**DEERWALK INSTITUTE OF TECHNOLOGY**

**Tribhuvan University**

**Institute of Science and Technology**



**MOBILE APPLICATION TO HELP VISUALLY IMPAIRED PEOPLE USING MACHINE LEARNING APPROACH**

**A FINAL PROJECT REPORT**

**Submitted to**

**Department of Computer Science and Information Technology**

**DWIT College**

***In partial fulfillment of the requirements for the Bachelor’s Degree in Computer Science and***

***Information Technology***

Submitted by

Shreejan Pandey(15499/074)

Aashish Sapkota(15468/074)

[02-24-2022]

**DWIT College DEERWALK INSTITUTE OF TECHNOLOGY**

# SUPERVISOR’S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by SHREEJAN PANDEY, AASHISH SAPKOTA entitled **“MOBILE APPLICATION TO HELP VISUALLY IMPAIRED PEOPLE USING MACHINE LEARNING AND FLUTTER”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Bijay Babu Regmi

Supervisor

DWIT College

Deerwalk Institute of Technology

**DWIT College DEERWALK INSTITUTE OF TECHNOLOGY**

# STUDENT’S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than that listed here have been used in this work.

…………………………………………

Aashish Sapkota

Date:- 2-24-2022

………………………………………...

Shreejan Pandey

Date:- 2-24-2022

**DWIT College DEERWALK INSTITUTE OF TECHNOLOGY**

# LETTER OF APPROVAL

This is to certify that this project prepared by SHREEJAN PANDEY, AASHISH SAPKOTA

entitled **“MOBILE APPLICATION TO HELP VISUALLY IMPAIRED PEOPLE USING MACHINE LEARNING AND FLUTTER”** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ……………………………  Mr. Bijay Babu Regmi  Project Supervisor  DWIT College  Deerwalk Institute of Technology | …………………………….  Mr. Hitesh Karki  College Chef  DWIT College  Deerwalk Institute of Technology |
| ……………………………  Mr. Bijay Babu Regmi  Coordinator /Year-In-Charge  DWIT College  Deerwalk Institute of Technology | ……………………………  Mr. Lokh Nath Regmi  IOST, Tribhuvan University |

# ACKNOWLEDGEMENT

We would like to thank DWIT College for providing us an opportunity to perform this project. We express our profound gratitude and deep regards to our project supervisor, Mr. Bijay Babu Regmi for mentoring us, providing appropriate resources and encouragement throughout the completion of this project, and guiding us in the right direction.

We are really thankful because we managed to complete this project within the time assigned by our coordinator. We would also like to thank our parents for the constant support and their inspirations. We are highly indebted to DWIT College and their staff for constantly allowing us to use the equipment and all the help we needed. This assignment cannot be completed without the effort and co-operation from our group members. Last but not the least, we would like to express our gratitude to our friends and respondents.

Shreejan Pandey (15499/074)

Aashish Sapkota (15467/074)

Date:- 02-24-2022

**ABSTRACT**

The world has experienced major growth in smart devices. Along with the growth in smart devices there has been significant advancement in AI and Machine learning. This has facilitated in developing various tools and technology to help each sector of human lives. The project is an attempt to help blind and visually impaired people to use their smart phones in performing basic tasks such as describing the environment, making emergency calls and even reading out the text. This attempts to make technology reachable to every individual. The rapidly growing development of mobile technology initially adversely affected the accessibility of device functionalities for visually disabled persons increasing their digital exclusion. Visually disabled persons through touch, gestures, voice commands and audio feedback can freely configure and run a set of applications available on a mobile device. The main goal of this project is to use the Machine learning algorithm to identify the environment. Another part of this system is to give the result automatically based on the recognized surrounding information. This paper describes the system that implements Machine learning and natural language processing to image captioning method, character recognition and carry some other useful functionality. As a result the system will assist visually impaired people to perform day to day activities.

**Keywords**: *CNN; RNN; API; Machine Learning; OCR; Visually Impaired;*

# Table of Contents

[SUPERVISOR’S RECOMMENDATION ii](#_Toc96543074)

[STUDENT’S DECLARATION iii](#_Toc96543075)

[LETTER OF APPROVAL iv](#_Toc96543076)

[ACKNOWLEDGEMENT v](#_Toc96543077)

[Table of Contents vii](#_Toc96543078)

[LIST OF FIGURES ix](#_Toc96543079)

[LIST OF TABLES x](#_Toc96543080)

[LIST OF ABBREVIATIONS xi](#_Toc96543081)

[CHAPTER 1: INTRODUCTION 1](#_Toc96543082)

[1.1. Overview 1](#_Toc96543083)

[1.2. Background and Motivation 1](#_Toc96543084)

[1.3. Problem Statement 1](#_Toc96543085)

[1.4. Objectives 2](#_Toc96543086)

[1.5. Scope and Limitation 2](#_Toc96543087)

[1.6. Development Methodology 2](#_Toc96543088)

[1.7. Outline 3](#_Toc96543089)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW 4](#_Toc96543090)

[2.1. Background Study 4](#_Toc96543091)

[2.2. Literature Review 4](#_Toc96543092)

[2.3. Current System 8](#_Toc96543093)

[2.4. The problem with Current System 9](#_Toc96543094)

[CHAPTER 3: SYSTEM ANALYSIS 10](#_Toc96543095)

[3.1. Requirement Analysis 10](#_Toc96543096)

[3.2. Feasibility Analysis 11](#_Toc96543097)

[3.3. Analysis 14](#_Toc96543098)

[3.3.1 Data flow diagram 14](#_Toc96543099)

[CHAPTER 4: SYSTEM DESIGN 16](#_Toc96543100)

[4.1. Design 16](#_Toc96543101)

[4.2. Algorithm Details 17](#_Toc96543102)

[CHAPTER 5: IMPLEMENTATION AND TESTING 19](#_Toc96543103)

[5.1. Implementation 19](#_Toc96543104)

[5.2. Testing 22](#_Toc96543105)

[5.3. Result Analysis 24](#_Toc96543106)

[CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION 26](#_Toc96543107)

