CHAPTER 1

INTRODUCTION

Nowadays, computer technologies play a major role in human life. The use of technologies to improve the quality of human life is becoming a common aspect of modern society. Recently, IoT [14] has received much attention and considered as the appealing technology that allows people and things to connect anytime and anywhere. It offers a platform for sensors and smart devices to be connected continuously within a smart environment in order to provide advanced and knowledgeable services for human beings. The IoT is a network of physical objects embedded with sensors, electronics, software and network connectivity that enables these objects to collect and exchange data. This means that a user can remotely monitor or even control the states of home appliances. In general, most smart devices are embedded with sensors to detect the environmental status, machine-operating status or to collect behaviour data and can respond to users or machines since it connects to the Internet and can be interactive with humans or machines.

AI is an emerging technology in the world, which makes all other technologies to work autonomously and helps humans to get more perfection and accuracy in daily life. Research in AI has been primarily dominated by impressive advances in Machine Learning, with a strong emphasis called Deep Learning framework. It has allowed considerable achievements such as human-level performance in a highly complex game of Alpha Go, image and speech recognition and description, and even in Video games. Nowadays, Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state of the art in object detection, visual object recognition, speech recognition etc.

A small system needs training with huge data, so Integrated IoT to collect data is used as an input to the training. AI is the best solution to manage huge data flows and storage in the IoT network. IoT nowadays becoming more and more popular with the inventions of high-speed Internet networks and many advanced sensors that can be integrated into a microcontroller. Since everything will be connected to the Internet, huge data, including sensors data, user data as well as the rich data sent from home appliances and a variety of wearable devices will be collected automatically. With the increase in the number of smart devices and more and more sensors, a learning model can be constructed by machine learning algorithms to make good use of the huge data and used for predictions such as Image Classification, Facial Recognition, Speech Recognition, and Object Detection etc. For these reasons, we are proposing a novel prototype implementation by integrating IoT and AI technologies. Traditional learning can be conducted in various ways, including textbooks, children's intelligent robot, teachers and parents, smart devices etc. All these types of traditional learning have some disadvantages, like single source knowledge, limited learning space and time, lack of interaction between traditional textbooks and persons as well as the health issue raised due to using electronic devices for a long time. Some other informal learning adopts mobile learning technique to shift the learning environment from the traditional classroom to museums or outside environment. However, the learning contents are predefined and the learning models are usually fixed. Furthermore, mobile learning is mainly designed for persons, which is not suitable for persons. This paper is motivated by all these types of traditional learning. Our designed Learning device is portable and therefore ideal to simplify the interaction with the smart environment. The main effort is to provide these persons with a Learning device, in order to explore the outside learning within the smart environment by using just their Learning device.

1.1 INTEGRATION OF IOT AND AI

Generally, IoT collects the data and interacts with people. AI helps the IoT to think very smartly and has a decision to complete the work very easily. The IoT with

AI helps the persons easily carry the Learning device and also easily identify the objects. In our proposed work, IoT devices like (Arduino, Raspberry Pi) acts as input and output devices but the decision of identifying the object is done by AI which acts as a brain or CPU. Hence in the proposed work, we integrated IoT with AI.

1.2 HIGH INTERACTIONS BETWEEN PEOPLE, LEARNING OBJECTS AND LEARNING

We aim to make every person knowing or learning about them without extra efforts. This could be discussed in several aspects. (a) Independence of learning: The persons learn to recognize an object by interacting with the Learning device, without the need of guidance from third person. (b) The mobility of learning: Our proposed Learning device belongs to portable learning technology which people can wear and move easily in the outdoor environment. Without any interference, persons can learn whatever they want since they have the knowledge database in their Learning device.

1.3 ACTIVE, MOBILE AND EXPLORING LEARNING

The persons have many disadvantages with traditional learning like single source knowledge, lack of interaction and limited space and time. Therefore, this inconvenience gives the motivation to develop a Learning device based on the IoT and AI technologies. This research is to use the Learning device as wearable learning device such that persons may actively learn about the objects in the outside environment and to arouse their interest to learn.

1.4 BEING THE BRIDGE OF AI AND IOT TO LEARNING TECHNOLOGIES

AI makes the machine learn from its experiences and manage with new data. Similarly, IoT is the combination of many devices or sensors over the Internet. The proposed system adopted IoT devices and sensors to generate a picture, apply the

Deep learning technology to recognize the picture and finally get the information from the Google Cloud Server.

1.5 PROJECT OBJECTIVES

- The designed Smart wearable device can identify objects in the outside environment and give output as an audio format.
- The designed Smart wearable device provides Voice Command Assistance
- This help visually disabled person aid them in the primary learning task of identifying objects without the supervision of the third party.

CHAPTER 2

LITERATURE REVIEW

2.1 AUDO GUIDANCE SYSTEM FOR BLIND

Audio guidance system for blind by Madhura Gharat, Rizwan Patanwala, Adithi Ganaparthi, 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA) [1] With over 280 million visually impaired people worldwide, there is a crucial need for an assistive device that allows the blind people to navigate freely. Location information for indoor environment is limited. Radio Frequency Identification (RFID) tags is an effective way of giving location information to the users. We propose a RFID based system for independent navigation in a building for blind or visually impaired people. The conversion of speech to text is carried out using speech recognition software modules. This system is initiated by providing a voice command, and specifying the destination to be reached by the blind person. This navigation system will guide the blind person along the path by providing audio navigation assistance to reach the desired destination. To avoid collision ultrasonic sensor will be interfaced with the Raspberry Pi. By implementing the above technique, blind people can navigate independently and they can acquire information about their current location within the intended building.

2.2 A SMART HOME SYSTEM USER INTERFACE

Hear to see - See to hear: A Smart Home System User Interface for visually or hearing-impaired people by L. Ciabattoni, F. Ferracuti, G. Foresi, A. Monteriù 2018 IEEE 8th International Conference on Consumer Electronics [2] .In this paper, we introduce a novel approach to design a user interface for commercial Smart Home Systems (SHS), following the needs of visually or hearing-impaired users. The interface is able to transform visual information and alarms into audio signals and vice versa by using a mobile application. The aim of the interface is to make a commercial SHS usable for visually or hearing-impaired people, while maintaining a high level of acceptability, due to the use of an inclusive device, i.e., the smartphone.

2.3 ROBOTIC ASSISTANT FOR VISUALLY IMAPIRED

Robotic Assistant for visually impaired using Sensor Fusion by Gaurao Chaudhari, Asmita Deshpande 2017 IEEE Smart World.Disabled people [12] are in a dire need of electronic assistance. There are a couple of electronic assisting devices that bring their life to an easier turn. In this paper, we describe the design and implementation of a personal assistant robot for blind people. Visually impaired people need such personal assistant devices for they provide a real-time assistance regarding any necessary problem that blind people face. Some of those main problems are navigation in the indoors, identifying objects around unless getting a physical sense of those objects and sensing the surrounding with the distance of multiple objects. Our paper discusses the various application targeting features like using the LIDAR for local mapping, using a 3D camera for understanding the depth of the surrounding so that the person understands the distance and other information of the objects around. This design has been experimentally validated and required observations are posted in this paper.

2.4 THE TALKING BRAILE KEYBORD

Voice4Blind: The Talking Braille Keyboard to Assist the Visual Impaired Users in Text Messaging by Robiah Hamzah, Mohammad Izzat, and Mohamad Fadzil, 2016 IEEE 4th international Conference [15] .Nowaday the mobile cell phone technology growing exponentially and it be a fundamaental element of our modern life style. This technology help users to explore and share their activities through various messaging systems regardless of location and time. Messaging via texting is the essence of mobile communication and connectivity. However, it become increasingly difficult for visually impaired to use Short Message Service (SMS) technology of the limited functionality such as auto checking the texting contents before sent or read the message. This project focusing on system that using an audio as medium for exchanging information between blind user and mobile cell phone devices. This application using the voice haptic architecture model. The system containing a module which adopt with the text-to-speech (TTS) technology that converts text together with speech on the SMS application for helping these visually impaired people. The aim of this research project is to study an innovative solution as an approach to aid and support human independent living (HIL) for visually impaired community. Most of the new methods featuring audio are still not integrated into commercially available products, but have matured and proven to be very useful. We expect these alternatives to have a much stronger impact on future mobile devices, in which auditory interfaces will play an important role. The cost impact on future mobile devices, in which auditory interfaces will play an important role. The cost impact for this product is very cost-effective because it is compatible with smartphone and computer available in market today.

2.5 SMART HAT: WEARABLE LEARNING DEVICE

Smart Hat: Design and Implementation of a Wearable Learning Device for Kids using AI and IoTs Techniques by Hsiung Chang, Chih-Yung Chang, Bhargavi Dande.[7] This paper aims to design and implement a Smart Hat, which is a wearable device and majorly applies the IoT and AI technologies, aiming to help a kid for exploring knowledge in a manner of easy, active, and aggressive. The designed Smart Hat can identify objects in the outside environment and give output as an audio format, which adopts the IoT and AI technologies. The learning Smart Hat intends to help kids aid them in the primary learning task of identifying objects without the supervision of the third party (parents, teachers, others etc.,) in real life. This Smart Hat device provides a sophisticated technology to kids for easy, active, and aggressive learning in daily life. Performance studies show that the obtained results are promising and very satisfactory.

