GU-EYE-DE

A Project

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

In

Computer Science

By

Darshan Jethwa

2018

SIGNATURE PAGE

PROJECT: GU-EYE-DE

AUTHOR: Darshan Jethwa

DATE SUBMITTED: Fall 2018

Department of Computer Science

Dr. Yu Sun
Project Committee Chair
Computer Sciences

_____

Dr. Abdelfattah Amamra
Computer Sciences

_____

ACKNOWLEDGEMENTS

I would like to express my gratitude to my project advisor, Dr. Yu Sun. He was always there to encourage me and support me throughout the course. He was the one who inspired me to make something which can be beneficial for human beings especially for the one who need it, and which can also solve their daily problems.

I would also like to thank the other committee member, Dr. Abdelfattah Amamra for being an excellent professor and advisor who have given his time. I really appreciate, how he was so much flexible with his schedule.

Additionally, I am grateful to all the other professors and, those with whom I have had the pleasure to work during this and other related projects. Last but not the least I would like to thank my parents and my family. They were the first ones to believe in me and have always been there for me since day one. Their constant support made my journey like a cakewalk.

Above all, I want to give all glory to God!

Thank You,

Darshan Jethwa

ABSTRACT

The human race is slowly becoming more and more dependent on technology to solve even the smallest of tasks. Nowadays, technology is a necessity in people's lives. From vibrating toothbrushes to self-driving cars, technology is infused in almost everything we do. While one can argue how technology has hindered our ability to perform simple tasks, we all can agree that it has allowed for another convenient outlet to solve a situation. As technology advances, machines have been utilized to perform complex operations such as predicting the stock price, classifying text according to the content, etc. The introduction of Artificial Intelligence, where machines perceive its environment and take actions that maximize its chance of successfully achieving its goals, is primarily responsible for the ease of technology use for individuals. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

Recently, applications have been employing Artificial Intelligence in order to make them "smarter" through their ability to provide vivid functionalities. Artificial Intelligence has been used in object classification and in providing recommendations based on the user click or interest, for instance. Even with access to this type of powerful technology, there are many untapped areas that either have not been exposed or have not been utilizing Artificial Intelligence. Specifically, there are very few fully-functioning applications that assist individuals with eye-sight impairments. What if there was an application created using Artificial Intelligence/Machine-Learning that acted like their eyes and would be able

to identify objects and guide them safely to their destination? I am looking to explore this idea further.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

1. Native Application – an android application developed specifically for one platform and can take full advantage of all the device features.

INTRODUCTION

As technology advances so does the industry. Due to the advancement in technology, lately industry is more focused on their products and platform which are powered to private sectors rather than services to the people. Especially services to disable community, industry should consider global development rather than just product or platform development for private sectors. In America according to U.S Census Bureau, there are almost 56 million people who are disabled out of which almost 13 – 20 million individuals have vision loss/ vision impairment. Globally there are 1.3 billion people who have vision impairment problem. Vision has played very important role in human life. To understand graphical information and see the difference between things vision comes into picture. Moreover, humans understand better if information is provided graphically. But here the case is different for visually impaired people.

Recently Machine Learning/ Artificial Intelligence has been introduced to humans, things around them have started to change. Artificial Intelligence has reached to a level where human is not needed to control machines, they have become self-smart by learning from data which is feeded to them. The best example you can consider is self-driving car. Also, there are many cases where machine learning is been used. Similarly, by these powerful technology and advancement, we can develop an application which can act as secondary vision for visually impaired people. Basically, an application which can observe and classify the object according to their category and help them to reach their destination by describing about the object on their way.

1.1 Problem Description

Industry is so focused in their product and platform development that they have failed to consider humans community especially disable community. Disability is bigger as well as important issue to be considered in the world. Having advanced technology, industry is failing or neglecting these communities. We can solve multiple problems by using these advanced technologies for the community especially disability like visual impairment. There are 6 important senses which human needs to sense, feel and understand the things around them. Out of 6 important sense, vision plays very vital role in human life, where you need to see and understand graphical data. Basically, humans understand better when data is represented to them in graphical form. Also, to avoid danger, admire beauty around you, understand the difference between things around you, you need to have vision.

1.2 Solution

With all the advancement in technology, where we can have self-driving car, why not vision? Vision is very important sense as other senses which humans have. It's difficult to process information easily without looking or feeling it in some way. "GU-EYE-DE" a phone application can bridge all those gaps by acting as a secondary eye for individual who have vision problems.

Basically, Gueyede is an android application which can provide all the basic functions of a human eye i.e. identifying the objects, differentiating the identified objects according to their categories. Not only that it will also help these individuals to reach to their destination safely by telling them what an identified object is.

REQUIREMENTS

2.1 Functional Requirements

- Observe and identify the object
- Classify the identified object to its category
- Create an image overlay of the identified object for better understanding
- Inform the user about the identified object in real time
- Generate amazon link for the product if needed to purchase the product

2.2 Technical Requirements

2.2.1 Database

Database are the backbone of the application. Partly they are also responsible to make application dynamic. Coming to Dynamic, the database should be scalable according to the application. It should also function in real-time. Realtime because it must fetch the image of the identified object in real time. Also, it should be scalable because the application will be having n number of images of multiple objects. With this, there will be multiple request made to the database and it should be able to serve those requests.

2.2.2 Front-End

In order to interact with the application, it is necessary to have front-end. Front-end adds the abstraction layer over the complex data which thus makes it easy for user to understand the application function. Moreover, user can control functions of application easily.

2.2.3 Service & Libraries

Key functions and all the dynamic parts of the application are handled by the services. They provide dynamicity to the application by accessing the functions offered by custom libraries.

The application should have services or API which are needed to access the library functions. It should also update user interface with the data. The libraries should also provide flexibility to overrides its own function. It should also provide flexibility to user in order to write custom function.

2.2.4 Hardware

The device should have high resolution camera in order to capture series of images. Also, it should be able to capture images in detail i.e. from brighter to darker area of the images.

The hardware of the devices should at least be capable to process the high-resolution image.

APPLICATION TYPE

There are majorly two types of application: Web and mobile. Introduction to native JavaScript application, desktop application is getting obsolete day by day. Because of this all the users are migrating their desktop application to Web or they are preferring web more than Desktop.

Even though having an advantage of being able to access Web application from any corner of the world using internet, there are many disadvantages too. Web application are more susceptible to attacks so if the communication channel between database and application is not properly established or the application is not properly secured, there are higher chances of application getting breached by attacker. Also, the sensitive information can get leaked.

