model, consisting of a deep convolutional neural network for feature extraction that makes use of transfer learning, and a recurrent neural network for building the descriptions. Due to the lack of efficiency of the device optimum results not achieved.

Varsha Kesavan [2] in this paper has done a comparative study between different pre-trained CNN models like inception v3, vgg16, resnet with and without attention model. A Comparative Study between the models is done. But the Conclusion did not have a proper result.

Faruk Ahmed [3] in this paper, has presented the outcome of the experiment of image captioning systems. The design and implementation of a system embedded in an RPi3 is part of the experiment. This system uses API calls that are network dependent which results in the delay of output.

Cristian Iorga in [4] this paper presents a model of Deep Convolutional Neural Networks (CNN) for image recognition. Use CNN system the large ImageNet dataset of 14 million images and 1000 classes in order to learn feature selection. Dataset Used: UC Merced Landa. They have achieved an accuracy of0.87.

III. PROPOSED WORK

The aim of the project is to make an assistive technology for blind people which would assist them to travel independently and hassle-free. The main objective of our project is to develop a system that will help them to understand and adapt to the changes in their immediate surroundings. The frame is extracted from the video after a certain interval to caption it. This caption is then converted to audio format using the text-to-speech technique. With audio assistance from our device, blind people can visualize the event occurring around them. Although there are some limitations that are pertained, users will have to charge the device regularly, output depends on the quality of the frame that is being used to generate output. To reduce the latency of response in the initial phase device is to be used in offline mode, hence the dataset used in the machine learning model is to be updated in regular intervals. Adding to the above instances, it is assumed that the device is used in ample light so that input video would be of good quality and would enhance the accuracy of output. The placement and position of the camera are crucial for achieving optimum results.

IV. PROPOSED ARCHITECTURE

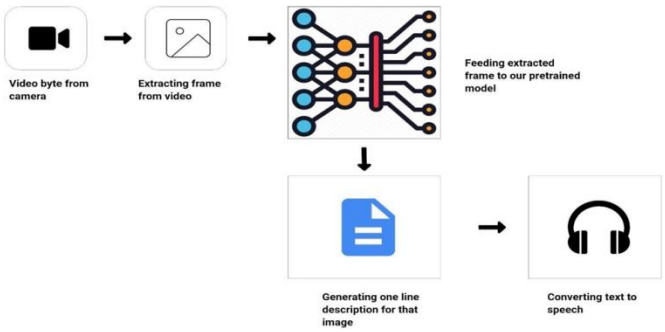

Fig.1

The proposed system includes a wearable device that helps visually impaired people to move around and get their day-to-day tasks done independently like every other person [3]. The wearable device will be a

pouch that contains a Raspberry Pi, a camera module connected to Raspberry Pi and an audio jack that will be worn by the visually impaired user.

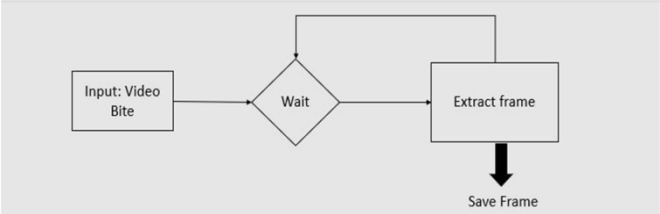

Fig.2

V. PHASES

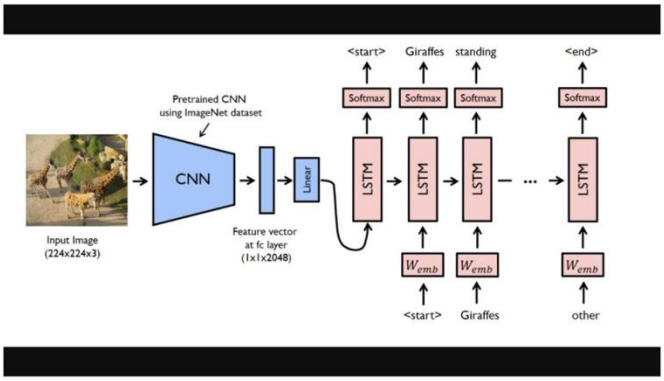

Fig.3

A. The dataset:

The dataset used for training and testing the model is MSCOCO, which stands for "Microsoft- Common Objects in Context" [5]. It was originally made available in 2014, its last revision being in 2015. This dataset is specifically created for usage in the type of problem addressed in this paper. According to its website, it "is large-scale object detection, segmentation, and captioning dataset". It contains more than 80000 labelled images, making it one of the most popular datasets for image-relatedprojects.