2.6 SPEECH BASED TEXT CORRECTION

Speech based text correction tool for the visually impaired Md. Nafiz Hasan Khan1, Md. Amit Hasan Arovi2, Hasan Mahmud3, Md. Kamrul Hasan4, and Husne Ara Rubaiyeat5. December 2015, IEEE 18th International Conference [14] and Information Technology In this modern age of revolutionary smart devices, technology reigns supreme. Visually impaired users are a part of this modern world. They deserve to taste the beauty of this technology. However, they require assistive technology which is the concern of HCI (Human Computer Interaction). In this paper, we present a text correction tool that is entirely designed for the visually impaired. Instead of having to use traditional keyboard or mouse, they can write or edit text using speech only. The text can be read to the user using speech synthesizer. The user will be notified of different events through audio feedback and the user will be able to write text entirely using speech. Different voice commands have been designed to interact with this tool. Also, important modes can be activated using a single click on a mouse button. Since clicking either of the mouse button may not be very difficult for the visually impaired, we can use this option for flexibility. Existing speech-based tools offer fast writing method, higher accuracy, but they are not optimized for the people with low or no vision. Text editing is a basic way to communicate with the computer. Providing a clear and efficient solution regarding this matter will obviously open a new door for the disabled which is in this case visual impairment. Five participants evaluated the system and the feedback was more than satisfactory.

CHAPTER 3

SYSTEM OVERVIEW

This chapter introduces about the key components of Smart Wearable Learning Device and how these peripherals interact with each other to identify the objects in the outside environment. The proposed system consists of four parts: (I) IOT controller (ii) IOT Hardware Platform (iii) AI software Platform and (iv)Google Cloud Platform. Figure 3.1 shows the system overview of the proposed system.

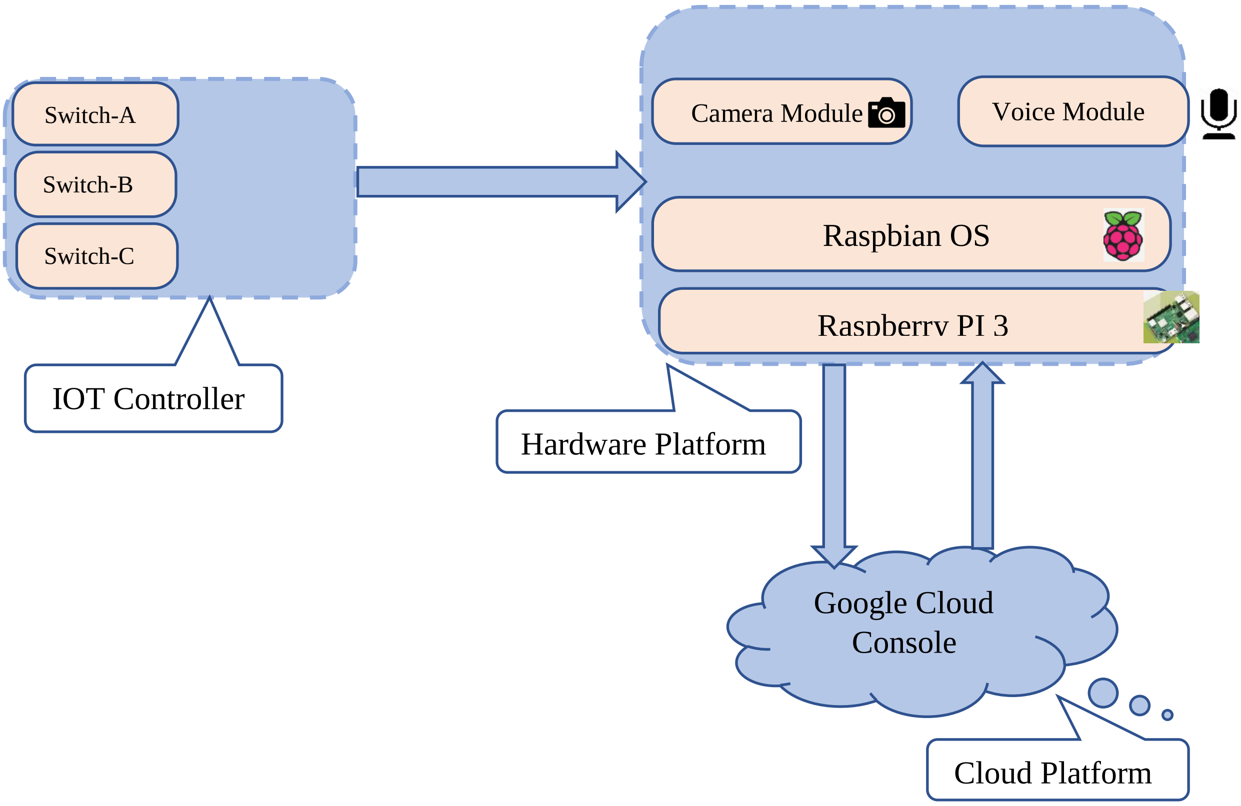

Figure 3.1 Proposed System Architecture

3.1 IOT CONTROLLER

The IOT controller platform consists of A bread board with a set of three push buttons interlined in a circuit combination with three resistors. The system allows the select different functionalities of the device. The person initiates the learning process by pressing the button in the hand, this will send a command to the raspberry pi to capture the image.

3.2 IOT HARDWARE PLATFORM

The IOT hardware platform is mainly embedded in the Smart Learning device. We used Raspberry PI 3 to implement the IOT hardware platform of this project. When the person presses the button, the command sent to the raspberry pi using wired connection. Then the camera on the raspberry pi captures the image and sends it to the image classification program then to the tensor flow. The speaker module on the raspberry pi then delivers the voice of the received content.

3.3 AI SOFTWARE PLATFORM

AI software platform is developed using Python language. The AI software platform consists of the Image Classifier, Fully Connected Network and SoftMax operation. Once the Image is captured the Image classifier program will execute and classifies the different feature of the image, then it is sent to the Fully Connected Network, Tensor Flow consists of huge set pre trained data modules, using this the image will be labelled. The TensorFlow takes the image the as input and gives out human readable string a set of five variable. One of the variables which is having highest Probability value is loaded to the Python Text to speech Converter (PyTTS) the same output is fed out through the speaker [16].

3.4 GOOGLE CLOUD CONSOLE

Google Cloud Console platform consists of Google voice assistant API (gVAPI). This voice assistant module is used to response the request, since its highest rate of performance we have chosen it. The raspberry pi will send as voice input to the gVAPI through the http request and the gVAPI will responds back with corresponding output. In this way it allows the user to resolve the related quarries.

CHAPTER 4

DESIGN AND IMPLIMENTATION

This section introduces the features of each hardware component. Then the design and implementation of the hardware and software platforms of this project.

4.1 FEATURES OF HARDWARE COMPNENTS

4.1.1 Raspberry pi 3

The Raspberry Pi is a hardware component that includes a circuit with simple I/O capabilities and Linux software. It can read a large number of switches and sensor signals and accordingly control the lights, cameras, and its various physical devices. Raspberry Pi can also be used as a computer to develop many embedded devices, perform a variety of software and applications on a 223 Linux PC and communicates with other devices in a wireless manner [11].

The various styles of the Raspberry Pi board have brought developers many advantages, whether professional or amateur. The Raspberry Pi 3 is equipped with 1GB RAM and 400MHz Video Core IV 226 GPUs. The size of the entire motherboard is the same as that of the Raspberry Pi 2. It has built-in 227 Bluetooth 4.0 and 802.11n WIFI and supports one to four USB ports. The architectural improvements and processor upgrades of Raspberry Pi 3 have increased its clock frequency by 33%. Compared to the 32-bit Raspberry Pi 2, the performance of Raspberry Pi 3 has improved by 50 to 60%. Raspberry Pi 3 has a processor speed ranging from 700 MHz to 1.4 GHz and onboard RAM memory ranging from 256 MB to 1 GB. Secure Digital (SD) cards are used to store SDHC or Micro SDHC-sized operating systems and programs. For video output, HDMI and composite video are supported with a standard

3.5 mm jack for audio output. Lower-level outputs are provided by many GPIO pins that support common protocols such as I2C.

As shown in Figure 4, the key to the success of Raspberry Pi 3 is the standard hardware interface, 236 which supports the 40-pin interface, 28 GPIO pins, I2C, SPI and UART connections. In addition to the GPIO pins, the Raspberry Pi's standard interface also provides 3.3 V, 5 V, and ground. Figure 4.1 shows the different features of the raspberry pi board.

