Bionic Eyes for Visually Impaired Using Deep Learning

Dahlia Sam[1], Jayanthi K[2], I. Jeena Jacob[3], N. Kanya[4], Shekaina Justin[5]

{dahliasam@drmgrdu.ac.in[1], jayanthi.cse@drmgrdu.ac.in[2], jeni.neha@gmail.com[3]}

Dr. MGR Educational and Research Institute, Maduravoyal, Chennai[1], Dr. MGR Educational and Research Institute, Maduravoyal, Chennai[2], Gitam University, Banglore[3].

Abstract: Today's world is a cluster of elegant creations made by God and men. Most of us are lucky enough to experience those wonders. It is not the same with the ones who cannot experience these mysteries, who cannot feel or make a visual of what is in front of them. It's a challenge to survive in this world without sight. The blind have to manage without even being able to make a proper shape and picture of what's in front of them. If technology can come to their aid, it would be a boon. This paper focuses on using deep learning to help the visually impaired draw an image in their minds using spectacles with a built in camera, which narrates the visuals or happenings around them through voice. This smart spectacles acts as the sixth sense for visually challenged people through which they can lead an independent life.

Keywords: Deep Learning, Visually Impaired, Voice Assistant, CNN, Bionic Eyes, Smart Spectacles, Blindness Support Services.

I. INTRODUCTION

Vision is the most important compared to all the other five senses. Not only human beings but almost all the living organisms rely on vision every day. They not just perform a task but tell us about the environment, explore and learn new things and make wonderful memories. However, not everyone are blessed with vision. There are still many people who cannot see and experience the happenings in front of them. These include people who are completely blind and the visually impaired. According to WHO [1], globally there are at least 2.2 billion people who have vision impairment or blindness which include both blindness by birth and later in their life. Although out of 2.2 billion, at least 1 billion people have a vision impairment that could have been prevented, the others still cannot possess vision. This paper covers a way to help the visually impaired imagine what's in front of them by describing the scene through an audio output device. This has been accompanied by the expansion and refinement of Convolutional Neural Networks (ConvNets or CNNs) for tasks such as face detection, recognition, emotion identification, caution board detection and classification, synchronized mapping and localization and so on.

Deep learning strategies have shown cutting edge results on various image recognition problems. What is generally great about these strategies is, an end to end model can be used to generate description, given a photograph, rather than requiring advanced information arrangement or a pipeline of explicitly planned models. Generating descriptions of an image requires computer vision to identify the objects in the image and natural language processing to put the words in the correct order. The main challenge is on where to place the camera in the person's body. The camera can capture the image and send it to the deep learning model for it to process the image and generate description using Microsoft Vision API [2] to detect handwritten text, emotions and celebrity faces. Vision processing results using deep learning are also dependent on how image resolution quality is. Accomplishing acceptable performance in object classification, for example, requires high resolution images or video – with the subsequent increase in the sum of data that needs to be processed, placed and transferred. Image firmness is especially important for applications in which it is necessary to detect and organize objects in the distance. This challenge is resolved by giving a spectacle fitted with a camera connected to Raspberry Pi which can also act as an eye. To make this more interactive, a speech assistant is given to the person which works on demand, that is, the device responds with the description in front of them only if the user asks it to describe the scene. To serve this purpose, Amazon Alexa was the best choice as it allows us to develop customized skills [3] apart from using the default skills already provided. Raspberry Pi serves as a computer to process the results, connect to the internet and store the generated description by the model in AWS DynamoDB, a database used to store data in the cloud and make it easy for Alexa to read the data from DynamoDB and recite.

The work targets to bring the real world as a narrative to the visually impaired. The scenes in front of the visually challenged are converted to narratives that depict the significant items in the scene or the whole scene. Some examples like, "A person is sitting and using a mobile phone" or "there is a beautiful landscape with flowers and trees" etc. The camera fitted spectacle processes the image caught on the camera using machine learning models and converts them into a description. Using a speech recognition device the user will be able to give commands to narrate the scenes in front of them. A bluetooth speaker or earphones is given to the person using it to hear the description.