[6.1. Conclusion 26](#_Toc96543108)

[6.2. Future Recommendation 26](#_Toc96543109)

[References 27](#_Toc96543110)

# LIST OF FIGURES

Figure1.1. Agile Methodology........................................................................................................ 3

Figure 2.1. Convolutional Neural Network Layer ..........................................................................5

Figure 2.2. Convolutional Neural Network Layers .........................................................................7

Figure 2.3. CNN Model of the System ............................................................................................8

Figure 3.1. Use Case Diagram .......................................................................................................11

Figure 3.2. Gantt chart .................................................................................................................. 12

Figure 3.3 Gantt chart ................................................................................................................... 13

Figure 3.4 Context Diagram/ DFD 0 ............................................................................................ 14

Figure 3.5 DFD level 1 .................................................................................................................. 14

Figure 3.6 DFD level 2 .................................................................................................................. 15

Figure 4.1 Activity Diagram for Environment Scan ..................................................................... 16

Figure 4.2 Process Diagram for OCR........................................................................................... 17

Figure 4.3 Process Diagram for Scanning Environment ............................................................... 18

Table 5.1 Test Cases mobile application to help Visually Impaired ............................................. 22

Figure 5.2 Prediction outcome on the console .............................................................................. 25

# LIST OF TABLES

Table 5.1 Test Cases mobile application to help Visually Impaired ............................................. 22

# LIST OF ABBREVIATIONS

API (Application Programming Interface)

CNN (Convolutional Neural Network)

NLP (Natural Language Processing)

OCR ([Optical Character Recognition)](https://www.analyticsvidhya.com/blog/2021/06/optical-character-recognitionocr-with-tesseract-opencv-and-python/)

RNN (Recurrent Neural Network)

# CHAPTER 1: INTRODUCTION

## 1.1. Overview

Accessing the situation and determining what is around us in the environment is one of key aspects of human behavior. For people with visual impaired however it is difficult to do these due to lack of sensory vision. Thus The emergence of widespread use of smart phones and tablets with touchscreens shifted the problem of digital exclusion in the area of mobile software. Development of mobile applications, especially designing the user interface accessible for a blind person is a very serious issue and this has led to the new field of research.

Thus, this paper presents the techniques to help visually impaired individuals to make their smart phones friendly to carry basic activities. With the incorporation of machine learning tools the mobile application is developed.

## 1.2. Background and Motivation

The project aims to include the blind and visually impaired in making them able to utilize the technology. This will aid in making their daily tasks easier by incorporating the latest technology. This idea led to building applications to help those individuals to perform their daily tasks easily. Hence the idea is predicted to decrease the gap created due to digital exclusion.

## 1.3. Problem Statement

* Visually Impaired people usually find it difficult to use smart phones because most of the features require visual input from the user. This makes it difficult for visual impaired people to take advantage of massive computational power available in mobile phones.
* Less number of the applications providing assistance for visually impaired people.
* The application guides the visually impaired person do activities like reading out the text, navigating through the route, and the emotion of the user using their facial

1

expression.

## 1.4. Objectives

The objectives of the project are as follows:

* To assist the blind to receive any voice notifications in their smartphones such as messages and also process voice commands.
* To help users read contents as well as provide other useful information.
* To provide emergency services by contacting the concerned unit.
* To automatically generate results.

## 1.5. Scope and Limitation

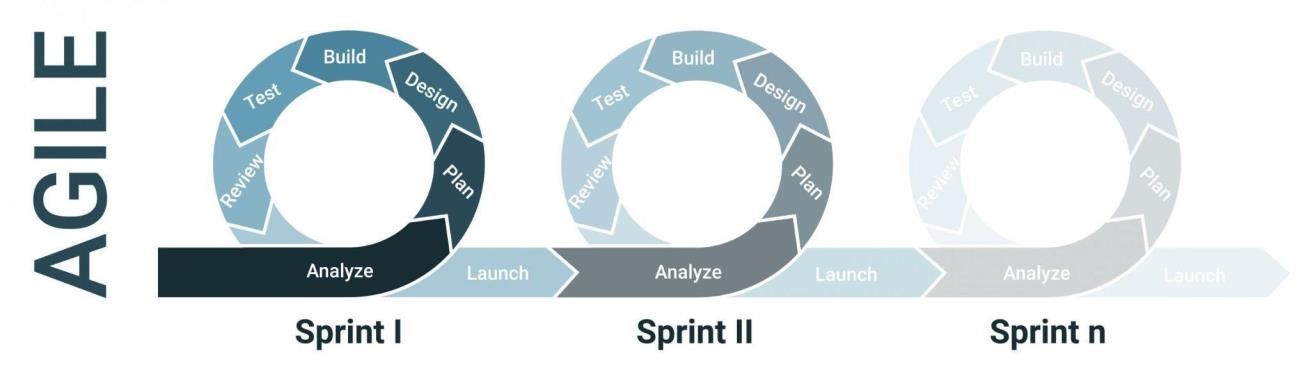
The project will be based on creating a mobile application with finding optimal solutions for various problems such as accessing the environment for visually impaired people.

The application that we have created is done with a limited time frame , and thus comes with some limitations which are.

* The application has a limited number of commands.
* There can be slight differences in what the app defines to the actual scene around the user.

## 1.6. Development Methodology

Agile methodology is used to develop the application .This approach for development uses short iterations. Initially the app with only one command was built and iteratively after each succession of the commands new commands were added. In this way, the results of each iteration are used to adjust the project plan.



**Figure1.1. Agile Methodology**

## 1.7. Outline

The outline gives an overview of the main points of your report. It clarifies the structure of your report and helps you find the correct focus for your work.