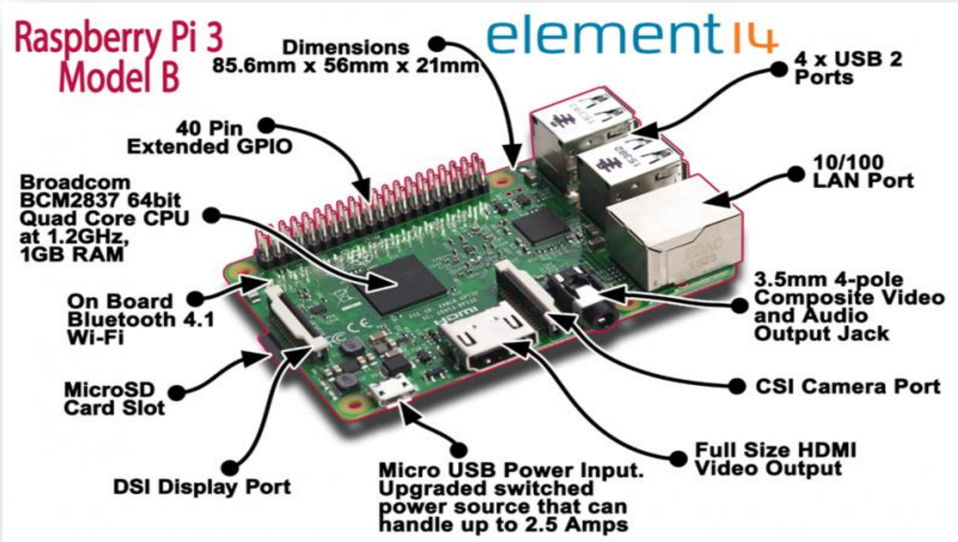

Figure 4.1 Raspberry Pi Physical Image

4.1.2 PI Camera

The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel Omni Vision OV5647 sensor of the original camera). The Camera Module can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners, but has plenty to offer advanced users if you’re looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and another video cleverness [9].

You can read all the gory details about IMX219 and the Exmor R back-illuminated sensor architecture on Sony’s website, but suffice to say this is more than just a resolution upgrade: it’s a leap forward in image quality, colour fidelity, and low-light performance. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. The camera module is very popular in-home security applications, and in wildlife camera traps**.** Figure 4.2 shows the PI Camera hardware representation; blue and white tag is used as a cable [6].

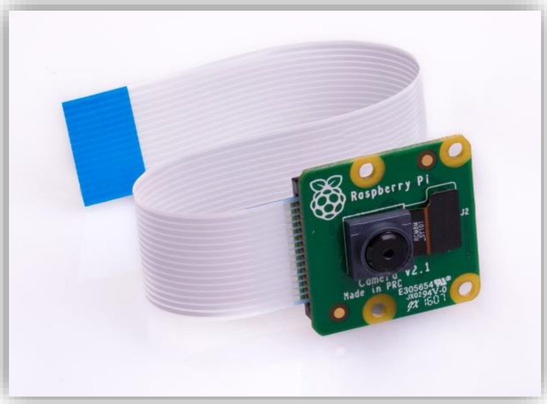

Figure 4.2 PI-Camera

4.1.3 USB Microphone

- Sensitivity: -47dB±4dB
- Frequency response: 100~16kHz

The physical outlook of microphone is shown in below figure 4.3

Figure 4.3 USB Microphone

4.1.4 Speaker

- 360degree surround sound
- FM module 280HZ-16K
- 520mAH Rechargeable Battery

The physical outlook of speaker is shown in below figure 4.4

Figure 4.4 Mini Speaker

4.2 DESIGN & IMPLEMENTATION OF HARDWARE

This subsection presents the design and implementation of the project hardware platform. We implemented the project using Raspberry Pi, which employs functions including photographing, wirelessly transmitting photos, receiving voices as well as playing learning contents. In addition, we also implemented the button controller, which can trigger the action of the device by using push buttons [15].

4.2.1 Button Controller

A small breadboard attached to the raspberry pi board consists of 3 push buttons. Each button carries a different functionality. When button A is pressed it send command to the PI board to capture a picture, when Button B is pressed Microphone starts recording, when Button C is pressed Google Assistance will be started. Figure 4.5 below shows the circuit diagram of the button controller [12].

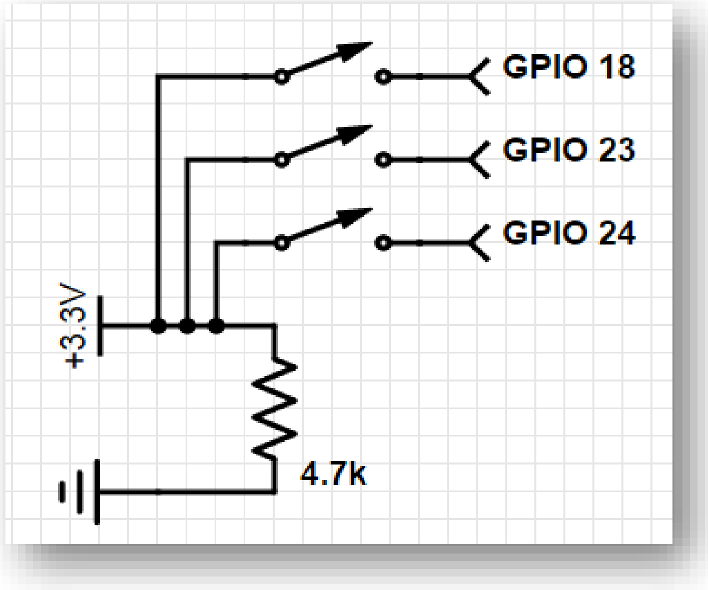

Figure 4.5 Circuit Diagram

4.2.2 Input and Output Interfaces

As an input interface three switches, camera and microphone are used, as an output interface speaker is used, connections of them is displayed as below. We programmed in such a way after capturing the picture, it sends the image classifier section to Convolutional Neural Network [3], finally the speaker of the 3.5mm jack outputs the voice of the received content. Figure 4.6 show the pictorial representation of the interfaces.

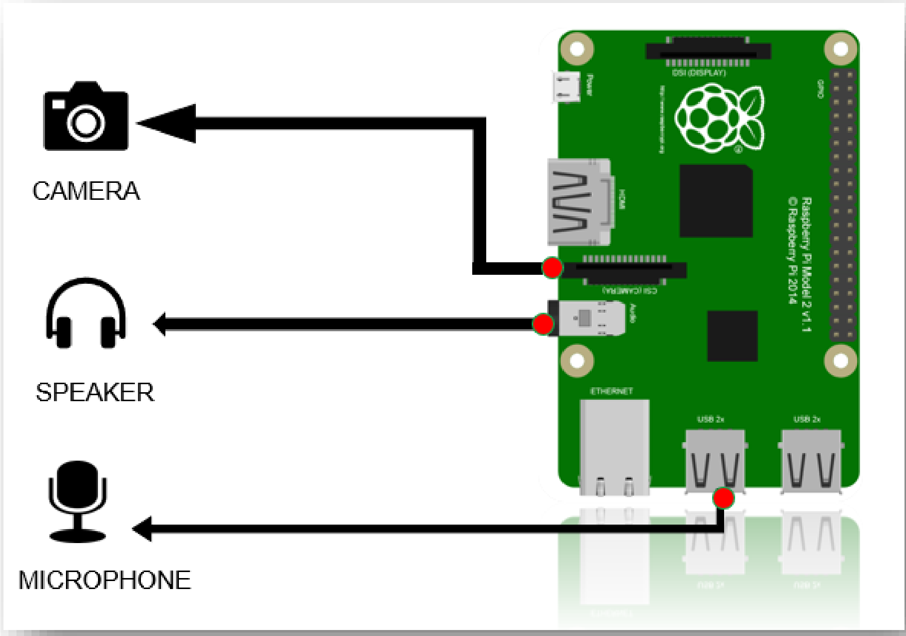

Figure 4.5 Input and output interfaces

4.3 DESIGN AND IMPLEMENTATION OF SOFTWARE PLATFORM

This section presents the design and implementation of the software platform. We used python as image processing module because of its libraries has more 2500 optimized algorithms and pre-defined functions which is helpful in Image processing. In addition, we applied deep learning framework called TensorFlow as image classification module [8].

4.3.1 TensorFlow

Google released a model called Inception-v3 with TensorFlow. The Inception-v3[15] with the TensorFlow is a pre-trained Deep convolutional neural network architecture for recognizing objects in images. The node lookup class will process the label_lookup_path and uid_lookup_path and give back a human-readable string for each classification result. The TensorFlow takes an image as an input, classifies the object in the image and then sends back a text of the object name. This object name is the major content for this device. Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning frameworks—such as Google's TensorFlow—that ease the process of acquiring data [16], training models, serving predictions, and refining future results.

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++. The single biggest benefit TensorFlow provides for machine learning development is abstraction. Instead of dealing with the nitty-gritty details of implementing algorithms, or figuring out proper ways to hitch the output of one function to the input of another, the developer can focus on the overall logic of the application. TensorFlow takes care of the details behind the scenes.

TensorFlow offers additional conveniences for developers who need to debug and gain introspection into TensorFlow apps. The eager execution mode lets you evaluate and modify each graph operation separately and transparently, instead of constructing the entire graph as a single opaque object and evaluating it all at once. The TensorBoard visualization suite lets you inspect and profile the way graphs run by way of an interactive, web-based dashboard.

4.3.2 Google Cloud Platform

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details. Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform [14]. Google Cloud Platform is a part of Google Cloud, which includes the Google Cloud Platform public cloud infrastructure, as well as G Suite, enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services

We used Google Assistant 2.6.1 in terms of implementing the device, TensorFlow successfully generates the English name of the object after identifying the object in the image. The google API used to resolve the quarries raised by the user. gVAPI will accepts the http request from the device based on the request it will gives out corresponding output to raspberry pi. The Google Assistant SDK lets you add hotword detection, voice control, natural language understanding and Google's smarts to your ideas. Your project captures an utterance (a spoken audio request, such as What's on my calendar?), sends it to the Google Assistant, and receives a spoken audio response in addition to the raw text of the utterance [4].

