JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Android Application for Visually Impaired Person Using Deep learning.

Image captioning with the help of Deep Learning and Neural Networks.

Shaikh Mohd Ashfaque, Hozefa Abbas Arielwala, Mohd Eklaque Shah, Salik Zulfiqar Ahmed Ansari,
Assistant Professor of Computer Department, Student, Student, Student,
B.E Computer Engineering,
Rizvi College of Engineering, Mumbai, India.

Abstract: Presently, visually impaired individuals carry stics with them which helps them to scan their surroundings for obstacles or orientation marks and to measure the distance between them and objects in their surroundings. If the visually impaired people are provided with an audio description of their surroundings, it would have a significant effect on their lives and help them understand their surroundings better. The recent advances in Deep Learning and Computer Vision have led to excellent Image Captioning models using advanced techniques like Deep Reinforcement Learning. These captions can then be read out loud to the visually impaired so that they can have a better understanding of their surroundings. Our model uses a convolutional neural network (CNN) to extract features from an image which is then converted into a sentence, describing the image invalid English by feeding it to a recurrent neural network or a Long Short-Term Memory network. We believe that our model with greatly improve the life of visually impaired people by allowing them to understand their surroundings.

Index Terms **- Component, formatting, style, styling, insert.**

1. PROBLEM STATEMENT:

The burden of visual impairment in India is estimated at **62 million**; of these, 54 million persons have low vision, and 8 million are blind. People who are visually impaired miss out on so many opportunities to experience the things going on in the world. Their life becomes easier if they have a person to support them and describe things to them and help them. Visually impaired people also have problems dealing with physical currencies, if they encounter dishonest people in the world, they get scammed by exchanging wrong amounts of money.

An assistant cannot be present at all times to help them and not everyone is able to afford an assistant or they might not have any support from family. But guess what thing is present with a person at all times? Yes, it's a smartphone. With help of modern-day technologies like Cloud Computing and Deep Learning, this project has devised a way for helping visually impaired people from describing things to them to detecting currencies and their denominations.

By using the application in this project, a visually impaired person can capture an image through the back camera of their smartphone and have the scenery described to them with the help of audio feedback, so they don't miss out on what's going on in the outside world. Similarly, this application would help them in describing currencies to them. The ultimate goal of this project is to help the visually impaired community and make them feel one with this world.

2. Literature Review:

1. Object Recognition in a Mobile Phone Application for Visually Impaired User (Refer Reference [1])

In this paper, the main problem in a practical use of the color detection module was caused by varying lighting conditions, i.e., when the camera must be used during photo capture. In good lighting conditions color detection performance was voted adequate by blind users of the application. Also, the usability of the light direction detector has been positively apprised by visually impaired users. The application allows to locate the light source after a short time delay needed by the built-in automatic procedure of image capture settings of the camera. Three image processing and recognition algorithms dedicated for blind users are proposed.

Namely the color detector, the light direction detector and the object recognition algorithm. The developed software tools were implemented and tested on the smart phones equipped with a digital camera: HTC Desire HD, HTC Explorer and Sony Xperia S. We noted that performance of these algorithms can depend on the quality of the built-in camera and image acquisition lighting conditions. The application is currently under tests among a number of blind users.

2. A Smartphone Based Obstacle Detection and Classification System for Assisting Visually Impaired People (Refer Reference [2])

This proposed obstacle detection module can function as an independent application, since it can detect dangerous moving objects very fast in complex environments without any prior information about the size, shape or position of the obstacles to be detected. We start by selecting a set of interest points extracted from an image grid and tracked using the multiscale Lucas -Kanade algorithm. Then, we estimate the camera and background motion through a set of homographic transforms. Other types of movements are identified using an agglomerative clustering technique. Obstacles are marked as urgent or normal based on their distance to the subject and the associated motion vector orientation. Following, the detected obstacles are fed/sent to an object classifier. We incorporate the HOG descriptor into the Bag of Visual Words (BOVW) retrieval framework and demonstrate how this combination may be used for obstacle classification in video streams. The method works in real-time and is implemented on a regular smart phone as a mobility tool attached to the user with the help of a chest mounted harness. This technique is able to detect both static/dynamic obstacles and to classify them based on the relevance and degree of danger to a blind person. We tested our framework in different environments and proved its consistency and robustness.