To avoid these kind of issues, Mobile application are growing day by day. Their handy nature makes it easy for user to access any kind of application easy. They provide high security and rich features to the application. Also, there are two types of mobile platform:

Native and Hybrid

3.1 Native

To target specific kind of audience (Android or iOS) or to test an application on one platform and see how the Market is responding, native application comes into picture. Native application also provides rich features and smooth user experience to the user. Performance is high because of easy access to hardware and OS feature in native application. Also, you don’t have to rely on third party libraries or frameworks after every software release for performance and features. There is impeccable support from OS

Providers compared to Hybrid application. Also, developer community is huge. You can support multiple version application of the same platform by changing small piece of code. All the security features and guidelines are also provided by OS providers.

3.2 Hybrid

To have single code base for all platforms which ensures low-cost maintenance and smooth updates then it a better idea to use hybrid applications. Hybrid application are developed using multiple web technologies. Basically, they are website application disguised in a native wrapper. The performance of the application is close to native application because they use n numbers of native libraries which helps them to access hardware of the device. Basically, you must rely on third party libraries to achieve performance. Support from developer's community is not as much as native application but still they are getting popular because of their performance. Other reason for them being popular is that application releases are faster, if the application doesn't need functionality that requires native codes you can change the common functionality built on top of the native libraries and ship the application.

3.3 Database

For an application to work in real-time, you need database. There is no single application where it doesn't require database to fetch and update the user interface with the data. Not only that, database has other features too, like handle large number of requests, security, authorization, consistency in case it includes financial transaction. The database should also be scalable. Also, depended upon the application, the structure should also be

changeable. Additionally, the database should be able to update the application in real-time.

With that, there two types of database which are follows: Relational Database and Non-relational database.

3.3.1 Relational Database

Oracle and MySQL are the two types of the relational database. In relational database, the data is organized in forms of one or more tables which are subdivided in rows and columns. Each row may have uniquely identifying key. Columns in table are also knows as attribute while rows are defined as records or tuples. These tables are relational i.e. there is logical connection between different tables and they are established based on their interaction. Relational databases are used where data needs to be structured, consistent, and, constraints are implemented strictly. These types of databases are majorly used in financial and business application. Major issue of these kind of databases is that they become slow as the database gets larger or userbase increases. With that, the scaling can be tedious task.

3.3.2 Non-Relational Database

As the name implies, these kinds of database are non-relational. In non-relational database, the data is stored in form documents which have collection and not tables. The format of the data is JSON thus it becomes easy to parse in web and mobile application. The data can be easily searched compared to relational database. Database is easily scalable. Because of

this, many applications prefer using these types of database where it requires only one server initially but can grow exponentially. There are many types of non-relational database but out of which firebase is the most renowned one in mobile application. The database itself is real-time. Meaning the updates in database are pushed in real time to the apps. These kinds of databases provides great scalability with impeccable pricing. Also, there are few databases which provide backend APIs to access data so mobile application developer has to concentrate only on the application development and not the backend framework. Security can be implemented on the database side itself as firebase provides rules for that. Firebase is also offline, meaning the APIs itself stores data for some time on device itself so when the device loose connection, changes can be made locally and pushed to the cloud database when the connection is on.

ARCHITECTURE

After considering all the functional and technical requirement, the application was developed using Native platform and non-relational database. Before discussing about the architecture lets go through the technological stack and libraries used for developing the application.

4.1 Android

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has also developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, also they have specific user interface. Also, variants of Android are used on game consoles, digital cameras, PCs and other electronics.

4.1.1 Features of Android

4.1.1.1 Interface

Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard. Game controllers and full-size physical keyboards are supported via Bluetooth or USB. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user.

Internal hardware, such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

Android devices boot to the homescreen, the primary navigation and information "hub" on Android devices, analogous to the desktop found on personal computers. Android homescreens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as a weather forecast, the user's email inbox, or a news ticker directly on the homescreen. A homescreen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on Google Play and other app stores can extensively re-theme the homescreen, and even mimic the look of other operating systems, such as Windows Phone.

Along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be "pulled" down to reveal a notification screen where apps display important information or updates. Notifications are "short, timely, and relevant information about your app when it’s not in use", and when tapped, users are directed to a screen inside the app relating to the notification. Beginning with Android 4.1 "Jelly Bean", "expandable notifications" allow the user to tap an icon on the notification in order for it to expand and display more information and possible app actions right from the notification.

4.1.1.2 Application

Applications which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language. Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API). The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application package) file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices.

4.1.1.3 Memory Management

Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and

automatically closing inactive processes, starting with those that have been inactive for the longest amount of time

4.2 TensorFlow

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

4.2.1 Features of TensorFlow

4.2.1.1 Deep Flexibility

It provides tools to assemble graphs for expressing diverse machine learning models. New operations can be written in Python and low-level data operators are implemented using in C++.

4.2.1.2 True Portability

It runs on CPUs, GPUs, desktop, server, or mobile computing platforms. That make it very suitable in several fields of application, for instance medical, finance, consumer electronic, etc.

4.2.1.3 Connect Research and Production

TensorFlow allows industrial researchers a faster product prototyping. It also provides academic researchers with a development framework and a community to discuss and support novel applications.

4.2.1.4 Auto-Differentiation

This is a key feature within the machine learning community. Gradient based machine learning algorithms benefit from automatic differentiation capabilities. As a TensorFlow user, you define the computational architecture for your predictive model, combine it with your objective function, and just add data to test your machine learning model.

4.2.1.5 Language Options

Python and C++. However, currently other APIs are being developed, for example a Ruby API.

4.2.1.6 Maximize Performance

It allows you to make the most of your installed hardware. Freely assign compute elements of your TensorFlow graph to different devices and let TensorFlow handle the copies.

4.3 Firebase

Firebase is a Backend-as-a-Service—BaaS—that started as a YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform. With that, it not only provides the Realtime database services but also other services like as follows:

4.3.1 Analytics

Firebase Analytics is a cost-free app measurement solution that provides insight into app usage and user engagement

4.3.2 loud messaging

Firebase Cloud messaging is a cross-platform solution for messages and notifications for Android, iOS, and web applications.

4.3.3 Authentication

Firebase Auth is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google (and Google Play Games). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

4.3.4 Storage

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

4.3.5 Hosting

Firebase Hosting is a static and dynamic web hosting service that launched on May 13, 2014. It supports hosting static files such as CSS, HTML, JavaScript and other files, as well as support through Cloud Functions. The service delivers files over a content delivery network (CDN) through HTTP Secure (HTTPS) and Secure Sockets Layer encryption (SSL). Firebase partners with Fastly, a CDN, to provide the CDN backing Firebase Hosting. The company states that Firebase Hosting grew out of customer requests; developers were using Firebase for its real-time database but needed a place to host their content.