* Chapter 1 starts with the introduction of the system. We have introduced why our system is built and the condition of traditional and manual systems which are unstable. In addition, objectives of the project are explained in detail with scope and limitation of the system.
* Chapter 2 includes a brief review of studies that have been done previously by other researchers along with requirements as well as feasibility analysis of the system. The data modeling and process modeling techniques are used to give the information about the system requirement.
* Chapter 3 discusses the system design and algorithm used in our project. In system design, we discuss database schema design, interface design and process design.
* Chapter 4 describes which tools are used for our project’s purpose to make it possible. In addition, the testing is also explained.
* Chapter 5 explains about the conclusion of how our project is done, what are its contributions, what did it achieve, its findings and many more. We also discuss future enhancement of the project.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1. Background Study

The core implementation of the functionality of the application is based on Machine learning algorithms, Natural Language Processing (NLP). This implementation is accessed using API calls to the backend. Here, the application operates by providing easy user input and implementation at the backend. This way it provides our aim to make daily use of the visually impaired easy and efficient.

Various additional features simplifying the access for users with a weaker sense of sight have been implemented. The set of system facilities is also expanded by external applications. Their developers often use the available application interface (API) in a creative way and they come to interesting solutions.

## 2.2. Literature Review

The number of Smartphone users in the world has increased up to 6.378 billion which means 80.63% of the world’s population owns a Smartphone. This result is expected because smart phones have mobile apps that give a lot of benefits and help us in many ways. However, it is only beneficial for normal people and most of the application is very limited to the visually impaired people. The visually impaired people face a lot of challenges that normally sighted people take for granted [1].

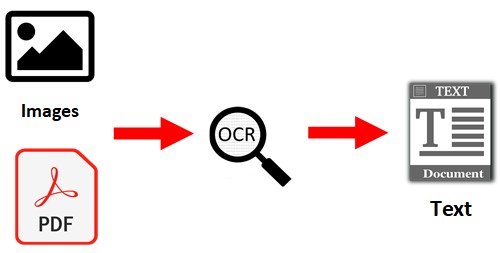
Visually impaired people face difficulties navigating outside the spaces they are usually to. Basically, physical movement is one of the biggest challenges for blind people. Traveling or even walking down a crowded street can be challenging for blind people. Besides, They must remember the location of just about any obstacle or object in their home environment, such as tables, chairs and bed which should not be moved unexpectedly and without advanced notice to avoid accidents [2].

To automate accurate result generation, CNN (convolutional neural network) and OCR (Optical Character Recognition) systems are used for character recognition. This system allows us to have fast and accurate information.

#### 2.1.1 Optical Character Recognition

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten, or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast).

Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data, and text mining. OCR is a field of research in pattern recognition, artificial intelligence, and computer vision [3].



**Figure 2.1 Convolutional Neural Network Layer**

#### 2.1.2 Convolutional Neural Network

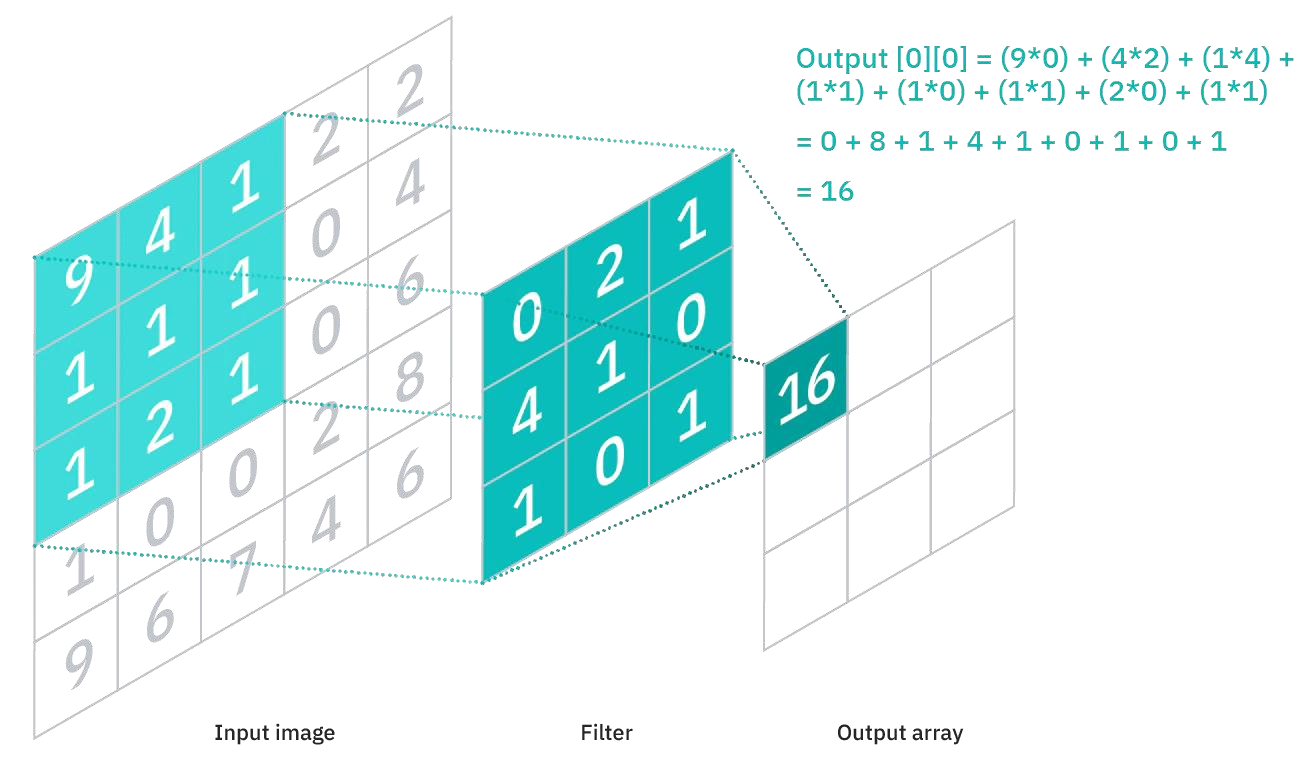
The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their sharedweights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

CNN’s are regularized versions of multilayer perceptions. Multilayer perceptron usually means fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNN’s take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extremity.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNN uses relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand engineered. This independence from prior knowledge and human effort in feature design is a major advantage.