3. Android Application for Visually Impaired Users (Refer Reference [3])

We have delivered an Android based smart phone as a system for image processing and object recognition modules which work on images captured by a visually impaired user using a built- in camera. The goal was to design an application which would recognize objects from images recorded by the camera of a mobile device. The object gets detected by applying an Artificial neural network algorithm An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

2.1 Comparison Table:

The above-mentioned algorithms use mere object detection and not provide detailed description of image	Our architecture (CNN + LSTM) provides detailed description of images Hence giving more clarity.
ANN cannot be used extensively in Natural Language Processing as the above algorithms merely use a classification problem which only provides one word to describe the image	Our architecture makes use of the LSTM (RNN) which is extensively used in Natural Language Processing which has an amazing future scope.
Thus, HOG features are low-level features which don't make use of hierarchical layer-wise representation learning hence not used in deep learning.	CNN is a hierarchical deep learning model which is able to model data at more and more abstract representations hence extensively used in deep learning.
The above-mentioned algorithms are relatively quick and hence can be used in real-time video detection	Our architecture can only be used in images but provides detailed descriptions. With fast processing and evolution of hardware. Our methodology in future would be suitable for videos also

3. Methodology:

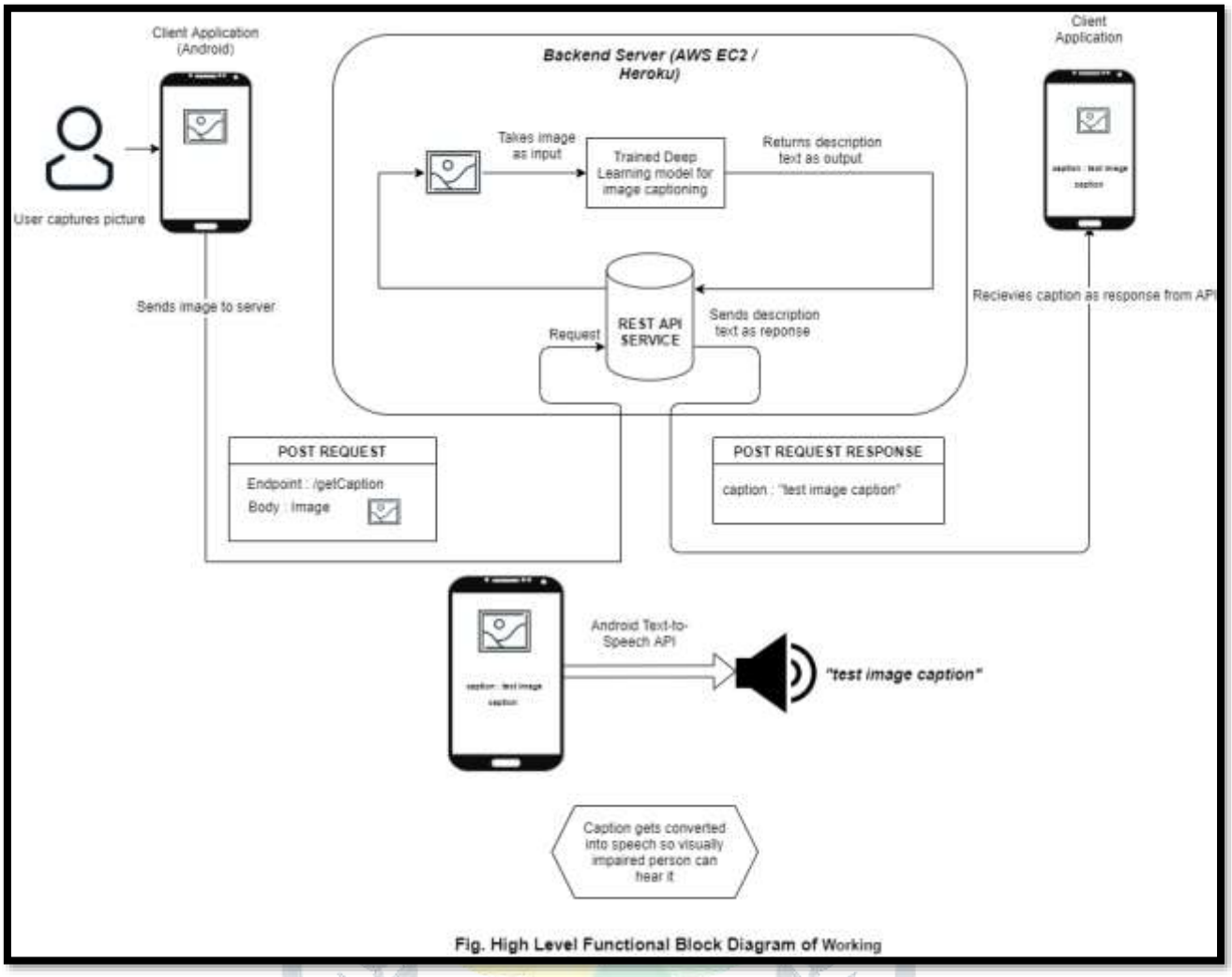


Fig. High Level Functional Block Diagram of Working

This application follows a three-tier client - server architecture which is divided into the front-end and backend.

Front End Technologies: -

1) Android
2) Retrofit REST Client API

Back End Technologies: -

1) AWS server/ Heroku Server
2) Python / Flask API
3) TensorFlow DL model

Backend: -

We will be using the AWS EC2 service for hosting our REST API service and our deep learning model as it is the most popular choice of a VPS (Virtual Private Server). A single EC2 server is known as an EC2 instance. A user having access to a terminal with SSH can easily control remote servers such as EC2. They just need to generate an SSH key-pair for securely logging in to the EC2 instance.