The Google Assistant SDK lets you add hotword detection, voice control, natural language understanding and Google's smarts to your ideas. Your project captures an utterance (a spoken audio request, such as What's on my calendar?), sends it to the Google Assistant, and receives a spoken audio response in addition to the raw text of the utterance. The SDK provides two options for integrating the Assistant into your project: the Google Assistant Library, and the Google Assistant Service. The Google Assistant Library for Python is a turnkey solution for anyone who wants to quickly integrate the Assistant into a project. The library is written in Python and is supported on hardware with linux-armv7l and linux-x86_64 architectures (like the Raspberry Pi 3 B and Ubuntu desktops) [6].

The library exposes a high level, event-based API that is easy to extend. It provides the following features out of the box:

- Hands-free activation –
- activate by saying Hey Google or Ok Google
- Audio capture and playback
- Conversation state management
- Timer and alarm management

CHAPTER 5

PROGRAMMING DESIGN FOR DEEP LEARNING SYSTEM

The programming design for the Deep learning system is divided into three modules:

i. Feature Extraction Module
ii. Image classification Module
iii. Speech Module

5.1 FEATURE EXTRACTION MODULE

In building the CNN feature extraction is the very important step of this network. It consists of three parts-

i. Convolution
ii. Pooling
iii. Flattening

5.1.1 Convolution

The primary purpose of Convolution is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. Since every image can be considered as a matrix of pixel values. Consider a 5 x 5 image (shown below table 5.1) whose pixel [13] values are only 0 and 1 (note that for a grayscale image, pixel values range from 0 to 255, the green matrix below is a special case where pixel values are only 0 and 1):

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Figure 5.1 5x5 matrix of an Image

Also, consider another 3 x 3 matrix as shown in below table 5.2: This matrix is used as convolution filter for the above given pixel matrix of image, from each set the convolution process is carried out and give out a Feature map matrix [11].

1	0	1
0	1	0
1	0	1

Figure 5.2 3x3 Convolutional matrix

Then, the Convolution of the 5 x 5 image and the 3 x 3 matrix can be computed as shown in the animation in Table 5.3 below:

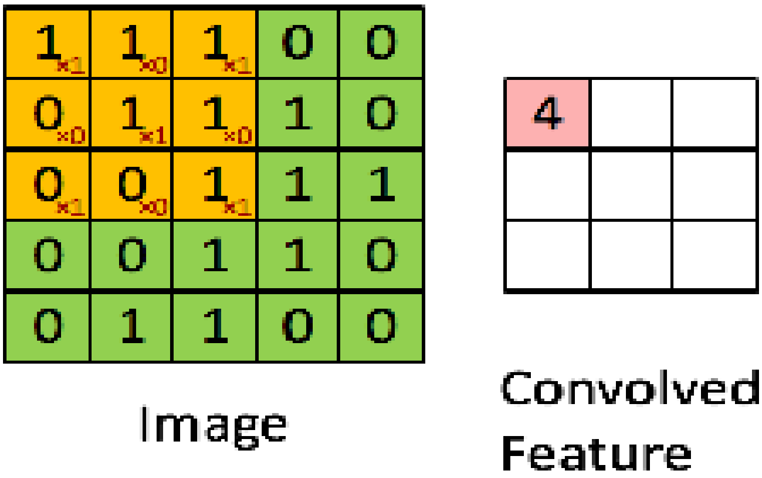

Figure 5.3 Featured Matrix

5.1.2 Pooling

Pooling (also called subsampling or down sampling) reduces the dimensionality of each feature map bur retains the most important information. In case of Max pooling, we define a spatial neighbourhood (for example, a 2x2 window) and take the largest element from the rectified feature map within that window [14]. Instead of taking the largest element we could also take the average (Average Poling) or sum of all elements in that window. In practice, Max Pooling has been shown (figure 5.1 below) to work better.

Max pooling is seemed to be give a better result when compared to the average pooling in tensor flow, so max pooling technique has been used. The below image shows the pixel matrix has been applied max pooling, and featured matrix has been obtained.

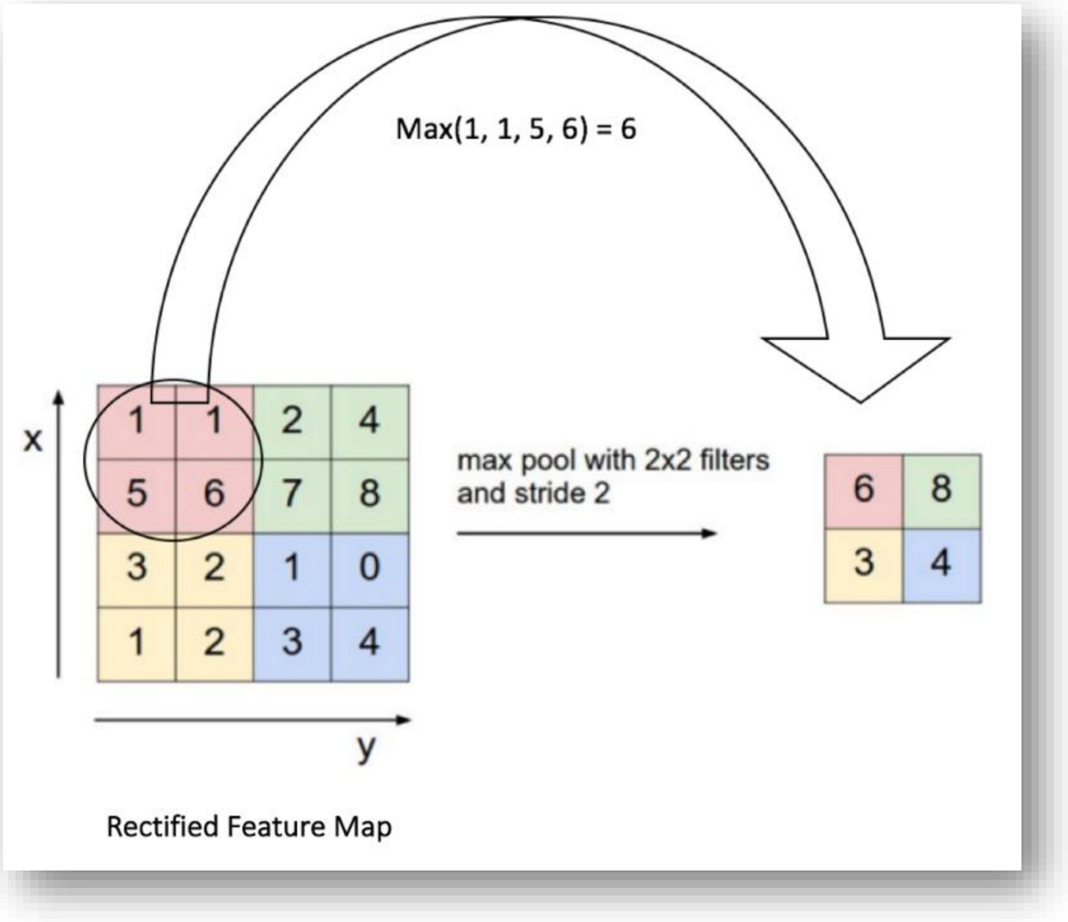

Figure 5.4 Max Pooling Operation

5.1.3 Flattening

After pooling comes flattening. Here the matrix is converted into a linear array (as shown in below figure 5.2) so that input it into the nodes of our neural network. This flattening technique will produce a weighted value to each node. These weighted values are used to calculate the extracted features. Each node consists of a weighted value. The matrix is sequentially converted into an array, starting from left corner of the matrix top row, towards the last element at right corner down side.

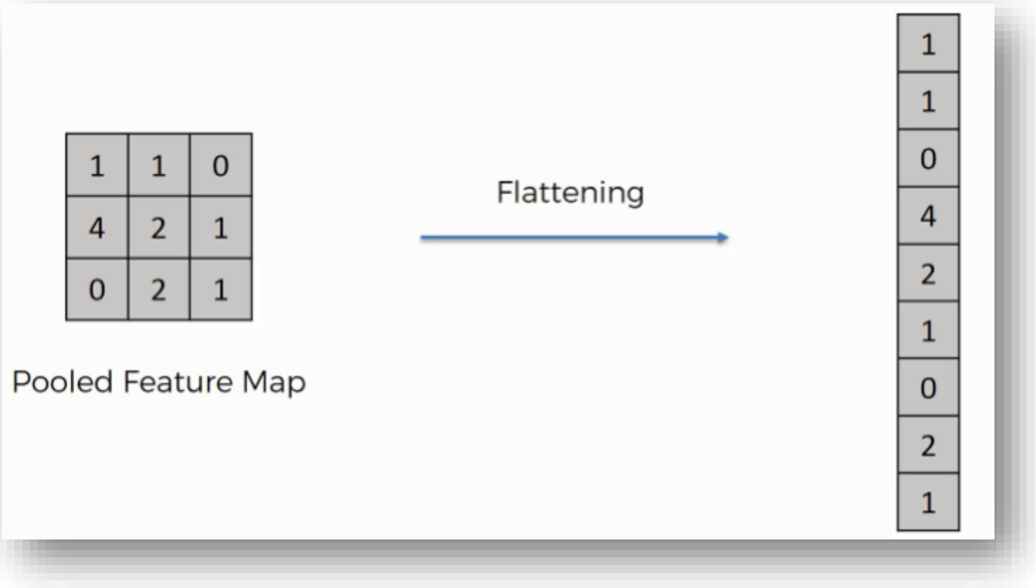

Figure 5.5 Flattening Technique

The flattening step is needed so that you can make use of fully connected layers after some convolutional layers. Fully connected layers don't have a local limitation like convolutional layers (which only observe some local part of an image by using convolutional filters). This means you can combine all the found local features of the previous convolutional layers. Each feature map channel in the output of a CNN layer is a "flattened" 2D array created by adding the results of multiple 2D kernels (one for each channel in the input layer) [15].