**Figure 2.2 Convolutional Neural Network Layers**

##### 2.1.2.1 Convolutional Neural Network (CNN) Model

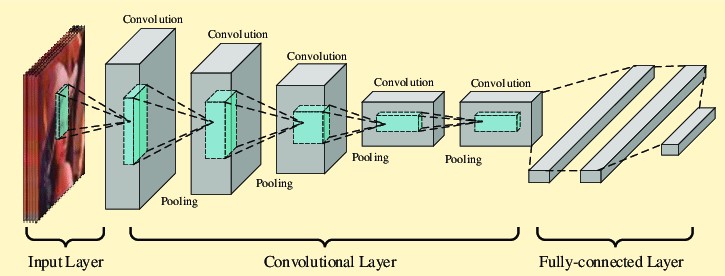
A typical convolutional neural network consists of a convolutional layer, pooling layer, and finally a fully connected layer. The convolution layers extract features; the pooling layer reduces the dimension of the image, and fully connected layers are then used for generating a prediction. The network also includes dropouts for ignoring some of the neurons.

The first layer consists of the Convolutional layer with 30 channels, a kernel of size 5x5, and a rectifier linear unit (relu) as an activation function. This layer then feeds 30 images into the second layer. This layer is accompanied by a max-pooling layer of size 2x2 to down sample, reduce parameters, computation involved in the network and create its feature map [4].

The next layer consists of another Convolution layer but with 15 channels, the kernel of size 3x3, and relu as an activation function. This layer is again accompanied by max-pooling of the same size as before.

Dropout of 0.2 is used when feeding the feature map to the flatten layer which then transforms the two-dimensional feature map into a format suitable for a fully connected network to work on. [5]

Finally, the fully connected layer consists of 3 dense layers each. The first dense layer consists of 128 neurons with relu activation function, the second dense layer consists of 50 neurons with relu activation function, and the last layer consists of 13 neurons with softmax as the activation function. These dense layers work together in tandem to classify the source image to any of the 13 characters (0, 1, …, 9, +, -, \*).



**Figure 2.3 CNN Model of the System**

Sequential recurrent neural networks (RNNs) are remarkably effective models of natural language. In the last few years, language model results that substantially improve over long-established state-of the-art baselines have been obtained using RNNs as well as in various conditional language modeling tasks such as machine translation , image caption generation, and dialogue generation . Despite these impressive results, sequential models are a priori inappropriate models of natural language, since relationships among words are largely organized in terms of latent nested structures rather than sequential surface order [6].

## 2.3. Current System

#### 2.3.1 TapTapSee

TapTapSee is an app that allows the visually impaired and blind community to accurately identify objects they encounter in their daily lives without the need for sighted assistance.

#### 2.3.2 Bespecular

The app offers a very unique and simple process to get a remote volunteer to help you out on something you need. You will and can get multiple responses to your questions. It gives you a variety of different answers to get that question answered that you want and need**.**

## 2.4. The problem with Current System

The current system consists of few apps that provide basic features such as.

1. Identifying the current currency amount.
2. Determine whether there is any obstacle or not.
3. This app does not consist of facilities to define the surrounding.

# CHAPTER 3: SYSTEM ANALYSIS

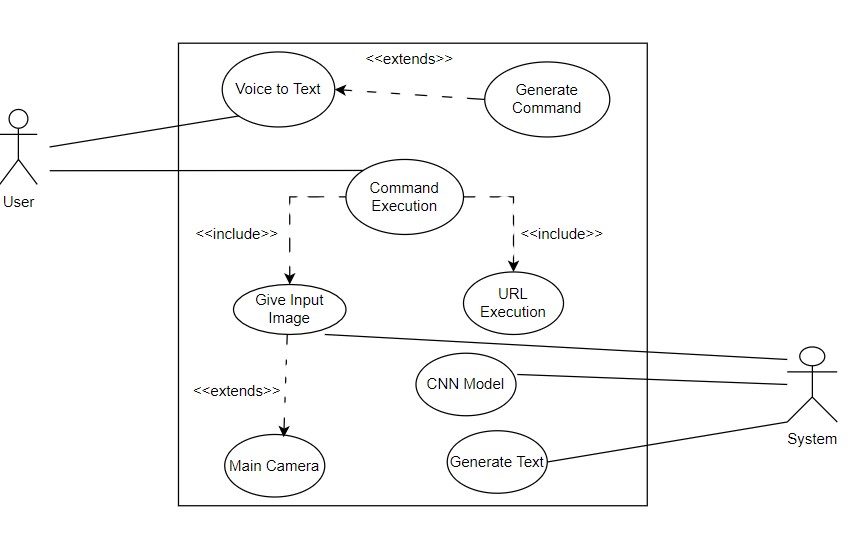
## 3.1. Requirement Analysis

The application must be able to perform basic fundamental tasks such as making a phone call, describing the events shown in the image, processing voice commands, recognizing the input facial expression in real time. These tasks are to be completed using api calls to the backend and are required to provide fast and accurate results. For the beginning phase of the application the range of environment to be described may be simpler but as the machine learning system gets more training data, the range is to be increased.

#### 3.1.1. Functional Requirement

A functional requirement specifies what the system should provide. The functional requirements are as follows

* It shall perform the image description, image to text, text to speech and input facial expression.
* It shall perform these operations in real time.
* It shall do these tasks more efficiently in terms of memory use.
* The application shall read out the text loudly.



**Figure 3.1 Use Case Diagram**

#### 3.1.2. Non-Functional Requirement

Non-functional requirements define the overall qualities or attributes of a system. So, the non-functional requirements for our system are as follows:

* The application must be easy to use.
* The application must have a high level of accuracy.
* The application must provide feedback to command.
* The application must operate on both operating systems.

## 3.2. Feasibility Analysis

#### 3.2.1. Technical Feasibility

The software used consists of python for Machine learning, flask to host the python backend, flutter to create mobile applications and visual scripting for programming. Using above programs ensuring the product can be built and elegantly integrated into any existing system. Hence the project is technically feasible.