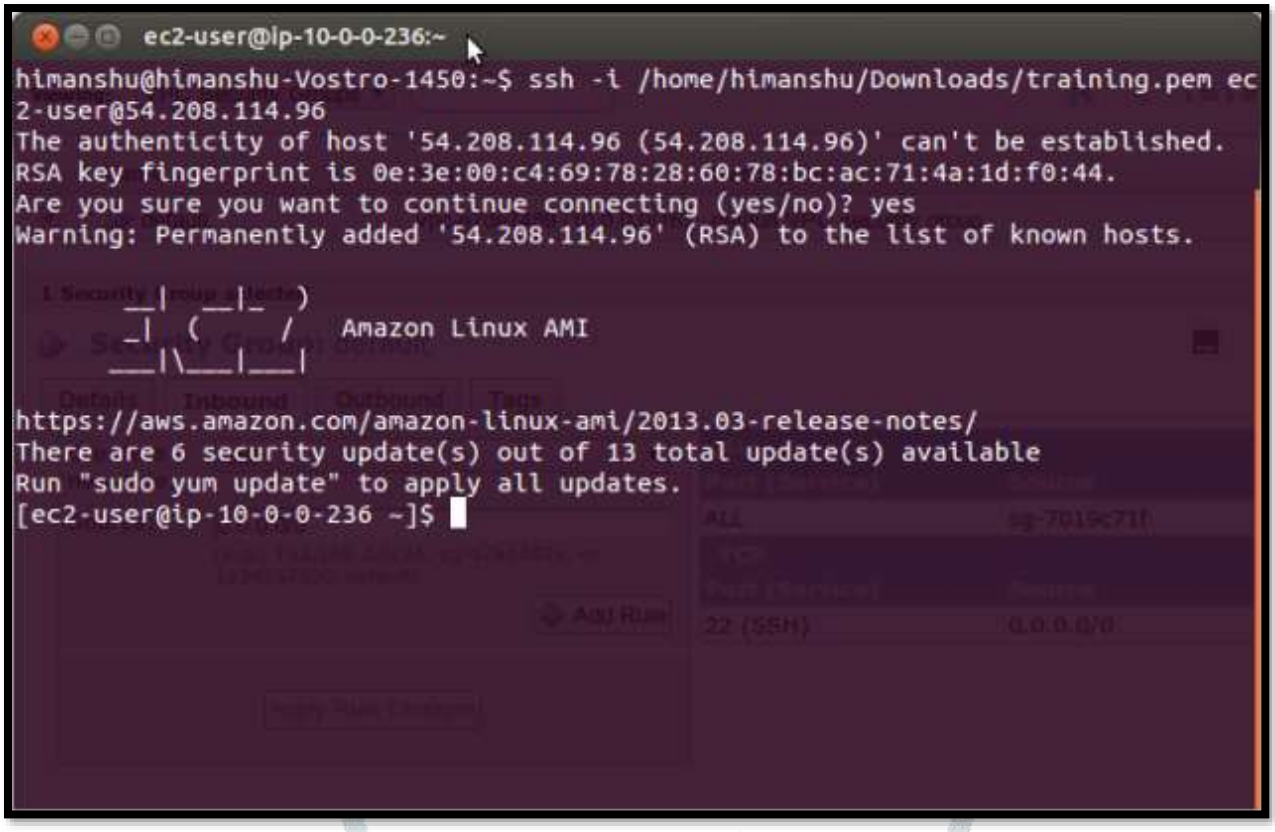


Fig. Remotely logging in to the AWS EC2 instance with help of SSH

Following this, a personal computer can anytime login into the EC2 server and control it from their own computer and manage or update the files. If there needs to be an update in the deep learning model, we would use SSH to login to EC2 instance and make changes. For managing the files and performing operations in the EC2 instance, one must be familiar with the Unix terminal as it does not provide a GUI unlike conventional computers.