4.3.6 ML Kit

ML Kit is a mobile machine learning system for developers launched by Google. ML Kit API's feature a variety of features including text recognition, detecting faces, scanning barcodes, labelling images and recognizing landmarks. It is currently available for iOS or Android developers. The API's can be used on-device or on cloud.

4.3.7 Crashlytics

Crash Reporting creates detailed reports of the errors in the app. Errors are grouped into clusters of similar stack traces and triaged by the severity of impact on app users. In addition to automatic reports, the developer can log custom events to help capture the steps leading up to a crash. Before acquiring Crashlytics, Firebase was using its own Firebase Crash Reporting.

4.3.8 Performance

Firebase Performance provides insights into an app's performance and the latencies the app's users experience.

4.3.9 Firebase Test Lab for Android and iOS

Firebase Test Lab for Android and iOS provides cloud-based infrastructure for testing Android and iOS apps. With one operation, developers can initiate testing of their apps across a wide variety of devices and device configurations. Test results—including logs, videos, and screenshots—are made available in the project in the Firebase console. Even if a developer hasn't written any test code for their app, Test Lab can exercise the app automatically, looking for crashes.

4.4 Glide

Glide is a fast and efficient open source media management and image loading framework for Android that wraps media decoding, memory and disk caching, and resource pooling into a simple and easy to use interface. Glide supports fetching, decoding, and displaying video stills, images, and animated GIFs. Glide includes a flexible API that allows developers to plug in to almost any network stack. By default, Glide uses a custom HttpUrlConnection based stack, but also includes utility libraries plug in to Google's Volley project or Square's OkHttp library instead.

Glide's primary focus is on making scrolling any kind of a list of images as smooth and fast as possible, but Glide is also effective for almost any case where you need to fetch, resize, and display a remote image.

4.5 Application working

Considering the requirements (Technical and Functional) and above technology, below is the application structure which is as follows:

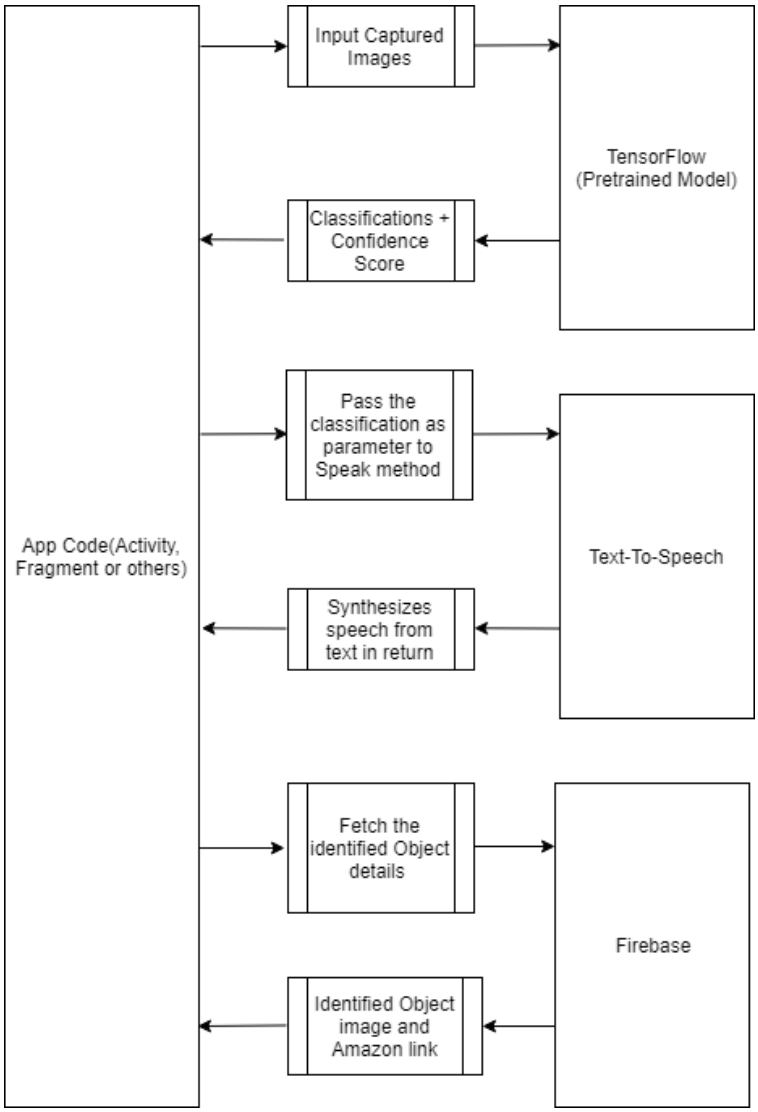

Figure 4.5: Application architecture

In the above figure, the application captures series of images using mobile camera. Later it converts the image into certain format and size using image bitmap function. Later the images are feeded to the TensorFlow pretrained model (Inception V1). The model

computes and tries to classify the object by applying various layers of CNN. The classifications and confidence score are provided in form result to the application. The Application later sends the classification class to the TextToSpeech method in order to synthesis it to speech. Once that is completed and overlay of the object is created by using Glide for the same classified class. Also, with image overlay, the amazon link is also generated which will redirect the user to amazon website in an android WebView enabling user to purchase product. This is done using firebase database.

4.5.1 The Model

The pretrained model which is used for classifying the series of inputted image is Inception model. It is one of the types of CNN (Convolution Neural Network) Architecture. Before we get to the Inception model, lets know the basic working of CNN.

4.5.1.1 CNN

CNNs have wide applications in image and video recognition, recommender systems and natural language processing. In simple words, CNNs are like neural networks, which are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs. Now let’s understand how the architecture of convolution looks like.