5.1.4 ReLU Activation Function

ReLU stands for rectified linear unit, and is a type of activation function. Mathematically, it is defined as y = max (0, x). Visually, it looks like the following (figure 5.3): The rectified linear activation function is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero.

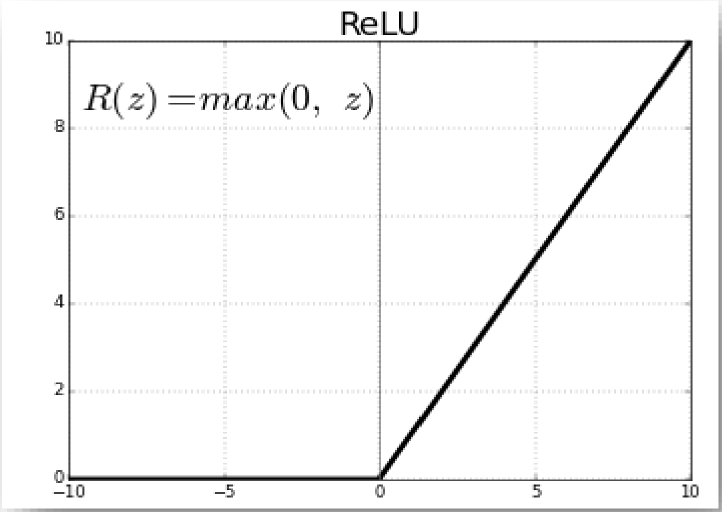

Figure 5.6 ReLU Function

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. The rectified linear activation function is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

if input > 0:

return input

else:

return 0

$$\boxed{g(z)=max\ \{0, z\}}$$

5.2 IMAGE CLASSIFICATION MODULE

Under image classification module the image will be given with a human readable string. This section contains of

i. Full connection

ii. Data Augmentation

iii. SoftMax Function

5.2.1 Full Connection

connection is connecting our convolutional network to a neural network and then compiling our network (shown in figure 5.4). Here we have made 2-layer neural network with a sigmoid function as an activation function for the last layer as we need to find the probability of the object being a cat or a dog [16].

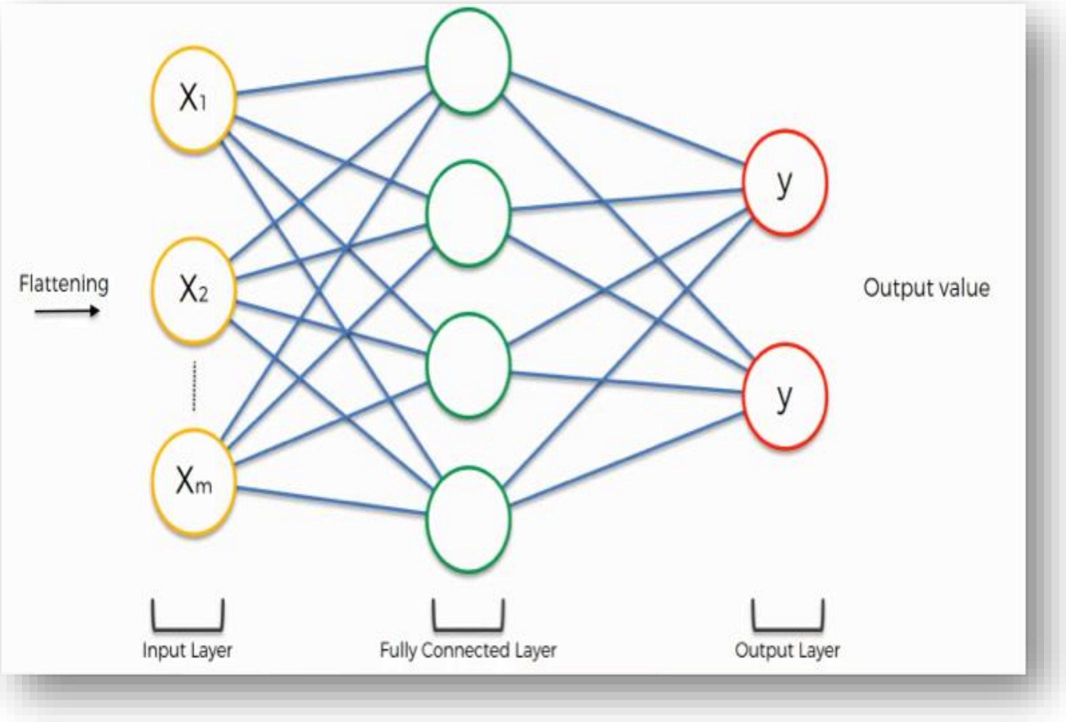

Figure 5.7 Full Connection

5.2.2 Data Augmentation

So, to get more data, we just need to make minor alterations to our existing dataset. Minor changes such as flips or translations or rotations. Our neural network would think these are distinct images anyway. Data augmentation is a way we can reduce overfitting on models, where we increase the amount of training data using information only in our training data. The field of data augmentation is not new, and in fact, various data augmentation techniques have been applied to specific problems. The Keras deep learning library provides the ability to use data augmentation automatically when training a model. This is achieved by using the ImageDataGenerator class. First, the class may be instantiated and the configuration for the types of data augmentation are specified by arguments to the class constructor [3].

A range of techniques are supported, as well as pixel scaling methods. We will focus on five main types of data augmentation techniques for image data; specifically

- Image shifts via the width_shift_range and height_shift_range arguments.
- Image flips via the horizontal_flip and vertical_flip arguments.
- Image rotations via the rotation_range argument
- Image brightness via the brightness_range argument.
- Image zoom via the zoom_range argument.

```
datagen = ImageDataGenerator()
```

Once constructed, an iterator can be created for an image dataset. The iterator will return one batch of augmented images for each iteration. An iterator can be created from an image dataset loaded in memory via the flow() function; for example: shown in figure 5.5

```
it = dataset.flow(X, y)
```

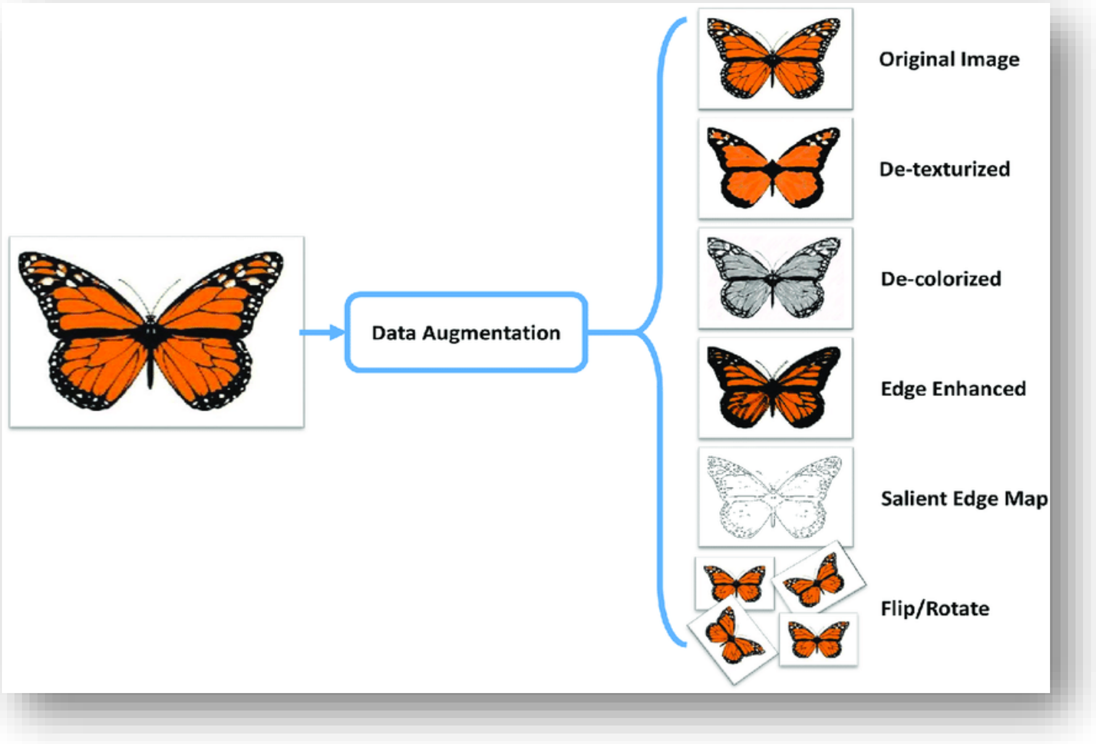

Figure 5.8 Data augmentation types

Once the iterator is created, it can be used to train a neural network model by calling the fit_generator() function. The steps_per_epoch argument must specify the number of batches of samples comprising one epoch. For example, if your original dataset has 10,000 images and your batch size is 32, then a reasonable value for steps_per_epoch when fitting a model on the augmented data might be ceil(10,000/32), or 313 batches

Model = model.fit_generator(it, steps_per_epoch=313, ...)