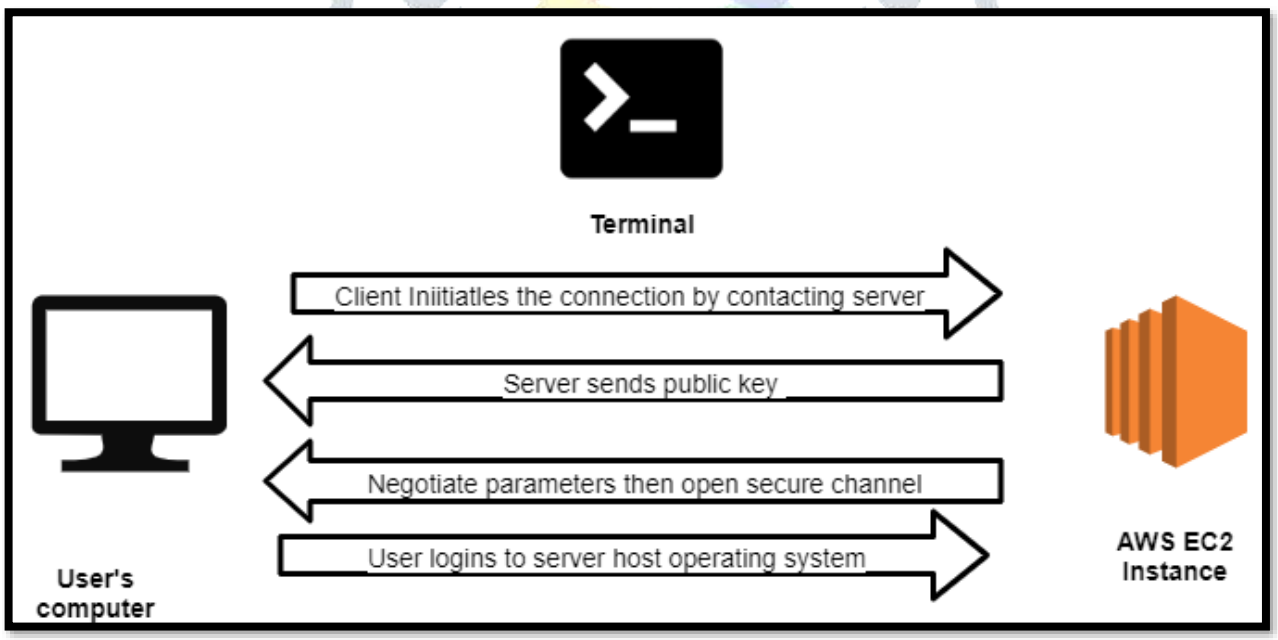


Fig. The process of Secure Shell (SSH) login to EC2 instance

After hosting Deep Learning Model along with REST API service, the routes are going to be tested with the help of Postman which is an API testing application, for making sure that the API works perfectly before building the client application.

The working starts with the client (Android application). The client application will make use of the camera to capture an image, which would be sent to the server

hosted on AWS / Heroku by making a POST request to the server with the help or Retrofit Client API in Android. The image would be sent in the body of the POST request.

The server would receive the image sent by the client on the specified endpoint as request body through the POST request made by the client (Android). When the image is received on the server, it is passed on the Trained Deep Learning Model whose architecture is shown in detail in the Algorithm part. The Deep Learning model then takes the image as input and predicts the image caption.

After predicting the image caption, the caption is passed on to the REST API service where it is passed as the response body to the POST request previously made by the client.

Front-End: -