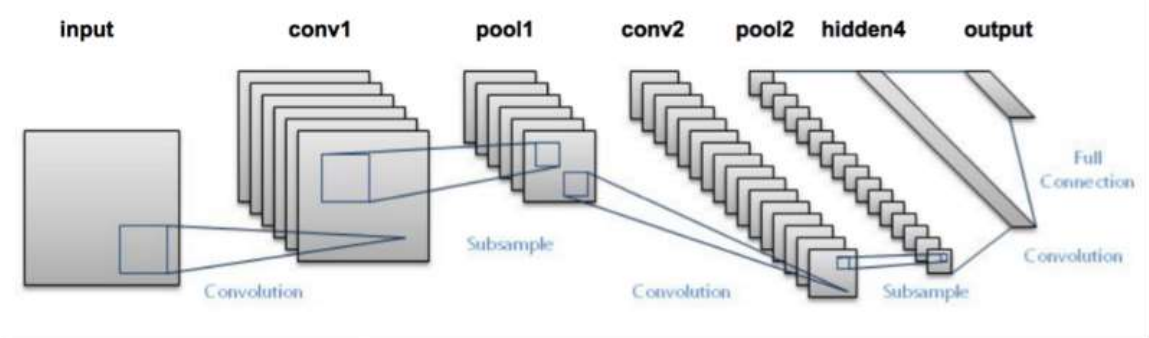

Figure 4.5.1.1: Architecture of CNN

In the above figure, the CNN consists of different layers and we`ll go through each of them in detail.

4.5.1.1.1 The Convolution Layer

Consider an example of 6*6 image and we define weight a matrix which extracts the feature from the image as shown below.

INPUT IMAGE

18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT

1	0	1
0	1	0
1	0	1

429

Figure 4.5.1.1.1a: Initial convolution output part 1

We have initialized weight as 3*3 matrix. The weight is then run across or slided over the input image matrix to give a convolved output. The value 429 above, is obtained by the adding the values obtained by element wise multiplication of the weight matrix and the

highlighted 3*3 part of the input image as shown in the above image.

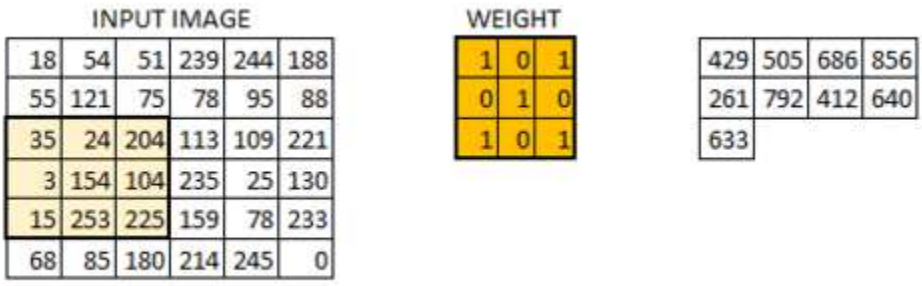

Figure 4.5.1.1.1b: Initial convolution output part 2

The process is carried out until and unless all the pixels are covered at least once as shown in the above figure.

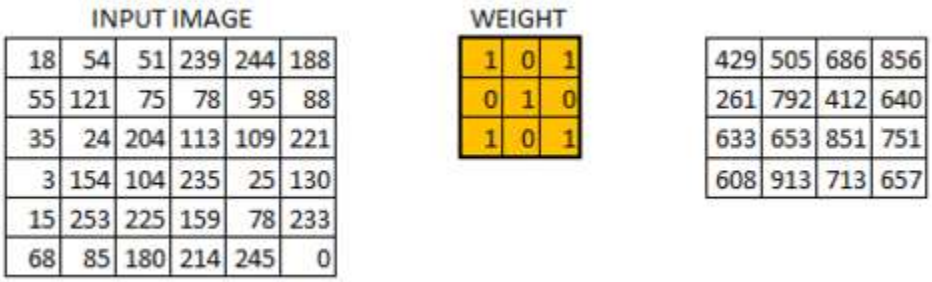

Figure 4.5.1.1.1c: Final convolution process output

Finally, the 6*6 image is converted into a 4*4 image as shown in the above figure. Think of weight matrix like a paint brush painting a wall. The brush first paints the wall horizontally and then comes down and paints the next row horizontally. Pixel values are used again when the weight matrix moves along the image. This basically enables parameter sharing in a convolutional neural network.

Now let’s see how this looks like in a real image.

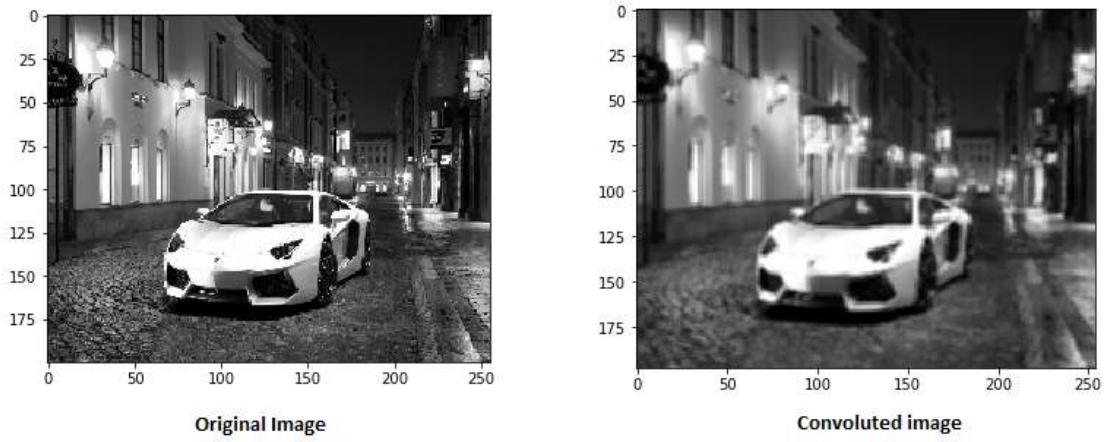

Figure 4.5.1.1.1d: Real convolution example

The weight matrix behaves like a filter in an image extracting particular information from the original image matrix. A weight combination might be extracting edges, while another one might a particular color, while another one might just blur the unwanted noise.

The weights are learnt such that the loss function is minimized similar to an MLP. Therefore, weights are learnt to extract features from the original image which help the network in correct prediction. When we have multiple convolutional layers, the initial layer extract more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more complex and more suited to the problem at hand.

4.5.1.1.1.1 The concept of stride and padding

As we saw above, the filter or the weight matrix, was moving across the entire image moving **one** pixel at a time. We can define it like a hyperparameter, as to how we would want the weight matrix to move across the image. If the weight matrix moves 1 pixel at a time, we call it as a stride of 1. Let’s see how a stride of 2 would look like.