II. RELATED WORK

ICASISET 2020, May 16-17, Chennai, India
Copyright © 2021 EAI
DOI 10.4108/eai.16-5-2020.2304024

Most of the existing systems focus mainly on navigation [5], [6], [13] that includes both indoor and outdoor with the help of sensors and many other systems focus only on individual activity such as only text detection [7], [14], giving alert buzz [14], [15], alert during disaster [8] and only scene description [9] including less portability. There are other object recognition and obstacle detection systems available for visually impaired but some fail to detect objects like stairs, holes [10] in the road etc. Even though these work in most of the cases, there are difficulties in using them like cost and not much easy to operate. Most of them don't identify signs or caution boards and above all the blind people will find it difficult to interact.

III. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN) has been used widely in computer vision community and applied to analyzing visual imagery. The important elements for the achievement of such methods is the accessibility of large amount of training data. Some popular Computer vision applications where CNNs are used are Facial recognition system, Smart cities (where digital cameras are widely used), Recommendation systems and so on. The improvements in Computer Vision with Deep Learning has been built and perfected with time, primarily over one particular algorithm called Convolutional Neural Network.

Naturally, you would observe the image and try to differentiate its features, shapes and edges from the image. Based on the information you gather; you would say that the object is a dog or van or cat so on. Let us see how this CNN works to detect object from image which is then converted as audio message since our target focuses on visually challenged persons. CNN takes image which is captured in glass fitted camera as input then it is analyzed with CNN algorithm using Convolution layer, polling and fully connected layers which is transferred it into object. That object is trained and tested with Flickr30k and object detection is made. This is illustrated in the Fig 1.

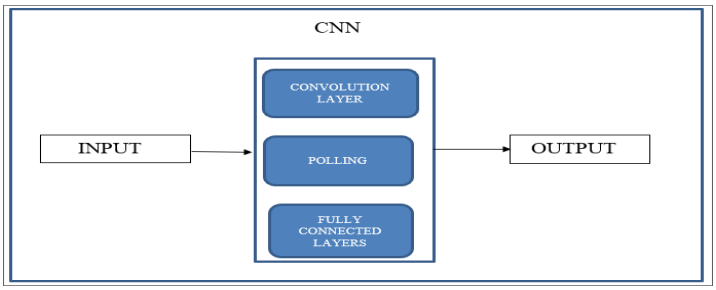


Fig 1

Convolution is the first layer to abstract features from an input image of bionic glass. Convolution conserves the association between pixels by learning image structures using small squares of input data. Below is an example to understand how image is carried out in convolution layer.

If Input image Dimensions = 3(Height) x 3(Breadth) x 1(Number of channels). The element that is involved in carrying out the convolution operation in the first part of a convolutional layer is the Kernel/Filter, K. If the selected K is a (2x2x1) matrix as represented in the equations (1), (2), (3), (4) & (5).

2	1	7		0	1		11	9
10	2	3	*	1	0	=	8	7
6	4	5						

(1)

$$(2*0 + 1*1 + 10*1 + 2*10) = 11 \quad (2)$$

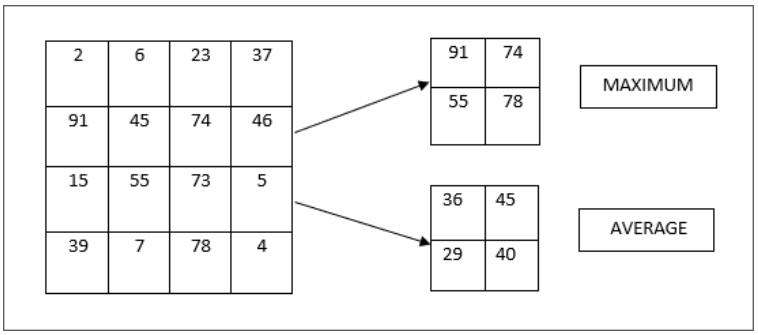
$$(1*0 + 7*1 + 2*1 + 3*0) = 9 \quad (3)$$