#### 3.2.2. Operational Feasibility

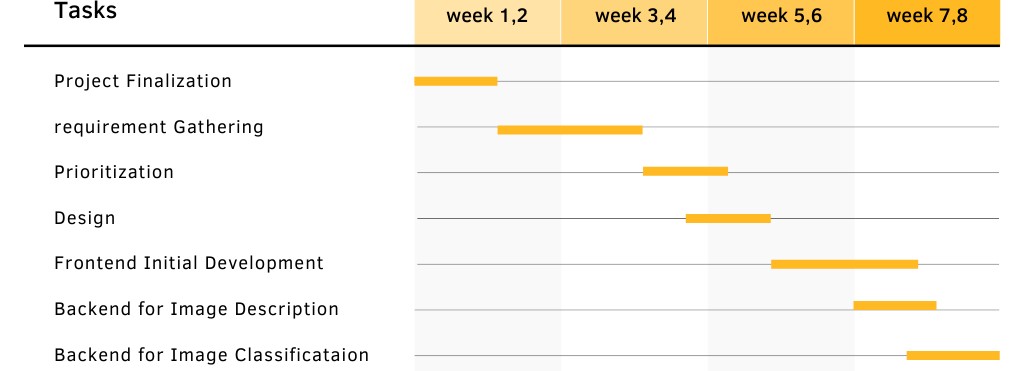
This project will have minimal interface and the interaction between user and system will be done via voice command. Thus this system can be operated by most of the users. Hence the project is operationally feasible.

#### 3.2.3. Economic Feasibility

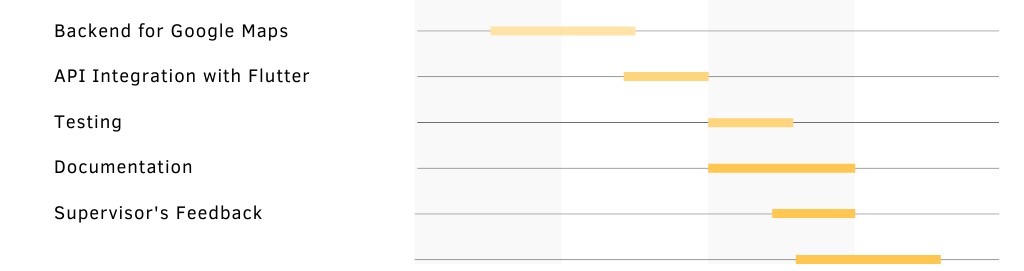
Each and every requirement for the project is available online and through various research papers, ejornels, no external expenses are needed. Hence the project is economically feasible.

#### 3.2.4. Schedule Feasibility

The project is estimated to be completed in 6 months with the first prototype of the project developed within the 1st month.



**Figure 3.2 Gantt chart**

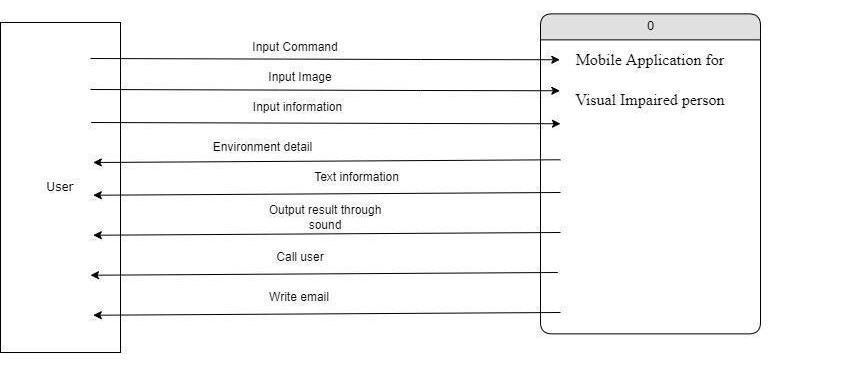


**Figure 3.3 Gantt chart**

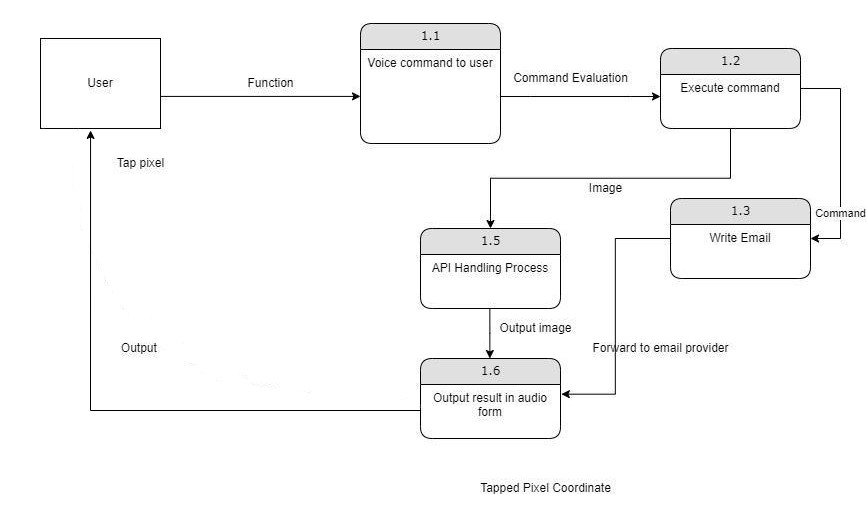
Hence the project is schedule feasible.