The Front-End of this application is made with the help of Android Studio as it is the most popular option for building native applications in Android. The Android API provides numerous tools for building an application. The question arises how would a visually impaired person use a smartphone. Thankfully that would not be a problem because most modern smartphones come with the option for Screen Readers also known as Talkback which assists the visually impaired in using smartphones for navigation through the phone, and majority of the visually impaired use the Talkback feature to operate smartphones.

The home screen isn't made complex or very design oriented. It would consist of a simple button that would take the user forward to capturing a picture. For the ease of the visually impaired person, there isn't a button placed for capturing a picture. They can simply tap on the screen and capture the picture. After capturing the picture, the image would be sent to the server by making a POST request with help of the Retrofit REST Client for Android which would then return the caption and convert the caption from text to speech so the user can hear the description.

4. Algorithm:

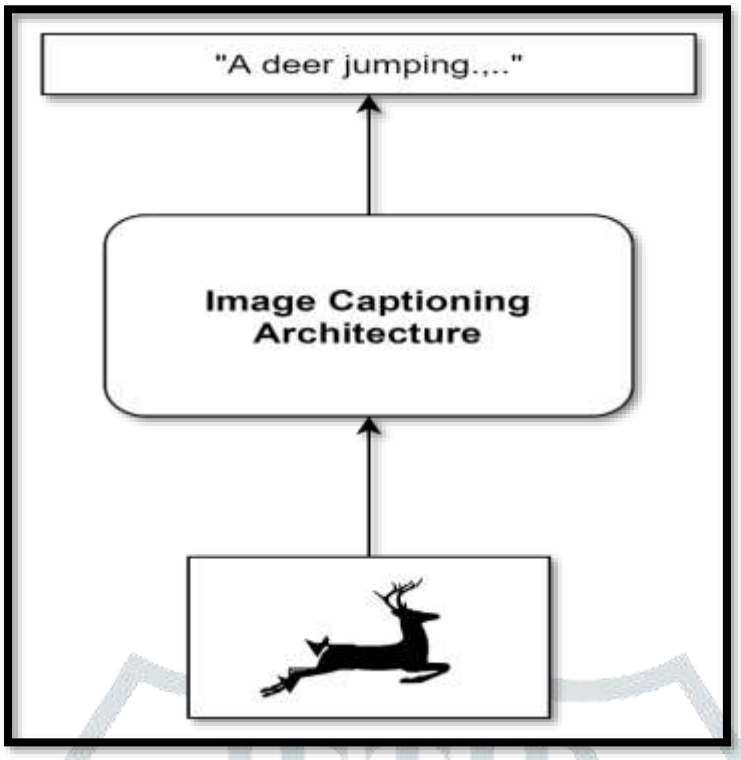
Major Deep Learning Concepts going to be used: -

1. Recurrent Neural Networks (CNN)
2. Convolutional Neural Networks (RNN)
3. Long Short-Term Memory Networks (LSTM)
4. Natural Language Processing (NLP)

Tools required: -

1. Python
2. Google Collab
3. TensorFlow Library for training Neural Networks.
4. NumPy

An Image Captioning application takes a picture as input and produces a brief matter outline describing the content of the picture.