How to Configure Image Data Augmentation When Training Deep Learning Neural Networks by Jason Brownlee on April 12, 2019 in Deep Learning for Computer Vision Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset [3].

Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

In this tutorial, you will discover how to use image data augmentation when training deep learning neural networks. Image data augmentation is used to expand the training dataset in order to improve the performance and ability of the model to generalize. Image data augmentation is supported in the Keras deep learning library via the ImageDataGenerator class [3].

How to use shift, flip, brightness, and zoom image data augmentation.

- Image Data Augmentation
- Sample Image
- Image Augmentation with Image Data Generator
- Horizontal and Vertical Shift Augmentation
- Horizontal and Vertical Flip Augmentation
- Random Rotation Augmentation
- Random Brightness Augmentation
- Random Zoom Augmentation
- Image Data Augmentation

The performance of deep learning neural networks often improves with the amount of data available. Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples.

Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image. Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more [17].

The intent is to expand the training dataset with new, plausible examples. This means, variations of the training set images that are likely to be seen by the model. For example, a horizontal flip of a picture of a cat may make sense, because the photo could have been taken from the left or right. A vertical flip of the photo of a cat does not make sense and would probably not be appropriate given that the model is very unlikely to see a photo of an upside-down cat.

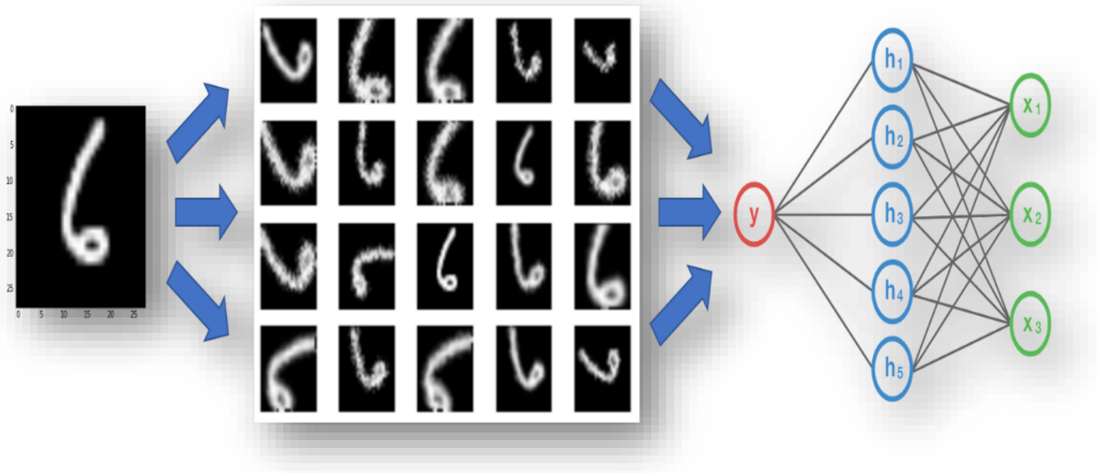

Figure 5.9 Augmented figure to full connection

As such, it is clear that the choice of the specific data augmentation techniques used for a training dataset must be chosen carefully and within the context of the training dataset and knowledge of the problem domain. In addition, it can be useful to experiment with data augmentation methods in isolation and in concert to see if they result in a measurable improvement to model performance, perhaps with a small prototype dataset, model, and training run.

Modern deep learning algorithms, such as the convolutional neural network, or CNN, can learn features that are invariant to their location in the image. Nevertheless, augmentation can further aid in this transform invariant approach to learning and can aid the model in learning features that are also invariant to transforms such as left-to-right to top-to-bottom ordering, light levels in photographs, and more [11].

Image data augmentation is typically only applied to the training dataset, and not to the validation or test dataset. This is different from data preparation such as image resizing and pixel scaling; they must be performed consistently across all datasets that interact with the model. Image Augmentation With ImageDataGenerator

The Keras deep learning library provides the ability to use data augmentation automatically when training a model. This is achieved by using the ImageDataGenerator class. First, the class may be instantiated and the configuration for the types of data augmentation are specified by arguments to the class constructor.

A range of techniques are supported, as well as pixel scaling methods. We will focus on five main types of data augmentation techniques for image data; specifically:

- *Image shifts via the width_shift_range and height_shift_range arguments.*
- *Image flips via the horizontal_flip and vertical_flip arguments.*
- *Image rotations via the rotation_range argument*
- *Image brightness via the brightness_range argument.*
- *Image zoom via the zoom_range argument.*
- *For example, an instance of the ImageDataGenerator class can be constructed.*

```
...
# create data generator
datagen = ImageDataGenerator()
# create data generator
datagen = ImageDataGenerator()
```

Once constructed, an iterator can be created for an image dataset. The iterator will return one batch of augmented images for each iteration. An iterator can be created from an image dataset loaded in memory via the flow() function; for example:

```
# load image dataset
X, y = ...
# create iterator
it = dataset.flow(X, y)
# load image dataset
X, y = ...
# create iterator
it = dataset.flow(X, y)
```

Alternately, an iterator can be created for an image dataset located on disk in a specified directory, where images in that directory are organized into subdirectories according to their class.

```
# create iterator
it = dataset.flow_from_directory(X, y, ...)
# create iterator
it = dataset.flow_from_directory(X, y, ...)
```

Once the iterator is created, it can be used to train a neural network model by calling the fit_generator() function. The steps_per_epoch argument must specify the number of batches of samples comprising one epoch. For example, if your original dataset has 10,000 images and your batch size is 32, then a reasonable value for steps_per_epoch when fitting a model on the augmented data might be ceil(10,000/32), or 313 batches.

```
model = ...
model.fit_generator(it, steps_per_epoch=313, ...)
```

The images in the dataset are not used directly. Instead, only augmented images are provided to the model. Because the augmentations are performed randomly, this allows both modified images and close facsimiles of the original images (e.g. almost no augmentation) to be generated and used during training.

A data generator can also be used to specify the validation dataset and the test dataset. Often, a separate ImageDataGenerator instance is used that may have the same pixel scaling configuration (not covered in this tutorial) as the ImageDataGenerator instance used for the training dataset, but would not use data augmentation. This is because data augmentation is only used as a technique for artificially extending the training dataset in order to improve model performance on an unaugmented dataset.

Now that we are familiar with how to use the ImageDataGenerator, let's look at some specific data augmentation techniques for image data.

We will demonstrate each technique standalone by reviewing examples of images after they have been augmented. This is a good practice and is recommended when configuring your data augmentation. It is also common to use a range of augmentation techniques at the same time when training. We have isolated the techniques to one per section for demonstration purposes only.

A shift to an image means moving all pixels of the image in one direction, such as horizontally or vertically, while keeping the image dimensions the same. This means that some of the pixels will be clipped off the image and there will be a region of the image where new pixel values will have to be specified. The width_shift_range and height_shift_range arguments to the ImageDataGenerator constructor control the amount of horizontal and vertical shift respectively. These arguments can specify a floating point value that indicates the percentage (between 0 and 1) of the width or height of the image to shift. Alternately, a number of pixels can be specified to shift the image [15].

5.2.3 SoftMax Function

Inside fully connection the node weightage will be given by SoftMax function. This will produce a probability value to each node, by calculating the input flattened array values. SoftMax function, a wonderful activation function that turns numbers aka logits into probabilities that sum to one. SoftMax function outputs a vector that represents the probability distributions of a list of potential outcomes [17].

Figure 5.10 SoftMax Input and Output System

The above image shows that SoftMax function that turns logits [2.0, 1.0, 0.1] into probabilities [0.7,0.2,0.1] and the probabilities sum to 1. In deep learning, the terms logits layer is popularly used for the last neuron layer of neural network for classification task which produces raw prediction values as real number ranging from [-∞, +∞] as shown in below equation 5.1.

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}} \quad (5.1)$$

Logits are the raw scores output by the last layer of a neural network. Before activation takes place. SoftMax is not a black box. It has two components: special number e to some power divide by a sum of some sort. y_i refers to each element in the logits vector *y*. Python and NumPy code will be used in this.

Based on the output probability values one the string which is having high probability value will be considered as the output. And this output will be given as input to the text to speech converter.

5.3 SPEECH MODULE

Main objective of this project is speech output. For speech output PyTTS is used for converting the obtained string text into speech. PyTTS is at text-to-speech conversion library in python. Unlike alternative libraries, it works offline, and is compatible with both python 2 and 3.

pyttx is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline. However, pyttsx supports only Python 2.x. Hence, we will see pyttsx3 which is modified to work on both Python 2.x and Python 3.x with the same code.

It works offline, unlike other text-to-speech libraries. Rather than saving the text as audio file, pyttsx actually speaks it there. This makes it more reliable to use for voice-based projects. The voices available will depend on what your system has installed. You can get a list of available voices on your machine by pulling the voices property from the engine. Note that the voices you have available on your computer might be different from someone else's machine. There is a default voice set so you are not required to pick a voice. This is only if you want to change it from the default.