## 3.3. Analysis

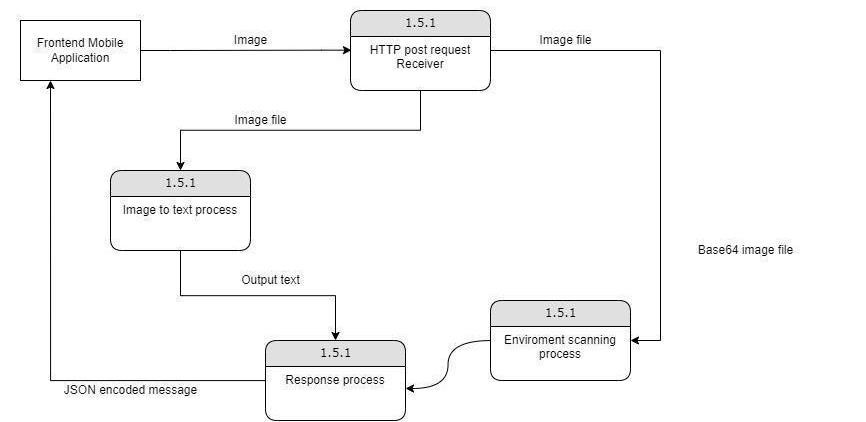
### 3.3.1 Data flow diagram



**Figure 3.4 Context Diagram/ DFD 0**



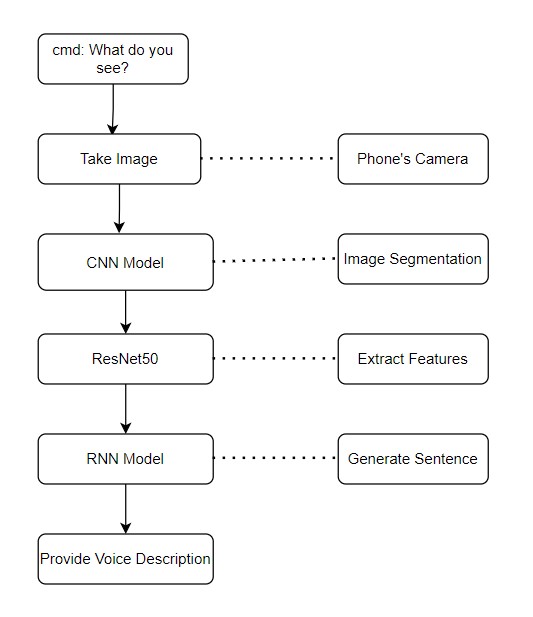
**Figure 3.5 DFD level 1**



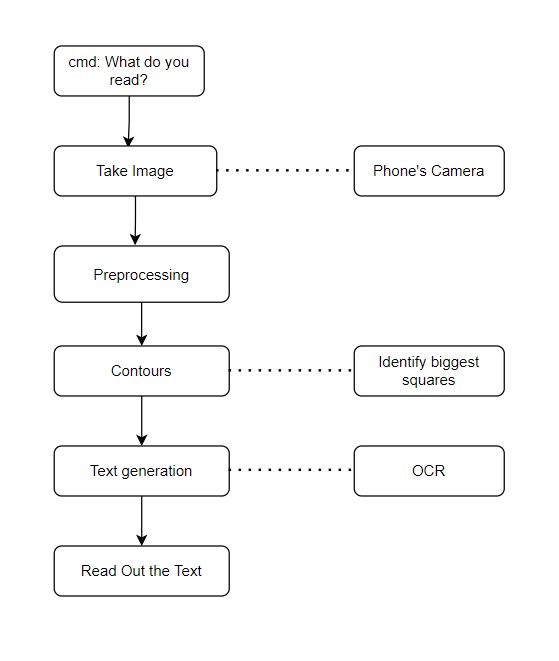
**Figure 3.6 DFD level 2**

# CHAPTER 4: SYSTEM DESIGN

## 4.1. Design

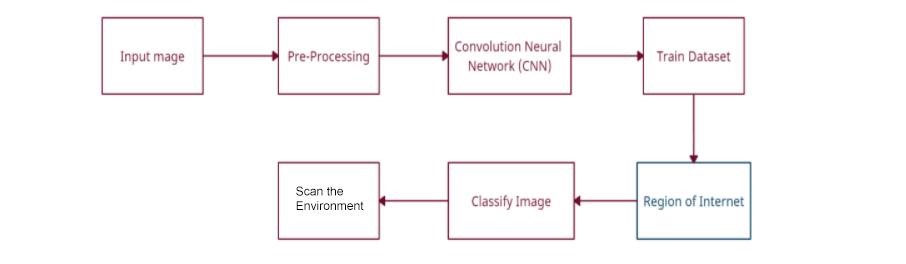


**Figure 4.1 Activity Diagram for Environment Scan**



**Figure 4.2 Process Diagram for OCR**

## 4.2. Algorithm Details



**Figure 4.3 Process Diagram for Scanning Environment**

To classify the image, CNN model is being implemented using Convolution Neural Network (CNN). The CNN model takes in the preprocessed image to fit into the algorithm. The algorithm takes in the input parameters in the form of a matrix and then calculates the dot product of the pixels. After calculating the area of interest then the image is classified and the features of the image are extracted. These features are then sent to RNN which is then processed for NLP which generates English sentences which are sent back to the user.

For OCR the image is first subjected to a pre-processing that changes the image into a black and white image. With the use of available contours the square shape object inside the image is identified and only the biggest square shape object is taken from the image. Then this is subjected to OCR which gives the text as output and sent back to the user.

## CHAPTER 5: IMPLEMENTATION AND TESTING

## 5.1. Implementation

An image-captioning model was trained in PyTorch on Microsoft's COCO dataset to understand images in many different environments and produce accurate textual descriptions of the image. The model consists of two parts, the pre-trained ResNet50 convolutional neural network and a recurrent neural network that is one layer deep. The ResNet50 model is used to extract features from the image, and the RNN takes those features and generates a textual description of the image.0

For the implementation of the application, there were some design complexities regarding language and actual understanding of how blind and visually impaired face in their everyday life. The application was implementing multiple API interactions and their final integration. In the frontend part, Flutter application provides the compatibility in android and ios operating systems. Flutter application makes an API call to the backend where the result of the request is given in the form of response to the Flutter application.

**5.1.1 Data Collection**

The COCO dataset was used to train the model that detects human presence in a room for this project. COCO is a large-scale object detection, segmentation, and captioning dataset [8]. As the name suggests, images in COCO dataset are taken from everyday scenes thus attaching “context” to the objects captured in the scenes.

#### 5.1.2. Tools Used

CASE tools:

1. Draw.io is used for UML designs.
2. PhotoShop was used to modify images.

Client Side:

1. Flutter framework is used to build mobile applications.
2. Dart was used as the primary language for building the application.
3. Flutter packages for implementing functionality.