For our application, we tend to begin with image files as input and extract their essential features during a compact encoded illustration. We are going to input these to a Sequence Decoder, consisting of many LSTM layers, which is able to decrypt the encoded image and predict a sequence of words that describes the picture.

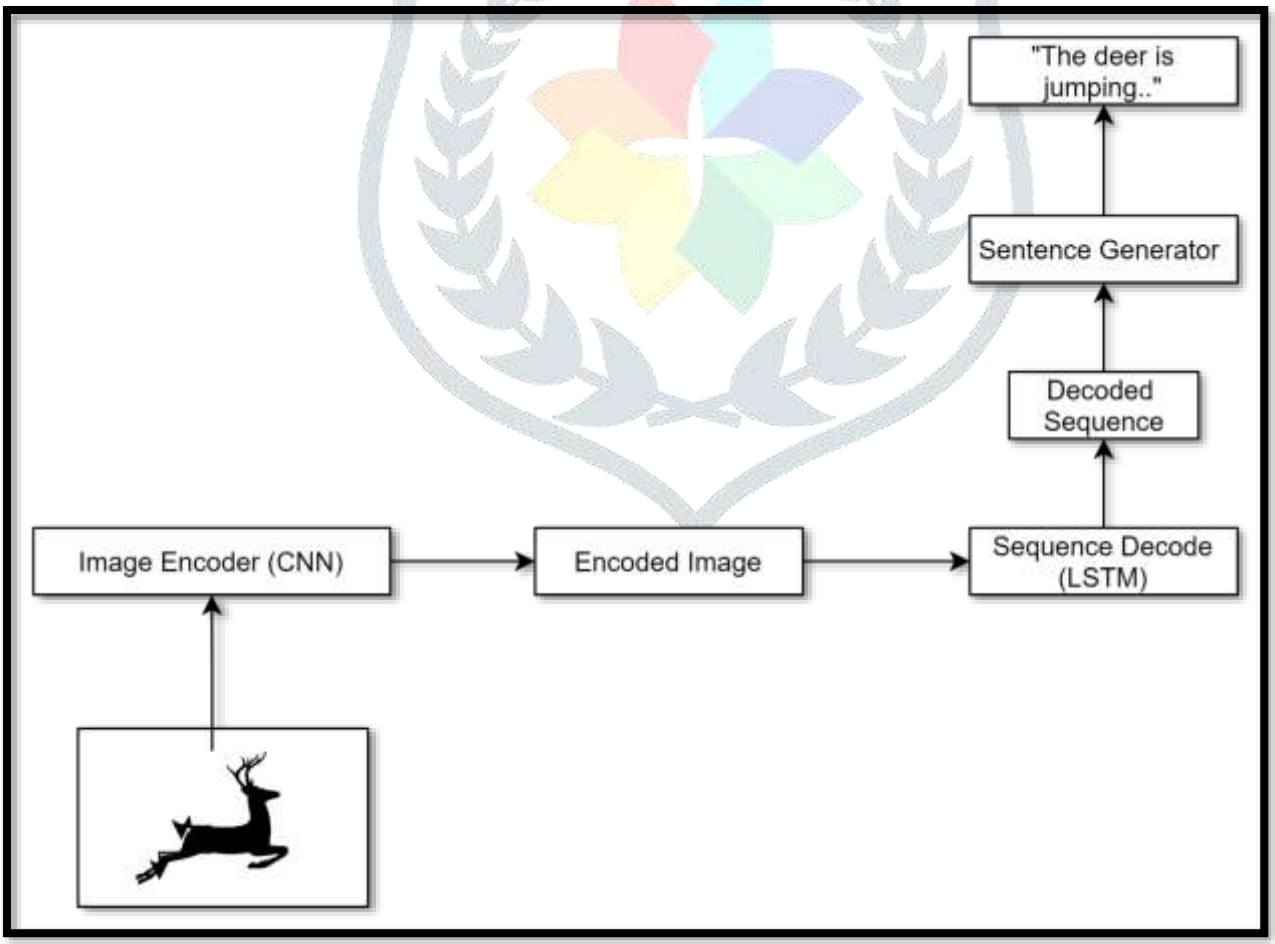


Fig. Overview of model Architecture

As for most of the Deep learning problems, we will follow the following steps**: -**

1. Download Dataset
2. Pre-Process Images and Captions
3. Prepare Training Data
4. Build Model
5. Train Model
6. Test and evaluate Model

Training data-set: -
There are many datasets available for images like the MS-COCO dataset which consists of over 30,000 images along with captions and takes over 13GB of your memory size and the training also takes a huge amount of time.

Due to our hardware constraints, we are going to use Flickr8K dataset. This is a suitable sized dataset for our problem which consists of 8,000 images, which is suitable to train our model without requiring a huge amount of RAM and disk space.

After downloading this to a '*dataset*' folder, we see that it consists of three parts:

- **Image files** in the '*Flicker8k_Dataset*' folder: This folder contains roughly 8000 .jpg files e.g., '*1000268201_693b08cb0e.jpg*'
- **Captions** in the '*Flickr8k.token.txt*' file in the main folder: It contains captions for all the images. Because the same image can be described in many different ways, there are 5 captions per image.
- **List of Training, Validation, and Test Images** in a set of .txt files in the main folder: '*Flickr_8k.trainImages.txt*' contains the list of image file names to be used for training. Similarly, there are files for validation and tests.

Training procedure: -

- For the first phase, we use transfer learning to pre-process the raw images with a pre-trained CNN-based network. This takes the images as input and produces the encoded image vectors that capture the essential features of the image. We do not need to train this network further.