INPUT IMAGE

18	54	51	239	244
55	121	75	78	95
35	24	204	113	109
3	154	104	235	25
15	253	225	159	78

WEIGHT

1	0	1
0	1	0
1	0	1

429

Figure 4.5.1.1.1.1a: Initial Strides process part 1

INPUT IMAGE

18	54	51	239	244
55	121	75	78	95
35	24	204	113	109
3	154	104	235	25
15	253	225	159	78

WEIGHT

1	0	1
0	1	0
1	0	1

429	686
633	

Figure 4.5.1.1.1.1b: Initial Strides process part 2

INPUT IMAGE

18	54	51	239	244
55	121	75	78	95
35	24	204	113	109
3	154	104	235	25
15	253	225	159	78

WEIGHT

1	0	1
0	1	0
1	0	1

429	686
633	412

Figure 4.5.1.1.1.1c: Final strides process output

As you can see the size of image keeps on reducing as we increase the stride value as shown in the above images. Padding the input image with zeros across it solves this problem for us. We can also add more than one layer of zeros around the image in case of higher stride values.

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

Figure 4.5.1.1.1.1d: Zero padding around image

We can see how the initial shape of the image is retained after we padded the image with a zero. This is known as **same padding** since the output image has the same size as the input.

0	0	0	0	0	0	0	0
0	18	54	51	239	244	188	0
0	55	121	75	78	95	88	0
0	35	24	204	113	109	221	0
0	3	154	104	235	25	130	0
0	15	253	225	159	78	233	0
0	68	85	180	214	245	0	0
0	0	0	0	0	0	0	0

WEIGHT

1	0	1
0	1	0
1	0	1

139

Figure 4.5.1.1.1.1e: Initial convolution process on zero padded image

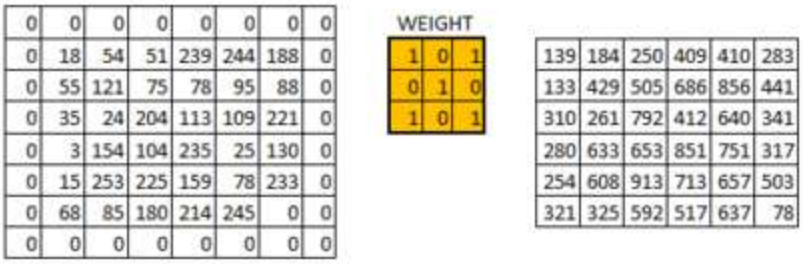

Figure 4.5.1.1.1.1f: Final output of convolution process on zero padded image

In the above figure we can see that we have retained more information from the borders and have also preserved the size of the image.

4.5.1.1.1.2 Multiple filters and the activation map

The depth dimension of the weight would be same as the depth dimension of the input image. The weight extends to the entire depth of the input image. Therefore, convolution with a single weight matrix would result into a convolved output with a single depth dimension. In most cases instead of a single filter (weight matrix), we have multiple filters of the same dimensions applied together. The output from each filter is stacked together forming the depth dimension of the convolved image. Suppose we have an input image of size 32*32*3. And we apply 10 filters of size 5*5*3 with valid padding. The output would have the dimensions as 28*28*10 as shown in the below figure.

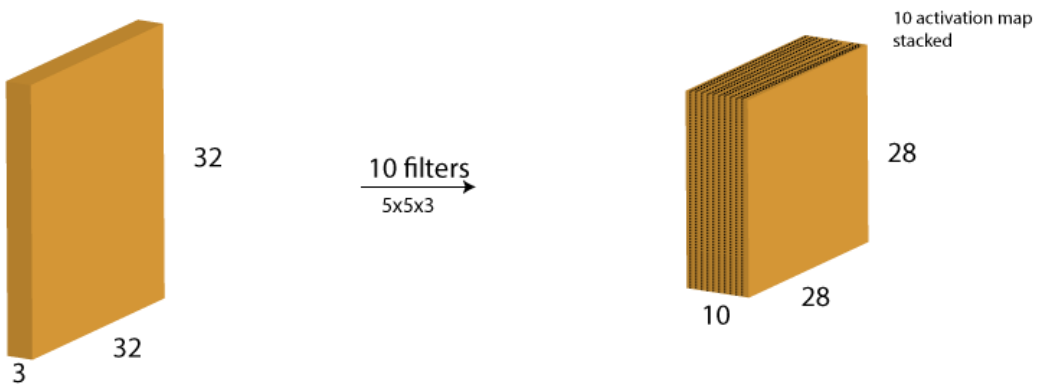

Figure 4.5.1.1.2: Convolution layer output.

4.5.1.1.2 The Pooling Layer

There are cases when images are too large, we would need to reduce the number of trainable parameters. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is performed in order to reduce the spatial size of the image. Pooling is done independently on each depth dimension; therefore, the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling.

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

792	856
913	851

Figure 4.5.1.1.2: Max pooling example

In the above image, we have taken stride as 2, while pooling size also as 2. The max operation is applied to each depth dimension of the convolved output. As you can see, the 4*4 convolved output has become 2*2 after the max pooling operation.

Let's see how max pooling looks on a real image.

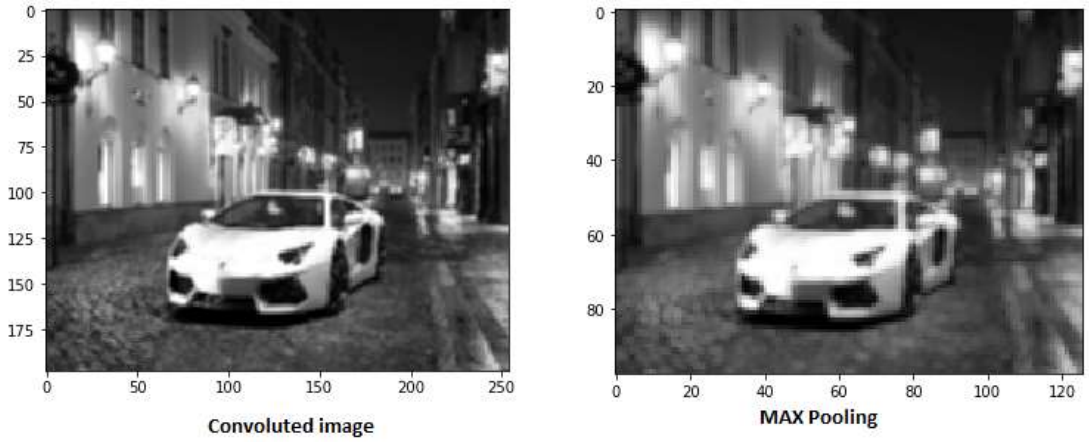

Figure 4.5.1.1.2a: Real Max pooling example

As you can see, convoluted image is taken, and then max pooling is applied on it. The max pooled image still retains the information that it's a car on a street. If you look carefully, the dimensions if the image have been changed basically, they are halved. This helps to reduce the parameters to a great extent.