$$(10*0 + 2*1 + 6*1 + 4*0) = 8 \quad (4)$$

$$(2*0 + 3*1 + 4*1 + 5*0) = 7 \quad (5)$$

Similar to this convolutional layer, the pooling layer is the one responsible for reducing the spatial size of the convolved feature. This will decrease the computational power required for processing the data through dimensionality reduction. Furthermore, it is useful to extract dominant features which are rotational and positional invariant. This enables maintaining the process of training of the model effectively.

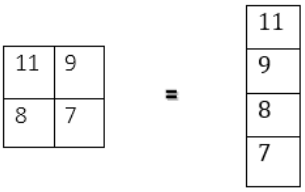
Pooling are of two types – Max Pooling and Average Pooling. Max Pooling will return the maximum value from the portion of the image covered by the kernel. On the other hand, average pooling will return the average of all the values from the portion of image covered by the kernel as shown in equation (6).



(6)

Finally a fully-connected layer is applied and it is an inexpensive way of learning non-linear combinations of the high-level structures as signified by the output of the convolutional layer. The fully connected layer is probably non-linear function.

Now the input image is converted into a suitable form for multi-level perceptron, the image can be flattened into a column vector as represented in equation (7). The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training.



(7)

IV. PROPOSED SYSTEM

The proposed system has mainly three modules to add more features to the existing system such as handwritten text, emotion detection and celebrity face recognition: The first module consists of a Raspberry Pi Camera that is connected to the Raspberry Pi. This camera helps to take pictures, record videos and also apply image effects. The camera is enabled in the Interfaces tab in Raspberry Pi configuration tool. The Python picamera library allows us to control the camera module and take pictures or record video. The camera preview only works if the monitor is connected to your Raspberry Pi. If it is accessed using remote access (such as SSH or VNC), the camera preview cannot be seen. The maximum resolution is 2592×1944 for still photos, and 1920×1080 for video recording.

In the second module, the captured image is processed to identify the objects in the picture. Deep learning algorithms such as convolutional neural networks (ConvNets or CNNs) that take an input image, process it and classify it under certain categories. Convolutional neural networks can be used for many applications such as image and video recognition, image classification, medical image study, and recommender structures to natural language processing. CNN has been used widely in computer vision community. The important element for the achievement of such methods is the accessibility of large amount of training data. Models are trained by using Flickr30k dataset that includes 30,000 images and 5 text descriptions for each image and neural network architectures that contain many layers. In deep learning, the computer model learns to perform classification tasks directly from images, text, or sound.

Models are trained using a large set of labeled data as well as neural network architectures that contain many layers. The main use of the deep learning model in this project is to take a picture as input, identify the objects in the picture and convert it into text format. All the process is done in AWS cloud and the output is stored in output.txt into

DynamoDB. Computer Vision includes a number of services that will detect and extract printed or handwritten text that appears in images. This is useful in a variety of scenarios that includes note taking, medical records, security, and banking.

Microsoft API is used to identify the texts, emotions and celebrity face if any, in the image. It is then classified into an emotion state from the following set (happy, sad, fear, anger, excitement, disgust). After the image is processed, the output string is stored in AWS DynamoDB for Alexa to read. This process is carried out for every 2 seconds and DynamoDB keeps on updating with a new record. After the output is stored in the database, in this third module the output is prompted as speech in Alexa. Alexa provides a variety of built-in capabilities, mentioned as skills. The first step in building a custom skill is to form a choice of what the custom skill will do. The functionality of the skill determines how the skill integrates with the Alexa service. A skill which can handle almost any type of request as an example: "Alexa, Start Bionic Eyes". Once the skill is started Alexa fetches the last record from the database (DynamoDB) and provides speech output. AWS Lambda is a computing service that allows run code without provisioning or managing servers. AWS Lambda executes the code only needed from a few requests per day to thousands per second. Once the output is narrated, Alexa waits for the next request and does all the process again.