What makes it a perfect fit for the problem at hand is that it has five different labels each containing one written description for each image. It is also made use of the "separate" dataset that is represented by all the labels (captions) for training and testing the decoder.

B. The data pre-processing:

Data pre-processing represents a very important step in every machine learning algorithm. Skipping this part results in the model raising an error because it does not receive the expected input. This application requires two different types of data preparation: one for the deep convolutional neural network encoder and one for the deep recurrent neural networkdecoder.

Given that the deep convolutional neural network encoder is based on the Inception- v3 model [2], the images must be resized to the expected format, i.e. (299, 299) and the pixels must be brought in the [-1, 1] range. TensorFlow offers the "image" module for processing images, which allows for them to be read into memory, decoded as jpeg and resized. The application of the Inception-v3 of the Keras high-level API offers the "preprocess_input" method, which normalizes the pixels in the desired range, mentioned above [1]. Data preparation for the language generatordecoder,

i.e. the deep recurrent neural network requires the pre-processing of the textual data, i.e. the captions. For this part, the module "pre-processing" and its methods, offered by Keras, are used.

The steps performed are:

- Caption tokenization, i.e., splitting the captions by white spaces, ending up only with the unique words.
- Vocabulary size limitation: the vocabulary size is limited to the top 5000 words, to save memory. Converting text into a sequence of numbers, Word-index mapping, Padding all the sequences to the length of the longest one. The result is a vector of a sequence of integers, padded to be of the same length with the longest caption that was present in the dataset. This result is depicted in the image below:

```
[ ] sample_caption = torch.Tensor(sample_caption).long()
    print(sample_caption)

    tensor([   0,    3,   98,  754,    3,  396,   39,    3, 1009,  207,  139,    3,
             753,   18,    1])
```

Fig.4

C. The Encoder-Decoder Architecture:

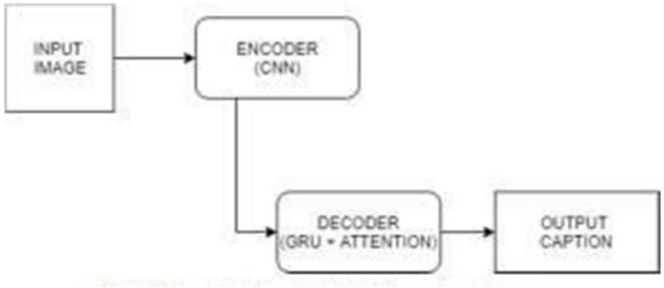

Fig.5

The Encoder-Decoder architecture is widely used in applications such as machine translation, by using two Recurrent Neural Networks. The main idea behind this architecture is to resize the input vector required by the second network. In this case, the first neural network is of convolutional type, while the second is a Recurrent Neural Network. The vector output of the first neural network is the input of the second one. A general schematic of this architecture is showcased below:

The attention mechanism, as described in the paper [4], acts as an interface between the encoder and the decoder. This is needed because without it, the data fed into the decoder would be just one vector representation. The attention mechanism offers information from every hidden state of the encoder, aiding the decoder in focusing on the useful parts.

D. The training phase:

The training phase is self-explanatory. This is how the algorithms are learning to map the function parameters. This is the most complex step, both computationally and programmatically. This consists of backpropagation of the data through the algorithm several times and requires some parameters to be set up and some functions to be chosen.

1) Loading ML model in raspberrypi

After training the model on a GPU enabled machine, we can load that model on raspberry pi directly.

2) Connecting Camera Module to raspberrypi

Fig.6

3) Text-To-Speech

Our paper's main purpose is describing the brief given by the ML model to blind people in speech format. We can convert the text generated by the model to voice by using.

```
pip3 install gTTS pyttsx3 playsound
```

Fig.7

python.

i) Installing requiredlibraries

ii) Open Python file andimport:

```
import gtts
from playsound import playsound
```

Fig.8

iii) It's pretty straightforward to use this library, you just need to pass text to the gTTS object that is an interface to Google Translate's Text-to-Speech API:

```
# make request to google to get synthesis
tts = gtts.gTTS("Hello world")
```

Fig.9

iv) Up to this point, we have sent the text and retrieved the actual audio speech, let us save this audio to a file:

```
# play the audio file
playsound("hello.mp3")
tts.save( hello.mp3 )
```

Fig.10

VI. HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement:

1) Raspberry Pi 4Model:

Raspberry Pi is a cost-effective, credit card sized. Computer that connects to a computer monitor or television. It's quite beneficial for both personal and commercial use.