4.5.1.1.2.1 Output dimensions

Three hyperparameter would control the size of output volume.

4.5.1.1.2.1.1 The number of filters

The depth of the output volume will be equal to the number of filter applied. Remember the output from each filter will form an activation map. The depth of the activation map will be equal to the number of filters.

4.5.1.1.2.1.2 Stride

If there Stride value is high, large number of pixels are considered and thus output produced is smaller

4.5.1.1.2.1.3 Zero padding

This helps to preserve the size of input image. If a single zero padding is added, a single stride filter movement would retain the size of the original image.

We can apply a simple formula to calculate the output dimensions. The spatial size of the output image can be calculated as([W-F+2P]/S) +1. Here, W is the input volume size, F is the size of the filter, P is the number of padding applied and S is the number of strides. Suppose we have an input image of size 32*32*3, we apply 10 filters of size 3*3*3, with single stride and no zero padding.

Here W=32, F=3, P=0 and S=1. The output depth will be equal to the number of filters applied i.e. 10.

The size of the output volume will be ([32-3+0]/1)+1 = 30. Therefore, the output volume will be 30*30*10.

4.5.1.1.3 The Output layer

After multiple layers of convolution and padding, we would need the output in the form of a class. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. However, to generate the final output we need to apply a fully connected layer to generate an output equal to the number of classes we need. It becomes tough to reach that number with just the convolution layers. Convolution layers generate 3D activation maps while we just need the output as whether or not an image belongs to a particular class. The output layer has a loss function like categorical cross-entropy, to compute the error in prediction. Once the forward pass is complete the backpropagation begins to update the weight and biases for error and loss reduction.

Finally putting every layer together, we get something as shown in the above architecture image.

- We pass an input image to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolution layer extract relevant features from the input image to pass further.

- Each filter shall give a different feature to aid the correct class prediction. In case we need to retain the size of the image, we use same padding(zero padding), other wise valid padding is used since it helps to reduce the number of features.

- Pooling layers are then added to further reduce the number of parameters

- Several convolution and pooling layers are added before the prediction is made. Convolutional layer help in extracting features. As we go deeper in the network more specific features are extracted as compared to a shallow network where the features extracted are more generic.

- The output layer in a CNN as mentioned previously is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network.

- The output is then generated through the output layer and is compared to the output layer for error generation. A loss function is defined in the fully connected output layer to compute the mean square loss. The gradient of error is then calculated.

- The error is then backpropagated to update the filter(weights) and bias values.

- One training cycle is completed in a single forward and backward pass.

4.5.1.2 Inception Model

Now with that the inception model is based on CNN architecture. The model is comprised of a basic unit referred to as an "Inception cell" in which we perform a series of convolutions3. at different scales and subsequently aggregate the results. In order to save computation, 1x1 convolutions are used to reduce the input channel depth. For each cell, we learn a set of 1x1, 3x3, and 5x5 filters which can learn to extract features at different

scales from the input. Max pooling is also used, albeit with "same" padding to preserve the dimensions so that the output can be properly concatenated.

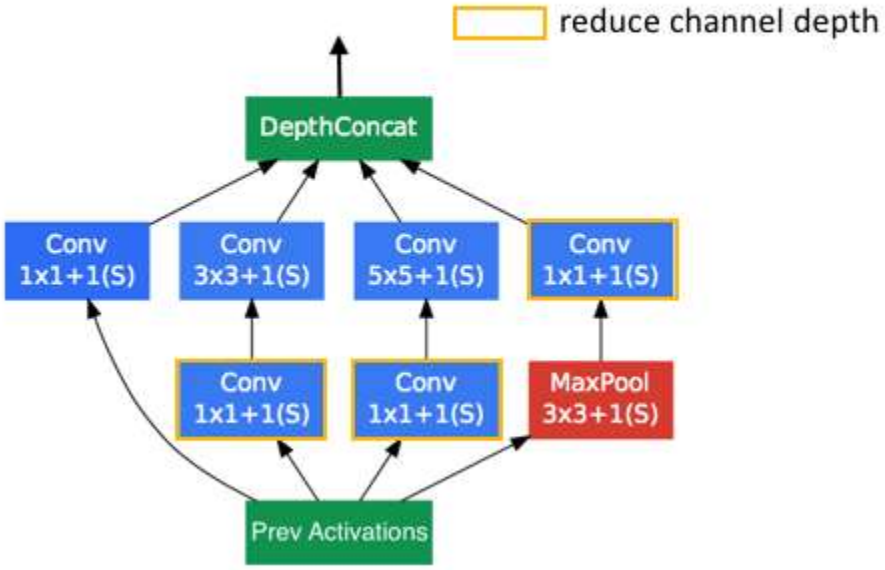

Figure 4.5.1.2a: Single Inception Cell

These researchers published a follow-up paper which introduced more efficient alternatives to the original Inception cell. Convolutions with large spatial filters (such as 5x5 or 7x7) are beneficial in terms of their expressiveness and ability to extract features at a larger scale, but the computation is disproportionately expensive. The researchers pointed out that a 5x5 convolution can be more cheaply represented by two stacked 3x3 filters.

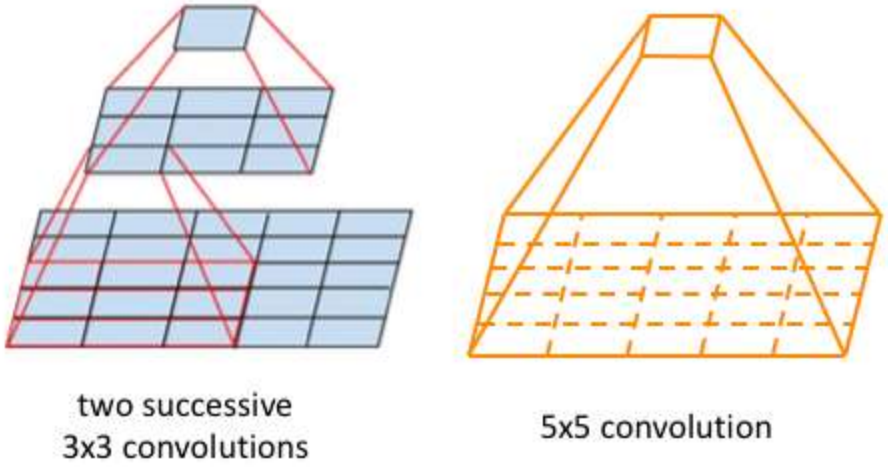

Figure 4.5.1.2b: Two successive 3*3 and 5*5 filters

As shown in the above image, 5×5×c filter requires 25c parameters, while two 3×3×c filters only require 18c parameters. In order to most accurately represent a 5x5 filter, we shouldn't use any nonlinear activations between the two 3x3 layers. However, it was discovered that "linear activation was always inferior to using rectified linear units in all stages of the factorization." It was also shown that 3x3 convolutions could be further deconstructed into successive 3x1 and 1x3 convolutions.