Server Side:

1. A Flask API built using python 3.9 for hosting systems locally.
2. Tesseract OCR used to process images to determine the character within the image.
3. OpenCV-contrib 4.5.4.60 used image processing functionalities such as grayscale the input image, apply Gaussian Blur, utilize adaptive threshold values, detection and drawing of contours in the edge-detected image etc.

#### 5.1.2. Implementation Details of Modules

**Image captioning model**

An image-captioning model was trained in PyTorch on Microsoft's COCO dataset to understand images in many different environments and produce accurate textual descriptions of the image. The model consists of two parts, the pre-trained ResNet50 convolutional neural network and a recurrent neural network that is one layer deep. The ResNet50 model is used to extract features from the image, and the RNN takes those features and generates a textual description of the image. The model can be found in model.py [1]

**Mobile Application**

The mobile app was created to offer an intuitive, affordable, and accessible tool for the visually impaired to use. The app has one screen with a button that fills the entire screen. On the press of this button, any valid command can be provided to the application. There are two variants of the command that can be sent to the application. One of these command type processes provides the result in the client side while another required to send an image is captured through the camera and this image is sent to the api using http.post to be run through the OCR and image-captioning model. The api then returns the generated sentence of the image back to the app where it is converted to audio for the user to hear. The app is not published, but it does work for iOS and Android because Flutter was used.

**Web Application**

The web app functions as an api for the mobile app and was built using Flask. The web app takes the image, preprocesses it to be run through the OCR or image-captioning model, and runs it through the image captioning model to get a textual description of the image or through OCR model to extract textual information. The description is then sent back to the app to read aloud to the user.

**For the Server Side:**

**Filename: main.py**

This python file handles all the incoming requests into the Flask API. The incoming requests may be requests for describing the environment or reading the text on a page . Based on the type of request, it then routes the request into one of the following key functions.

1. read\_image() :

The image on base64 is sent to a ocr\_core(image) function which then has scripting convert the image to text. This text is then forwarded back to the user as a response which is then turned to sound.

1. scan\_enviroment() :

The image on base64 is decoded and transformed and evaluated. This process then generates the image description which is then sent as response to the user.

1. get\_prediction(image, encoder, decoder)

This function takes the image, encoder and decoder and returns the sentence. This function is called from the scan environment.

## 5.2. Testing

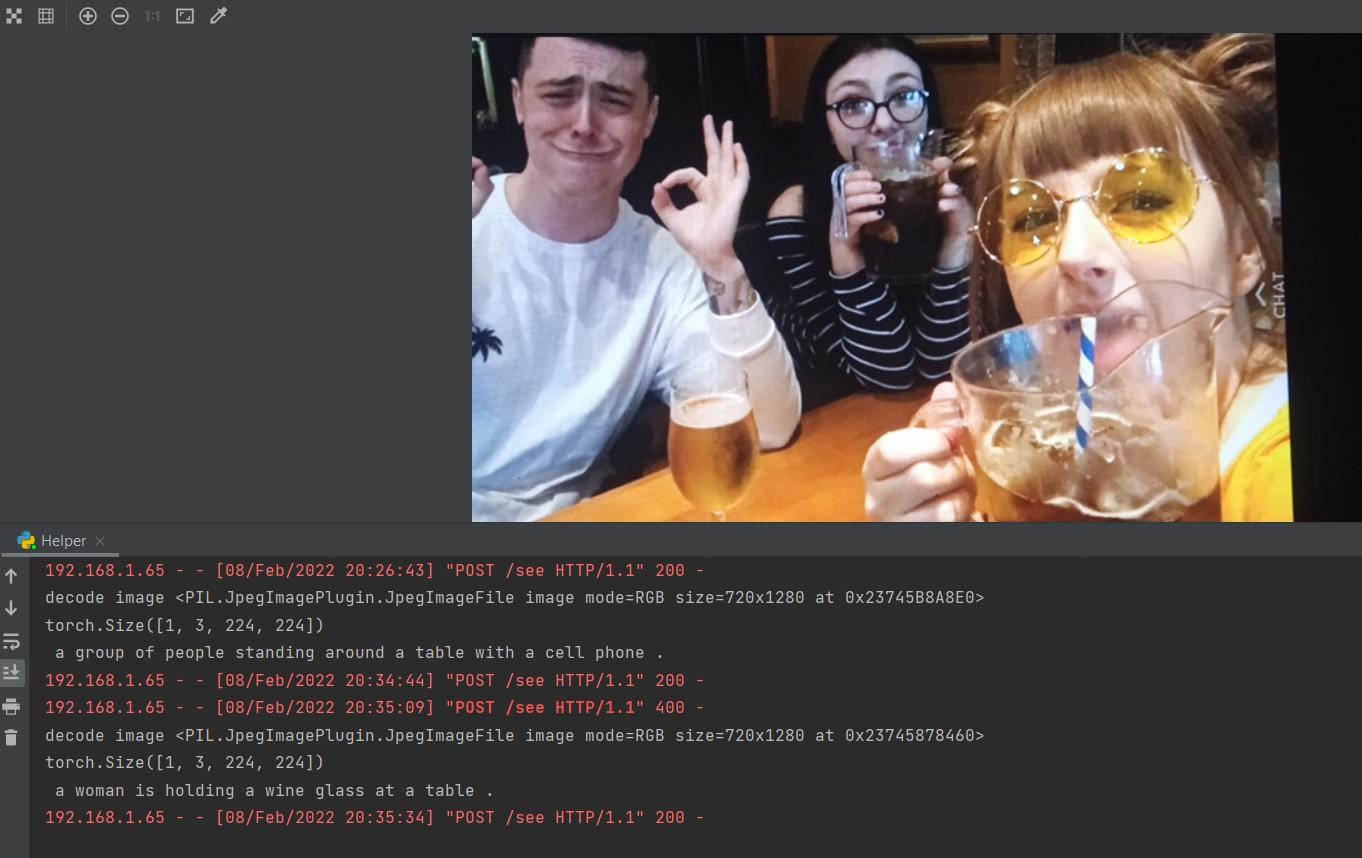
**Table 5.1 Test Cases mobile application to help Visually Impaired**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Test Scenario | Test Case | Test Steps | Expected  Result | Actual  Result | Status |
| T001 | Verify Start  App  Functionality | To check whether home page is displayed when app is opened for first time | 1.Open the App | Start page should be displayed | Start Page is displayed | TRUE |
| T002 | Verify Start  Page  Functionality | To check whether main button works | 1. Open the App 2. Click anywhere in the screen | User should be allowed to give command | User is allowed to give  command | TRUE |
| T003 | Verify Voice input functionality | To check whether audio input is taken or not. | 1. Open the App 2. Give voice command | Voice command processed | a voice  command  is written on the screen. | TRUE |
| T004 | Verify  Choose  Method  Functionality  1 | To check whether the app chooses a given command. | 1. Open the App 2. Give predefined voice command | Give required result for given  command | App gives the targeted result set. | TRUE |
| T005 | Verify  Choose | To check whether read | 1. Open the App 2. Give command “ | Users should get the text to | Text is converted | TRUE |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Method  Functionality  2 | text command is functional | read this” | speech. | to voice. |  |
| T006 | Verify open given link Functionality | To check given link is opened or not | 1. Open the App 2. Give Command 3. “Open   Youtube.com” | User should be redirected to Youtube.com | User is redirected to  youtube.co m | TRUE |
| T007 | Verify whether command are received by backend | To check if backend flask app is accepting any command | 1. Open the App 2. Click anywhere on the screen. 3. Give voice commands from available commands that access the backend. 4. See the console log on backend to see if the command is received | Status code should be provided to verify receiving of command. | Status code was output on receiving command. | TRUE |
| T008 | Verify whether model is fitted or not | Test dataset to verify and validate model | 1. go do backend code 2. Build the machine learning model 3. Test the model with datasets 4. 4. Verify with sample datasets | The data should fit the designed model | Model responded to sample datasets | TRUE |
| T009 | Verify Connection with backend | To check whether the command is flown from frontend to backend or not. | 1. Open the App 2. choose any of the command 3. Check for the console response 4. Check for console message 5. Check for log message | Intermediate result in console | Connection between frontend and backend is established. | TRUE |
| T010 | Error handling | To check whether any error is thrown while running an Image captioning model. | 1. Open the App 2. choose “What do you see “ Command. 3. Wait for the command to reach the backend. 4. Look for the console log on pycharm | No error should be thrown and return type should be print | Predicted result is printed | TRUE |
| T011 | Verifying text recognition capability | To check whether text can be read in dim light condition or not | 1. Open the App 2. Choose “What do you read “ Command. 3. Wait for the command to reach the backend. 4. Look for the console log on pycharm | Inaccurate reading with non existing words | Predicted result is not printed | FALSE |
| T012 | Verify environmental scanning capability | To check whether text can be scanned in presence of low light | 1. Open the App 2. Choose “What do you read “ Command. 3. Wait for the command to reach the backend. 4. Look for the console log on pycharm | Inaccurate scanning of the text written | Predicted result was not printed | FALSE |