- *import os*
- *import time*
- *def speech(var):*
- *message = var*
- *print(var)*
- *os.system('echo "' + message + '" | festival --tts')*
- *time.sleep(1)*

CHAPTER 6

RESULT AND ANALYSIS

6.1 PHYSICAL PROTO-TYPE

This section deals about the physical prototype of the model with all the peripherals connected and also it contains the demonstration of the working model.

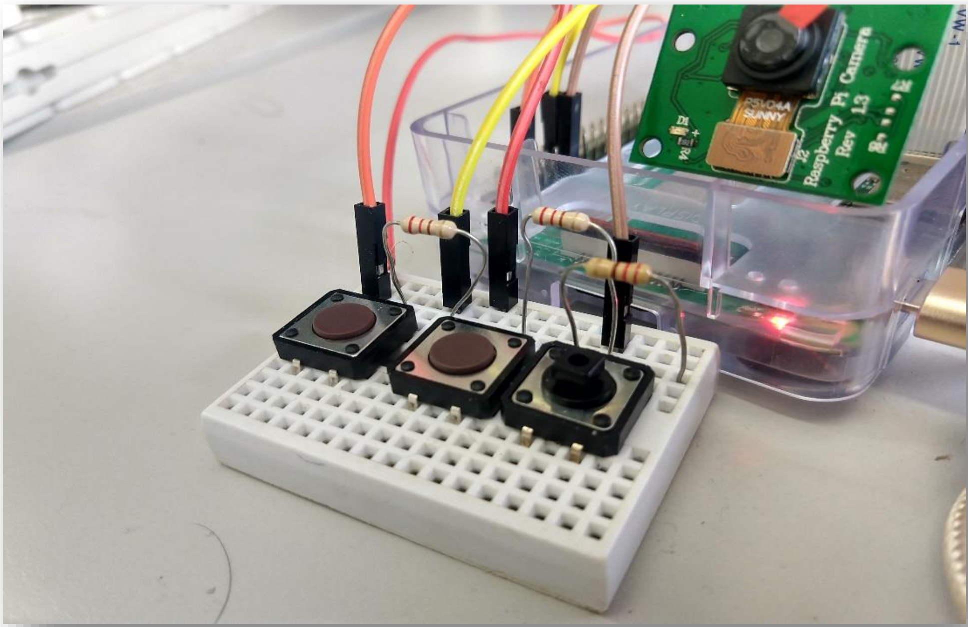

Figure 6.1 IOT Controller

The above image shows the IOT controller which contains 3 Push buttons which are used for different functionalities. Button-A is used to capture a picture, Button-B is used to capture audio, Button-C is used for voice assistance.

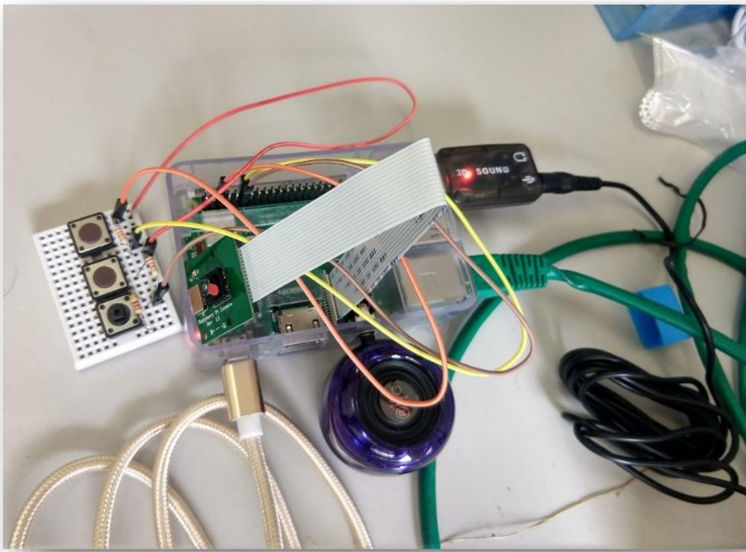

Figure 6.2 Prototype

The above image shows the complete input and output peripherals interfaced to the raspberry pi board and the working model will be shown below. Push buttons, camera and microphone were taken as input interfaces and speaker is take as an output interface.

6.2 CONTROLLER SIMULATON

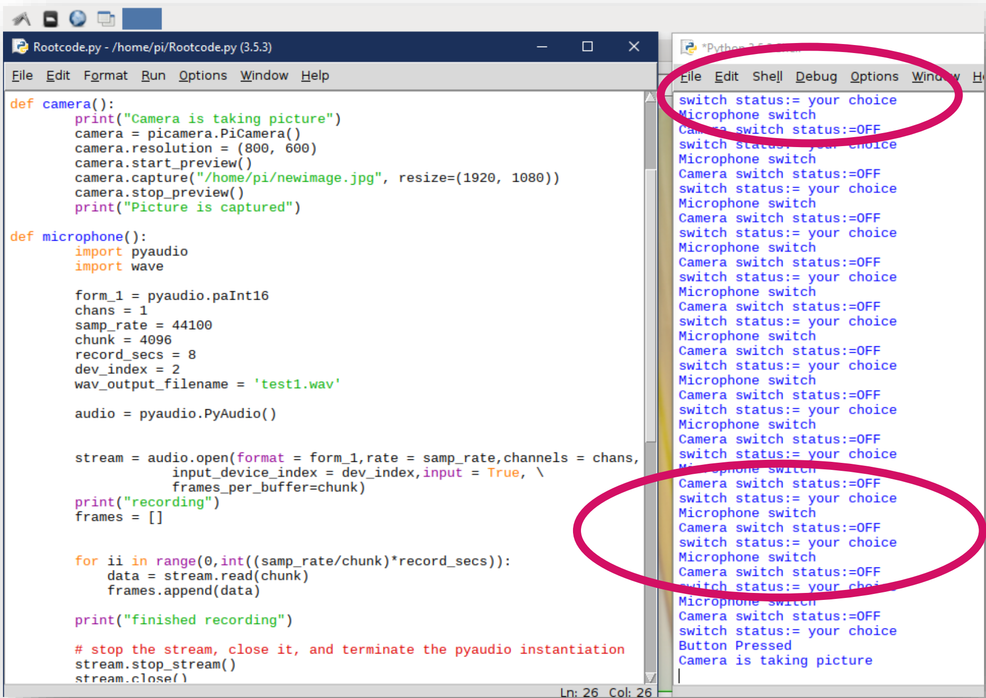

Figure 6.3 Simulation screenshot

As shown in above figure 6.3 initially camera and microphone are in off state, whenever corresponding is pressed it displays *"Camera is Capturing"* and starts the clicking image. As we can see the rounded up with pink color area state *"Button Pressed"* and *"Camera is taking Picture"*. Similarly with the microphone show in below figure 6.4.

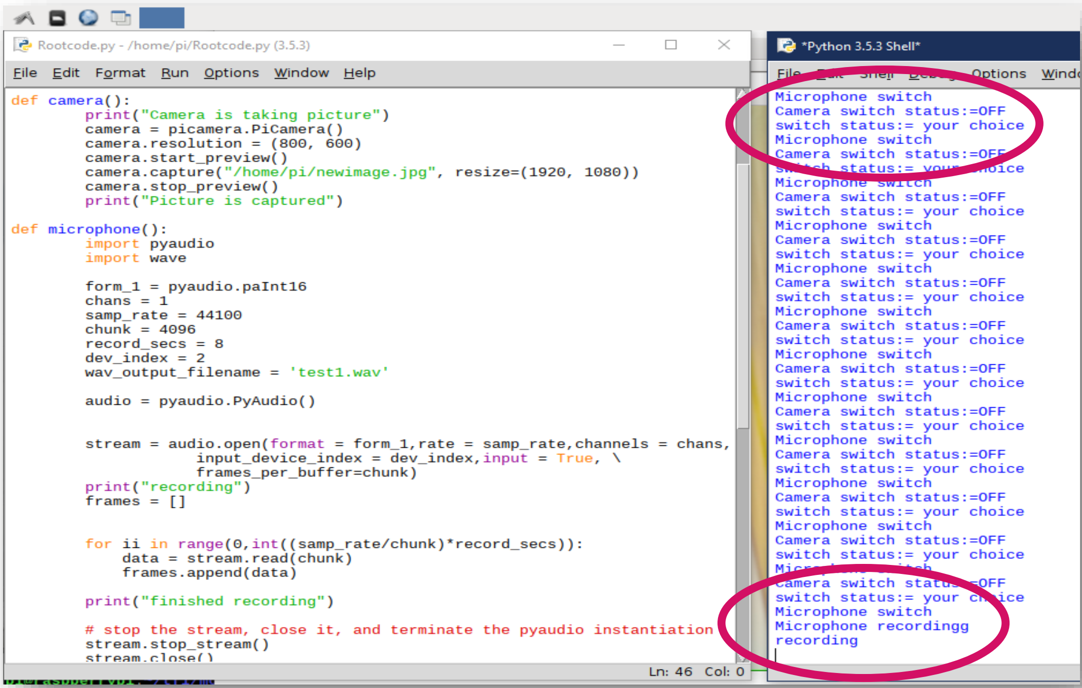

Figure 6.4 Simulation Screenshot

6.3 PREDICTION PROBABILITY

Once the image is captured it will send the image for *classify_image* section here the image will undergo the prescribed process and pop ups the probability score. As shown in below figure (6.5) a set of with 5 different strings arranged according to their probability score will be pop up. Image 6.5 shows the captured image.