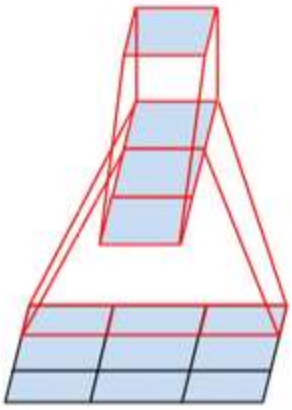

Figure 4.5.1.2c: Successive 3*1 and 1*3 convolutions

Generalizing this insight, we can more efficiently compute an n×n convolution as a 1×n convolution followed by a n×1 convolution.

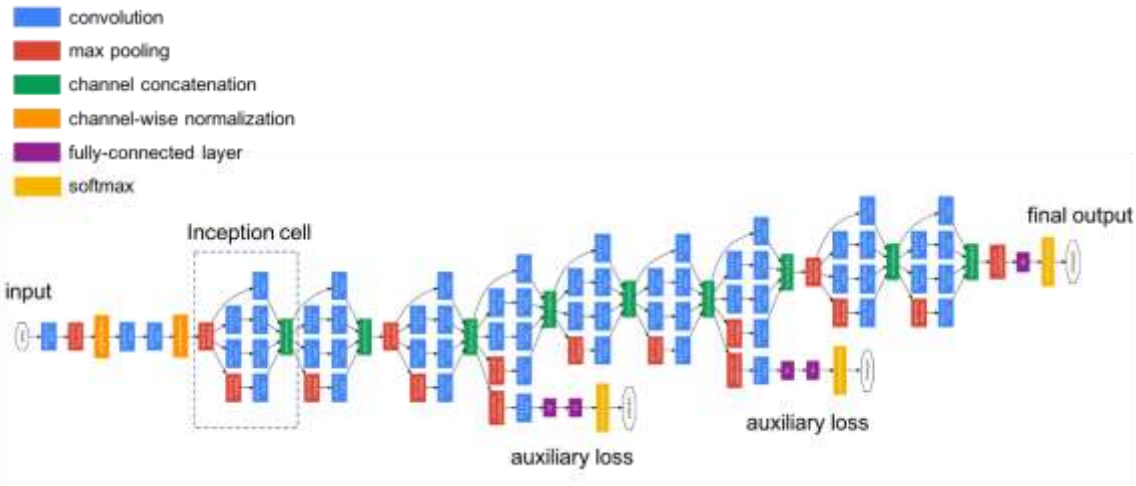

Figure 4.5.1.2d: Inception model Architecture

In order to improve overall network performance, two auxiliary outputs are added throughout the network. It was later discovered that the earliest auxiliary output had no discernible effect on the final quality of the network. The addition of auxiliary outputs

primarily benefited the end performance of the model, converging at a slightly better value than the same network architecture without an auxiliary branch. It is believed the addition of auxiliary outputs had a regularizing effect on the network.

USER INTERFACES

5.1 Image capture and classifying activity

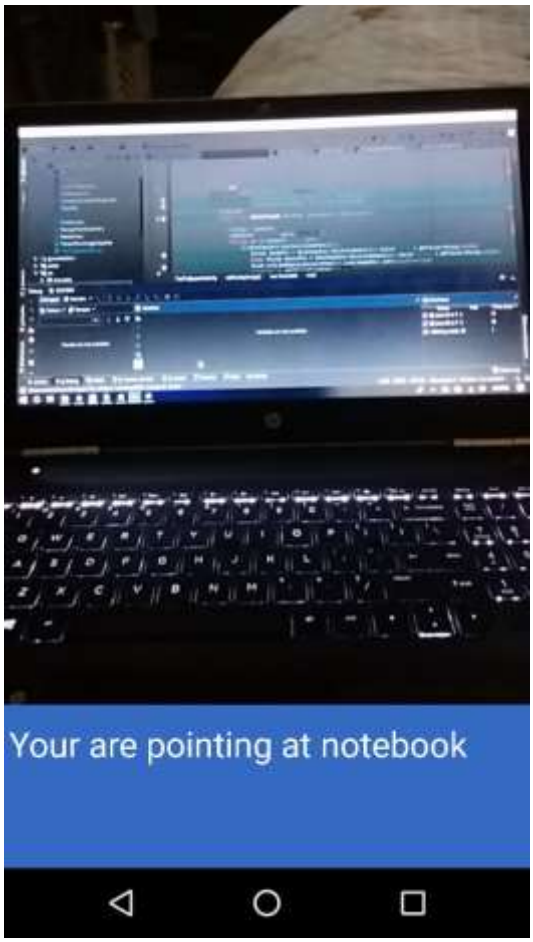

Figure 5.1: MainActivity capturing images

The above image is of main activity where the camera captures images continuously and is feeded to pretrained model that is inception model. The model converts the image and applies different computation, classifies the image to appropriate class and also tells the user to where he pointing at. The class which is obtained from the model is displayed in form of text under TextView as shown in the above image. Also, the text which is obtained

from the model is synthesis to Speech using TextToSpeech module.

5.2 Image overlay and Amazon link activity

Figure 5.2: Image overlay Activity

In this activity, as the object is identified via pretrained model, image overlay is created by glide framework over it. Before all this process, the identified class obtained from the model is parsed to firebase and accordingly image path from the firebase database is fetched. Not only that, the amazon product link is obtained from the firebase respectively.