V. REQUIREMENTS

a. HARDWARE REQUIREMENTS

- Raspberry Pi 3B, 1GB RAM, MicroSD card (Minimum 8GB)
- Power source (Mobile Power bank is sufficient)
- Raspberry Pi Camera or any camera that fits in the spectacles
- Amazon Alexa Preferably Echo or Echo dot for portability

b. SOFTWARE REQUIREMENTS

- Operating system: Linux / Raspbian OS
- Amazon Alexa Application for enabling custom skills
- Python Libraries: Keras, NumPy, Boto3, Azure, Pickle, msrest, awscli and all general libraries
- Dataset: Flickr30k [4]

VI. SYSTEM ARCHITECTURE & IMPLEMENTATION

The architecture of the proposed system is shown in Fig 2. It shows the set of concepts that are part of the architecture including the elements and components. It contains both physical elements such as Raspberry Pi, Alexa, camera and logical elements such as AWS Lambda function, Amazon Skill Set and ML Model.

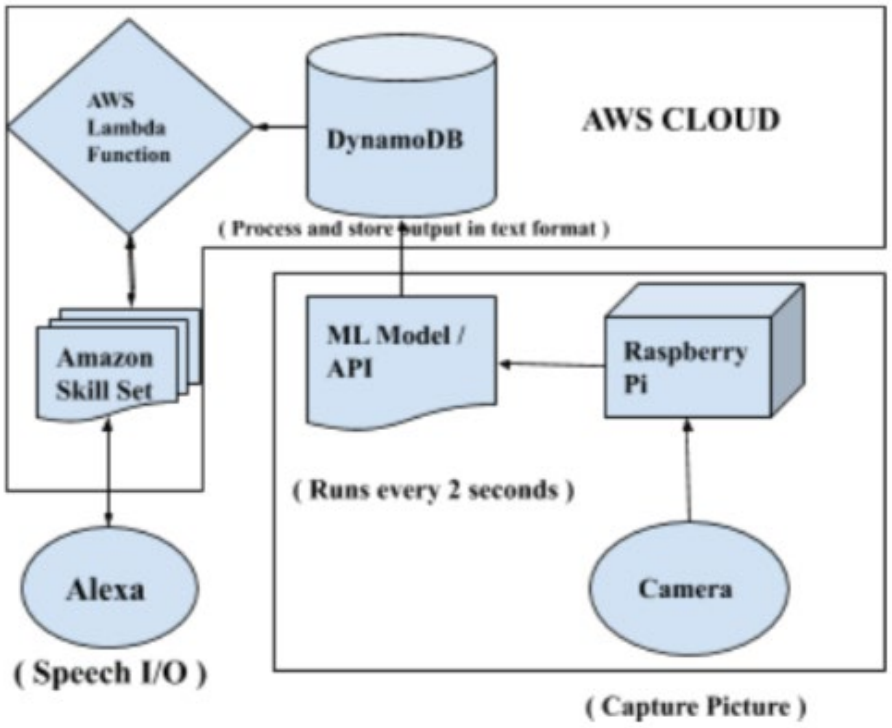


Fig.2: System Architecture

The data flow diagram shown in Fig 3 represents the flow of data from the user to Alexa. The data flows from the user's input to take a picture then the picture is converted to description using deep learning models and then it is stored in DynamoDB to recite as output.