- We then input these encoded image features, rather than the raw images themselves, to our Image Caption model. We also pass in the target captions corresponding to each encoded image. The model decodes the image features and learns to predict captions that match the target captions.

For the Image Caption model, the training data consists of:

- The **features (X)** are the encoded feature vectors
- The **target labels (y)** are the captions

To prepare the training data in this format, we will use the following steps:

- Load the Image and Caption data
- Pre-process Images
- Pre-process Captions
- Prepare the Training Data using the Pre-processed Images and Captions

5. Conclusion:

The purpose of this research was to identify effective strategies for helping visually impaired people with readily available tools and technologies. We have proposed the use of technologies like Mobile App Development, Deep Learning and Cloud Computing.

Thus, we have demonstrated and researched the use of different technologies and tools to achieve the solution for the given problem statement. This can be considered as a proposed model of use of mobile applications and Deep Learning to aid visually impaired people. With the ever-increasing hardware capabilities of smartphones and in other domains of technology, in the very near future, this proposed model can even be upgraded to a real-time video captioning project which would be a massive breakthrough in the domain of Computer Vision and Deep Learning.

6. Reference:

[1] K. Matusiak, P.Skulimowski and P. Strumio." Object Recognition in a Mobile Phone Application for Visually Impaired Users" 2013.

[2] Ruxandra Tapu, Bogdan Mocanu, Andrei Bursuc, Titus Zaharia." A Smartphone Based Obstacle Detection and Classification System for Assisting Visually Impaired People" 06 March 2014.

[3] A.J. Kadam1, Sharvari Awate2, Saba Desai3, Rajlakshmi Khese4 & Gauri Patange5.” Android Application for Visually Impaired Users”.

[4] Shadakshari, Dr. Shashidhara H R.” Survey on Smart Assistance for Visually Impaired Person”.

[5] Divesh Kamble, Prasanna Bakshi, Saurav Ingawale, Sagar Borhade, Anupam Choudhary.” Visual Aid Using Real-time Image Captioning” October 2018.