5.3 Amazon webview activity

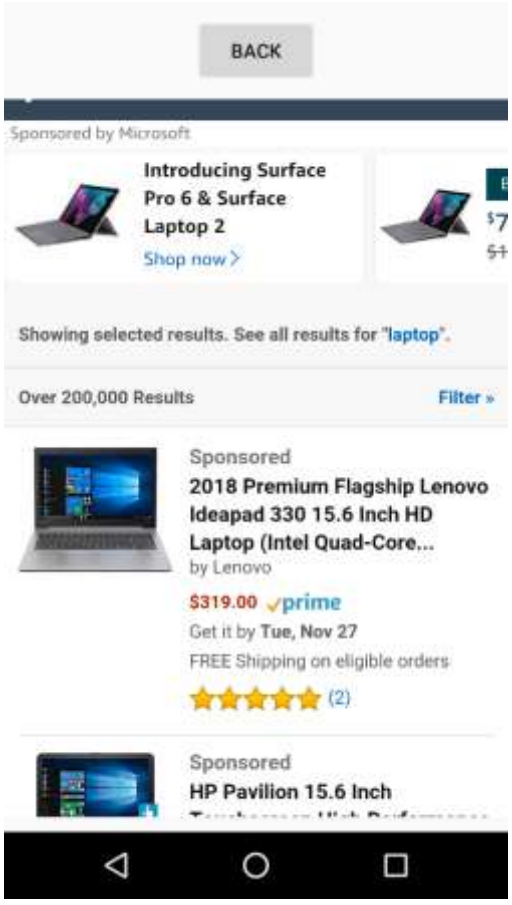

Figure 5.3: Amazon Webview Activity

When clicked on the image overlay, an activity with webview opens up as shown in the above screenshot. The activity displays amazon website. The amazon website fetches the similar product which is been clicked which thus enables the user to purchase the product. There is back button feature where on clicked, it takes you back to the previous activity.

FUTURE WORK

Even after developing application, there is always room for improvement or additional features. Current prototype only provides image classification, image overlay, text to speech synthesis and Amazon link of the product in order to purchase the product. There are few more feature and improvement which will be added in future releases. Some of them are listed below.

- Update/Train the Image Classification model

Current classification model is been trained to multiple categories, but the classification result is not as accurate it should be. Upgrading the current model or training the current model with more datasets can improve the accuracy of the model to identify the objects and classify the objects.

- Make application more Intuitive and Accessible

Currently in the prototype, there are few operations which are carried out manually for example: to navigate to amazon webview activity, you must click on the image overlay. Also, for purchasing the item from amazon, you have to click many links. All these processes can be improved by using artificial intelligence where it can user where it wants to buy the product, or the user can command to the system so that I can carry out most of the operations for the users.

- Add augmented reality navigation feature

Now a days augmented reality has been booming a lot in the software industry. It would be great to include this feature. It can be combined with navigation system where it can help user to reach their destination safely by detecting objects, giving directions in Realtime and also describing about the environment while walking on the way.

- Integration with the 3rd party application

In future releases, 3rd party application such as uber, yelp, lift etc. will be provided so that if user wants to book a ride or order a food through the application, it will be easy for user.

CONCLUSION

With more advancement in technology and tools, Industry is more inclined toward product and platform development that they are forgetting to consider people especially disabled community. Considering todays application, where features are paid more attention to rather than accessibility. Also, these applications tend to solve general problems while they should be focusing on bigger problems. Looking at the current figures for disability around the world, it is so high especially, figure of people who are visually disabled. Vision plays very important role as other senses in human life. Without vision it becomes difficult for one to understand things around them. Also, humans understand easily if information provided is in graphical form.

Thus, to overcome this problem, the above application can be useful in many ways where it can act as a secondary vision for people who have difficulty in seeing objects around them. The application is built in such a way that it will classify and tell them what the object is and where they are pointing at. Additionally, the application is not specifically focused to these individuals but it does consider wide user base, so it provides imageoverlay and Amazon product purchase feature where user can see the image of the identified object on the screen which will help those user who have difficulty in seeing. Plus, if the user wants to purchase products then, user can purchase the product by clicking on the image which will redirect them to the Amazon product purchase activity. It is the activity where all the amazon products are displayed thus giving them a variety of options. By this, the user doesn't have to specifically go to the website or other application to purchase the product which saves their time an energy.

REFERENCES

1. "Hybrid Vs Native Mobile Apps - The Answer Is Clear". *YML | Products And Experiences With Lasting Impact*, 2018, https://www.ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/. Accessed 22 Nov 2018.
2. "What Is Firebase? – How To Firebase". *How To Firebase*, 2018, https://howtofirebase.com/what-is-firebase-fcb8614ba442. Accessed 22 Nov 2018.
3. "Documentation | Android Developers". *Android Developers*, 2018, https://developer.android.com/docs/. Accessed 22 Nov 2018.
4. "Google". *Google.Com*, 2018, https://www.google.com/. Accessed 22 Nov 2018.
5. Learning, Deep, and Architecture demystified. "Architecture Of Convolutional Neural Networks (Cnns) Demystified". *Analytics Vidhya*, 2018, https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/. Accessed 22 Nov 2018.
6. "Firebase". *En.Wikipedia.Org*, 2018, https://en.wikipedia.org/wiki/Firebase. Accessed 22 Nov 2018.
7. "Tensorflow · Delft Students On Software Architecture: DESOSA 2016". *Delftswa.Gitbooks.Io*,2018, https://delftswa.gitbooks.io/desosa2016/content/tensorflow/chapter.html. Accessed 22 Nov 2018.
8. Alammar, Jay. "Supercharging Android Apps With Tensorflow (Google's Open Source Machine Learning Library)". *Jalammar.Github.Io*, 2018,

http://jalammar.github.io/Supercharging-android-apps-using-tensorflow/. Accessed 22 Nov 2018.

9. "Tensorflow Lite - Knowledge Transfer". *Knowledge Transfer*, 2018, http://androidkt.com/tenserflow-lite/. Accessed 22 Nov 2018.
10. "Bumptech/Glide". *Github*, 2018, https://github.com/bumptech/glide. Accessed 22 Nov 2018.
11. "The Best Explanation Of Convolutional Neural Networks On The Internet!". *Medium*, 2018, https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8. Accessed 22 Nov 2018.
12. "Common Architectures In Convolutional Neural Networks.". *Jeremy Jordan*, 2018, https://www.jeremyjordan.me/convnet-architectures/. Accessed 22 Nov 2018.
13. "Image Recognition | Tensorflow". *Tensorflow*, 2018, https://www.tensorflow.org/tutorials/images/image_recognition. Accessed 26 Nov 2018.
14. Catherinecheney. "The Technology Sector Could Make or Break Disability Inclusion." *Safety & Security Manager | Devex*, 9 Jan. 2017, www.devex.com/news/the-technology-sector-could-make-or-break-disability-inclusion-89387. Accessed 26 Nov 2018.
15. "Facts and Figures on Adults with Vision Loss - American Foundation for the Blind." *American Foundation for the Blind*, www.afb.org/info/blindness-statistics/adults/facts-and-figures/235#demographics. Accessed 26 Nov 2018.