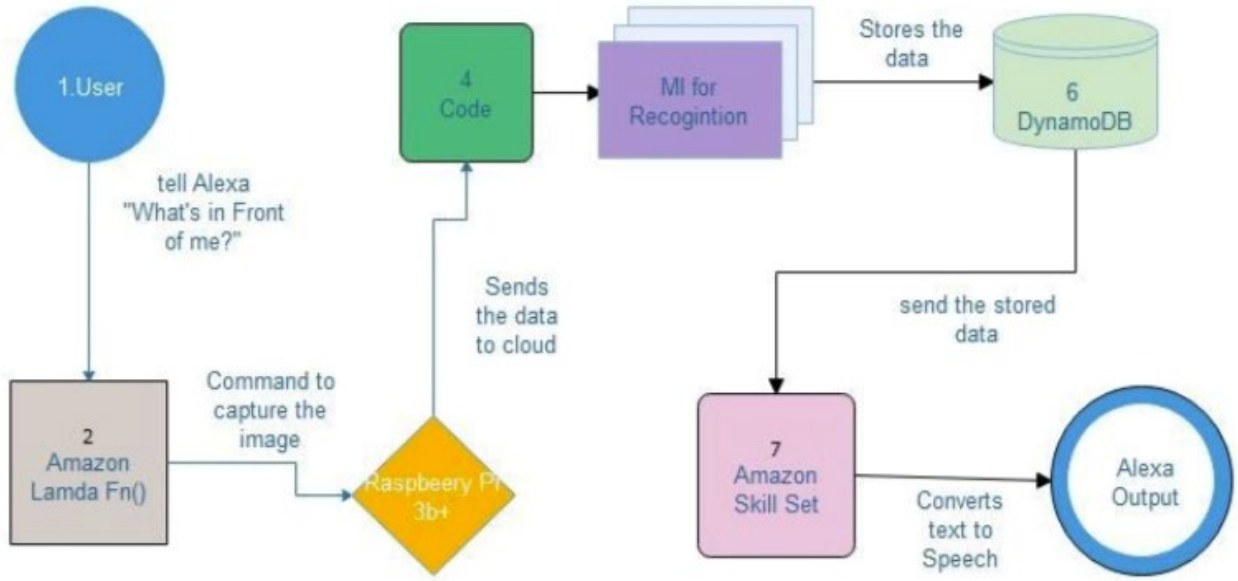


Fig. 3: Data Flow Diagram

The implementation process is described in detail below.

Phase 1: Creating custom skills using Alexa Developer Console to get user input and lambda function in AWS Console for triggering the action to fetch the last record from the database (DynamoDB).

The developer console provides various ways to create, manage and publish custom skills. The core functionality of the skill are the intents that represent operations that the user can do with the skills. There are some built-in intents already available with the skill. The intents are mapped with a set of sample utterances that represents the phrases the user can say to invoke the intents, for example "What's scene in front of me". Each skill should have an invocation name that serves as a purpose to initiate the custom skills For example "Start Bionic Eyes". Once the skills are developed the skills can be tested using the simulator in the test tab or in the Alexa device. These skills have to be enabled in the Amazon Alexa Application paired with the Alexa device in order to use.

AWS Lambda Function is written using python programming language to respond for the user with the last record from the database. Alexa Skills Kit and DynamoDB are added as a trigger to merge all of them. The function has to be explicitly called using the voice command such "Describe the scene in front of me".

Phase 2: Developing Deep Learning model

The deep learning model to generate description is the implementation of the caption generator model used in [11], [12]. The model can be described into three parts: A VGG Model with 16 Layers pre-trained using ImageNet dataset. The input features for the model is a vector of 4096 elements. The images are preprocessed with the VGG Model excluding the output layer which generates an extracted feature that is used as input. The word embedding layer is used to handle text input which is then followed by LSTM - RNN layer (Long Short-Term Memory recurrent neural network layer) with 256 memory units. The input to this layer is the predefined length of 34 words that are then fed into the embedding layer. Both the input models are regularised using 50% dropout to avoid overfitting. To make a softmax final prediction over the vocabulary (output) for the next word in the sequence, a dense layer is processed using the merged VGG and word embedding layer (Fig 4).