Fig.11

Model B of the Raspberry Pi 4 features a 1.5GHz quad-core 64bit ARM Cortex- A7 Processor, 1 GB or 2 GB or 4 GB SDRAM, complete Gigabit Ethernet, Bluetooth. of dual-band 802.11ac, two USB 3.0 and two USB 2.0 and supports up to 2 monitors of 4K resolution.

[4] Fig 2 depicts a Raspberry Pi 4 Model B. First needed to configure the Raspberry Pi from the programming side to communication mode and then upload the code which is in the python programming language. The online simulation mode is after the compilation program. The online simulation model is used to check how the program is running step bystep.

2) CameraModule:

Fig.12

The Raspberry Pi requires a camera module to take high-definition videos and images which is later used as inputs for various training and analyzing purposes. It also supports 1080 p30, 720p60 and VGA90 video types and still captures. Capturing the video using the camera module is easy with OpenCV, as it does not require any additional software.

Software Requirement:

- Google Collab: It allows anybody to write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis
- Tensor Flow Lite: It is an open-source deep learning framework for on-device inference.
- Keras: It is an open-source software library that provides a Python interface for artificial neural networks.
- OpenCV: It is a library of programming functions mainly aimed at real-time computer vision.
- PyTorch: It is an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing.

Functional Specification:

- COCO Dataset API (Only Used for model Training)
- Camera
- Earphone

- Google Text-to-Speech (GTTS): Text-to-Audio conversion.

Interactions:

- Continuous Video input is taken from the camera.
- After a certain interval frame is extracted and captioned.
- The text-to-speech module gives audio as an output.

VII. CONCLUSION

This Project report presents a synopsis of enabling a real-world experience through a speech-based feedback system. The idea of a device that includes a Raspberry Pi 4 and camera module to provide a brief description of the surrounding. Similarly, objects present in front of the user are identified and communicated to the person who is using the device. In a study conducted, it was found that visually impaired people had difficulty in identifying whether there are any hindrances in front of them. Our project solves these challenges and aids the visually impaired to get their tasks done in the same manner as that of a normal person. Our project, therefore, is aiming to make the living of the visually impaired easier as well as help them get through their daily activities without meeting any dangerous obstacles and wish to incorporate several new features to the system likenavigation.

VIII. REFERENCES

[1]. Adela Puscasiu , Alexandra Fanca, Dan-IoanGota, HonoriuValean, “Automated image captioning” Department of Automation Technical University of Cluj-Napoca Cluj-Napoca, Româniadoi: 10.1109/AQTR49680.2020.9129930.

[2]. Varsha Kesavan Electronics and Telecommunications Fr. Conceicao Rodrigues Institute of Technology, Mumbai University “Deep Learning based Automatic Image Caption Generation” 2019 Global Conferencefor Advancement in Technology (GCAT) Bangalore, India. Oct 18-20, 2019, doi: 10.1109/GCAT47503.2019.8978293

[3]. Faruk Ahmed, Md Sultan Mahmud, Rakib Al-Fahad, ShahinurAlam, and Mohammed Yeasin Department of Electrical and Computer Engineering The University of Memphis, Memphis, TN 38152, USA, “Image Captioning for Ambient Awareness on a Sidewalk”, 2018 1st International Conference on Data Intelligence and Security, doi: 10.1109/ICDIS.2018.00020.

[4]. Cristian Iorga, Victor-Emil Neagoe, Department of Applied Electronics and Information Engineering “Politehnica” University of Bucharest Bucharest, Romania, “A Deep CNN Approach with Transfer Learning for Image Recognition”, ECAI 2019 - International Conference– 11th Edition Electronics, Computers and Artificial Intelligence 27 June-29 June, 2019, Pitesti, ROMÂNIA, doi: 10.1109/ECAI46879.2019.9042173.

[5]. Tsung-Yi Lin Michael Maire Serge BelongieLubomirBourdev Ross Girshick James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollar, “Microsoft COCO: Common Objects in Context” arXiv:1405.0312v3