Figure 6.5 Captured image

In below figure 6.6 the *node _look_up_string* will pop ups with the 5 different strings varriying probabilitie values. One of the string with high probability is *append()* to the PyTTS assuming it will be recognized object. As we can see in above figure 6.5 the captured image is a water bottle, the device comes with a conclusion of water jug with 47% scoring. In this way it can be helpful to the blind to know what is in their surroundings.

```
Button Pressed
Camera is taking picture
Picture is captured
WARNING:tensorflow:From /home/pi/Rootcode.py:19
rflow.python.platform.gfile) is deprecated and
on.
Instructions for updating:
Use tf.gfile.GFile.
water jug
water jug (score = 0.47978)
water bottle (score = 0.26398)
pill bottle (score = 0.04873)
soap dispenser (score = 0.00872)
nipple (score = 0.00871)
>>>
```

Figure 6.6 Node_Look_up_string with probability

6.3 gVAPI

The working of the google voice assistant api is displayed below. Raspberry pi board captures the user input, and triggers the command to the gVAPI. The same thing is synced with users Gmail, every notification will be transferred to particular users' android device.

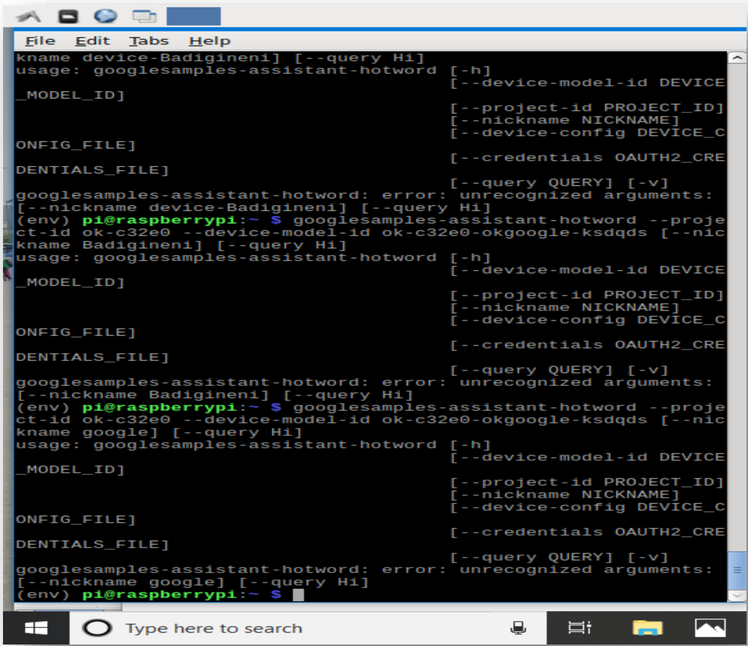

Figure 6.7 Response of gVAPI

Figure 6.7 above shows the notification received on the android device, so the admins or guide or parents can have quick look on what information they have been learnt, and it can also be used as future reference. As we can see the user asks *"How a dog bark"* giving output from speaker, same thing is also sent to the user android device as well.

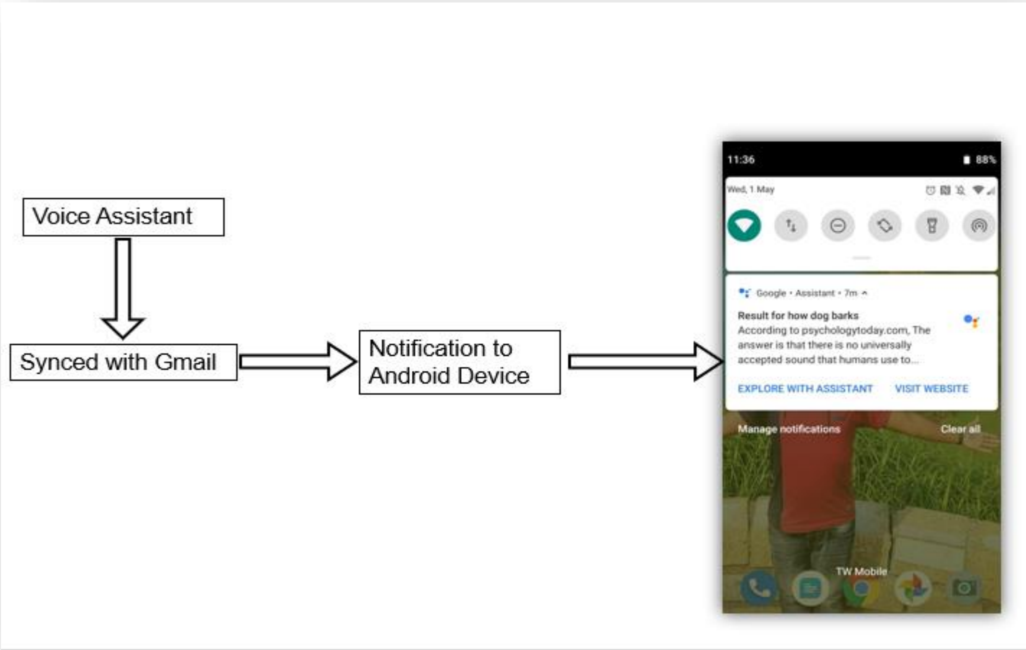

Figure 6.8 Notification to Android Device

CHAPTER 7

CONCLUSION

Blinds are very different from others, as the mind of the blind people is very different and sensitive. This research focused on investigating how to support self-learning of modern-days, blinds to recognize the objects in the outside environment. The main effort is to provide them with a Smart device, in order to help blind in their primary learning task of identifying objects just by using their own Smart Device. The proposed solution is to address the problem of creating a learning platform for the blind by combining IoT, AI and speech corpus models to provide an interactive learning experience for the particular objects in the outside environment. Compared to traditional learning, our proposed strategy eventually increases them learning capability with great memorization skills. Since our proposed system is unique and effortless learning method, the blind will be more fascinated to learn new things by using our Smart device. At the end of the paper, performance results confirm the goodness of our design choice.

REFERENCES

1. Audio guidance system for blind by Madhura Gharat ; Rizwan Patanwala Adithi Ganaparthi , 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)
2. Causo, A.; Win, P. Z.; Guo, P. S.; Chen, I. Deploying social robots as teaching aid in pre-school K2 classes: A proof-of-concept study,2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 4264-4269
3. Culjak, I.; Abram, D.; Pribanic, T.; Dzapo, H.; Cifrek, M. A brief introduction to OpenCV, 2012 Proceedings of the 35th International Convention MIPRO, Opatija, 2012, pp. 1725-1730.
4. Grlaš, V.; Marinović, M. Ecological education in preschool education through visual communication, The 33rd International Convention MIPRO, Opatija, 2010, pp. 959-963.
5. Hafidh, B.; Al Osman, H.; Arteaga-Falconi, J. S.; Dong, H.; Saddik, A.E. SITE: The simple internet of things enabler for smart homes, in IEEE Access, vol. 5, 2017, pp. 2034-2049.
6. Hear to see - See to hear: A Smart Home System User Interface for visually or hearing-impaired people by L. Ciabattoni, F. Ferracuti, G. Foresi, A. Monteriù 2018 IEEE 8th International Conference on Consumer Electronics.
7. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, 1 April 2017, pp. 664-676.
8. Li, F.; Du, Y. From AlphaGo to power system AI: what engineers can learn from solving the most complex board game, in IEEE Power and Energy Magazine, vol. 16, no. 2, March-April 2018, pp. 76-84.

9. Lewis, L. L.; Kim, Y. A.; Bey, J. A. Teaching practices and strategies to involve inner-city parents at home and in the school, Journal of Teaching and Teacher Education, vol. 27, no. 1, 2015, pp.221 - 234.
10. Mulfari, D.; Longo Minnolo, A.; Puliafito, A. Building TensorFlow applications in smart city scenarios, 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, 2017, pp. 1-15
11. Perera, C.; Liu C. H.; Jayawardena, S. The emerging internet of things marketplace from an industrial perspective: a survey, in IEEE Transactions on Emerging Topics in Computing, vol. 3, no. 4, Dec. 2015, pp. 585-598.
12. Robotic Assistant for visually impaired using Sensor Fusion by Gaurao Chaudhari, Asmita Deshpande 2017 IEEE Smart World.
13. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. ImageNet large scale visual recognition challenge, International Journa of Computer Vision (IJCV), vol. 115, no. 3, 2015, pp. 211–252.
14. Speech based text correction tool for the visually impaired Md. Nafiz Hasan Khan1, Md. Amit Hasan Arovi2, Hasan Mahmud3, Md. Kamrul Hasan4, and Husne Ara Rubaiyeat5. December 2015, IEEE 18thInternational Conference and Information Technology
15. Voice4Blind: The Talking Braille Keyboard to Assist the Visual Impaired Users in Text Messaging by Robiah Hamzah and Mohammad Izzat Mohamad Fadzil, 2016 IEEE 4th international Conference.
16. Yang, G.; Xie, L.; Mantysalo, M.; et al., A Health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box, in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, Nov. 2014, pp. 2180-2191.
17. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for smart cities, in IEEE Internet of Things Journal, vol. 1, no. 1, Feb. 2014, pp. 22-32.