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 4096)	0	
embedding_1 (Embedding)	(None, 34, 512)	3880448	input_2[0][0]
dropout_1 (Dropout)	(None, 4096)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 512)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 512)	2097664	dropout_1[0][0]
lstm_1 (LSTM)	(None, 512)	2099200	dropout_2[0][0]
add_1 (Add)	(None, 512)	0	dense_1[0][0] lstm_1[0][0]
dense_2 (Dense)	(None, 512)	262656	add_1[0][0]
dense_3 (Dense)	(None, 7579)	3888027	dense_2[0][0]

Total params: 12,227,995
Trainable params: 12,227,995
Non-trainable params: 0

Fig. 4: Layers in Deep Learning model

Phase 3: Microsoft Cognitive Services

The Computer Vision Application Programming Interface (Computer Vision Documentation n.d.) and Face Application Programming Interface (Microsoft Build n.d.) provided by Microsoft is used to detect text (text in images and handwritten text), emotions with 8 emotions (anger, contempt, sadness, disgust, fear, neutral, happiness and surprise) and celebrity faces to improve the detail of the description. The azure.cognitiveservices (models, ComputerVisionClient) and msrest.authentication (models, FaceClient) libraries are used to invoke the API call.

VII. RESULTS

Different types of input were given and the description given was checked. The inputs taken were of 5 different categories – a scene, text written by print or handwriting, celebrity face, sign boards and facial emotions. The results obtained is shown in the images below (Fig 5 to Fig 14).

- SCENE DESCRIPTION

Fig. 5: Sample Input Images

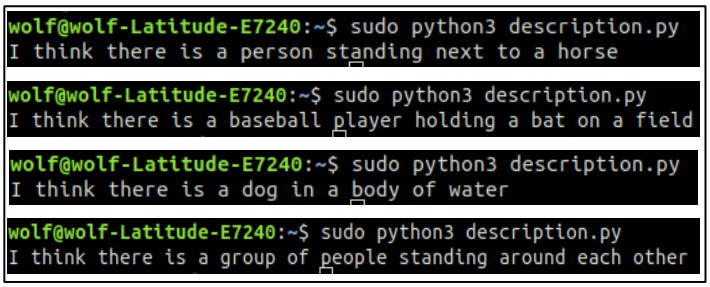


Fig. 6: Generated Description for the above images (Left to Right and Top to Bottom)

- TEXT/ HANDWRITING

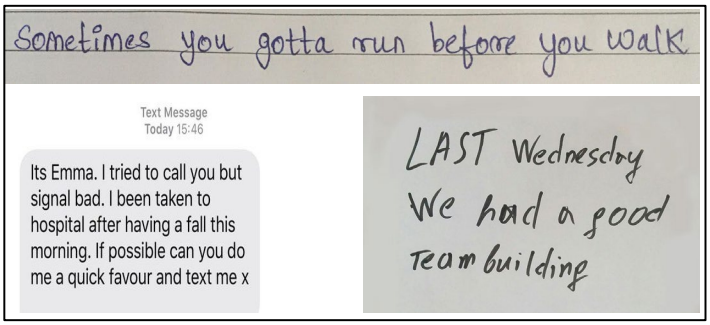


Fig. 7: Sample Input Text Images

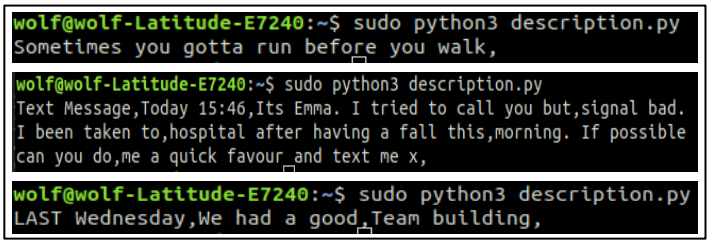


Fig. 8: Generated output for the above text images

- CELEBRITY FACE RECOGNITION

Fig. 9: Sample Input Images of Celebrities

```
wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
I think there is Emma Watson wearing a suit and tie

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
I think there is Albert Einstein standing in front of a mirror posing
for the camera

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
I think there is Steve Jobs, Bill Gates sitting in a chair

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
I think there is Satya Nadella wearing a suit and tie
```

Fig. 10: Generated Prediction of the above images

- SIGN BOARDS

Fig. 11: Sample Input Images of Sign/ Caution boards

```
wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
CAUTION,WET FLOOR,CLEANING,IN PROGRESS,

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
CAUTION,STAIRWAY,

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
CAUTION,HIGH,VOLTAGE,

wolf@wolf-Latitude-E7240:~$ sudo python3 description.py
DANGER,CONSTRUCTION,SITE,KEEP OUT,
```

Fig. 12: Prediction for the above input images

- EMOTION DETECTION

Fig.13: Sample Input Images of Different Emotions

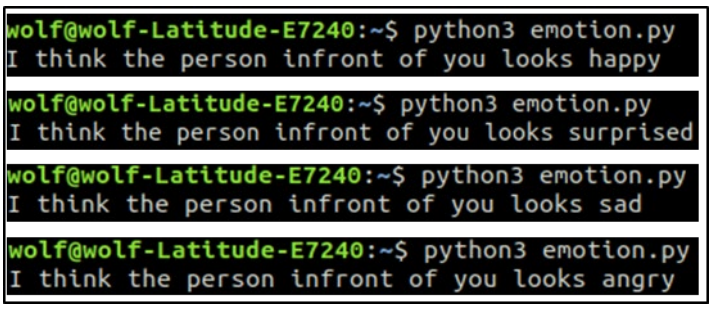


Fig.14: Predicted Emotions of the above images (Left to Right, Top to Bottom)

VIII. ANALYSIS

The accuracy of the deep learning model is 85%, it is able to predict most of the description but in some cases the model deviates from the actual description and for blurry images, cartoon images the model makes wrong predictions, Microsoft's Application Programming Interface performs very well in text/handwriting detection and celebrity face recognition. In emotion detection also it detects most of emotion but for some emotions it predicts falsely. The results are shown in the table 1 and graph in Fig 15 below.

Table 1: Accuracy Achieved

ACTION	ACCURACY (in %)
Scene Description	85
Text/Handwriting Detection (including sign boards)	93
Emotion	90
Celebrity Face	94

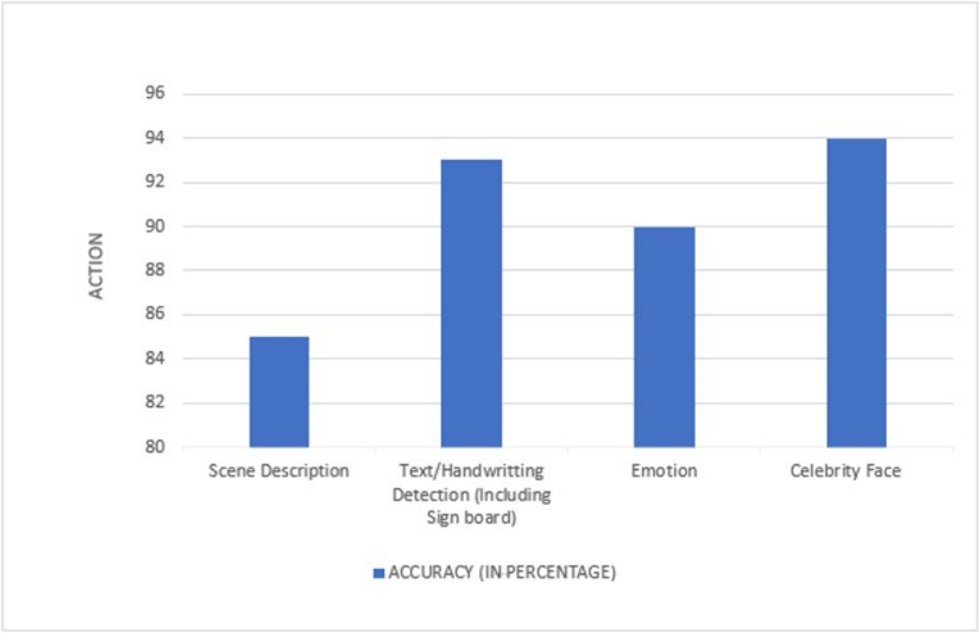


Fig 15: Accuracy Level of deep learning

IX. CONCLUSION

This paper has proposed a system to help visually impaired people to experience and draw a picture in mind of the scene in front of them using a camera and speech output device. Image is captured using a camera, converted into description, stored in a database and finally recited as an output on demand using Alexa. The user will be able to give commands to narrate the scenes in front of them. A Bluetooth speaker or earphones is given to the person using it to hear

the description. This system has extra features such as text and handwriting detection, emotion and celebrity face recognition. The results have proven to be accurate in almost all situations. There is no doubt that this system will effectively serve in describing the scenes and will be very helpful for visually impaired people in the coming days.

XI. REFERENCES

[1] *WHO, World report on vision 2019.* [Online] https://www.who.int/publications-detail/world-report-on-vision (Accessed 2 February 2020).

[2] *Computer Vision Documentation n.d.* [online] https://docs.microsoft.com/en-in/azure/cognitive-services/computer-vision/ (Accessed 4 February 2020).

[3] *Amazon Alexa n.d.* [online] https://developer.amazon.com/en-US/docs/alexa/custom-skills/understanding-custom-skills.html (Accessed 15 February 2020).

[4] *Microsoft Build n.d.* [online] https://docs.microsoft.com/en-in/azure/cognitive-services/face/ (Accessed 15 February 2020).

[5] Mary Praveena, S., Teena Sree, G, Winonah Rajendran, Pavithra, P. Li-Fi based indoor navigation system for the visually impaired people. International Journal of Engineering Applied Sciences and Technology, 2020; Vol. 4, Issue 9, pp. 372-376.

[6] Mariya, I. A., Ettiyil, A.G., George, A., Nisha, S and Joseph, I.T. Li-Fi Based Blind Indoor Navigation System. Paper Presented at the 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 15-16 March 2019. Coimbatore, India.

[7] *OrCam Read n.d.* [online] https://www.orcam.com/en/read/ (Accessed 15 February 2020).

[8] Fox, Michael., White, Glen., Rooney, Catherine., Rowland, Jennifer. Disaster Preparedness and Response for Persons With Mobility Impairments: Results From the University of Kansas Nobody Left Behind Study. Journal of Disability Policy Studies, 2007; Vol. 17. Issue 4, pp.196-205.

[9] Samaneh, Madanian., David, Airehrour., Marianne, Cherrington., Nikhilkumar, Patil. Smart Cap for Visually Impaired in Disaster Situations. 2018.

[10] Sarojini, L., Anburaj, I., Aravind, R I. Smart electronics gadget for visually impaired people. International Journal of Science and Research. 2017.

[11] Young, Peter., Lai, Alice., Hodosh, Micah., Hockenmaier, Julia. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2014; Vol. 2, pp. 67-78.

[12] Mahmudur Rahman, Md. A Cross Modal Deep Learning Based Approach for Caption Prediction and Concept Detection. CEUR Workshop Proceedings, Avignon, France, 2018; Vol. 2125.

[13] Maghfirah, Ali & Tong Boon, Tang. Smart Glasses for the Visually Impaired People. International Conference on Computers Helping People with Special Needs, Switzerland, 2016; pp. 579-582.

[14] Ali, Ramadhan. Wearable Smart System for Visually Impaired People. Sensors. 2018; pp. 843.

[15] Jayashree, S., Chetan Kumar, V. Intelligent System for Visually Impaired People Using Android. International Journal of Science and Research, 2018; Vol. 7 No. 6, pp. 308-310.

ACKNOWLEDGEMENT

We thank Mr. P. Sakthi Anand, Mr. N Rahul Raju and Mr. Rajan Aiden Richard Alexander for their technical support.