## 5.3. Result Analysis

In order to test the model, the dataset was divided into 9:1 (train: test) after image processing. For training the dataset, a total of 118,287 images were taken. For testing the data, a total of 40,670 images were taken. The whole dataset was divided into different batches of size 32 each and was trained for 3 epochs which gave model accuracy of 0.8502 and validation accuracy of 0.8623 as shown in the fig below. Now, in the console itself when few images were predicted after training, they gave satisfying outcomes.



**Figure 5.2 Prediction outcomes on the console**

After integrating, we checked whether the model was integrated correctly or not by taking an image at random from the internet. The application showed the correct result of the image which was taken for prediction. Likewise, different images were taken for prediction in which result was accurate for some images while some images showed incorrect results. Similarly, we took the image clicked from the camera. The clicked photo was taken as an input and the application showed the correct result of the image which was taken for prediction.

Regression testing was also performed to verify every feature of the application. The menu button clicking functionalities, camera functionalities, image gallery functionalities all of these tests were carried out in system testing.

The overall running of the application can be characterized as smooth and functional. However, it should be noted that the smoothness and fluidity of the application depend on the system, the light source when taking the picture as well as how frequent the object in the image appears in the coco dataset. Sometimes the application crashes because the integrated model takes more time to load.

# CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

## 6.1. Conclusion

This project defines an application that modifies the way visually impaired people use mobile applications by allowing its users to use smart applications with the use of voice commands. The main focus within the application lies on creating optimal functionalities that help in carryout daily activities and many more. With the use of trained data, these functionalities thus provide an appropriate solution to existing problems like determining the environment, and the surrounding structures. By analyzing the train data and the accuracy of the output, this approach has shown how ML can be used to navigate problems existing with visually impaired people.

This solution can be further used in other fields that include AI to determine its surroundings, and helper robots for differentiable people.

Further research is needed to determine the optimal approach that would drastically decrease the computational power for the process while increasing the accuracy to make the process completely offline which will further enhance the capabilities of the application.

## 6.2. Future Recommendation

Currently the system is only limited to only a few commands. In the future we would like to implement more commands that would help users have more improvement on their life with the use of smart devices. Likewise the system currently cannot fully identify in detail about the object it sees when there are multiple objects that don't frequently occur in coco Dataset.

This can be enhanced with having a more trained dataset. For OCR the current system isn’t able to work correctly on bright light when the light is directly subjected to the camera. This can be improved by improving the image detection technique.

# References

# 

1. IBM Cloud Education, "https://www.ibm.com/cloud/learn/recurrent-neuralnetworks#:~:text=A%20recurrent%20neural%20network%20(RNN,data%20or%20time%20se ries%20data.," *Recurrent Neural Networks*.
2. Ash Turner, "How Many People Have Smartphones Worldwide (Dec 2021).," p. 15, 2021.
3. Devender R. Banda Nora Griffin-Shirley, "A Survey on the Use of Mobile Applications for People Who Are Visually Impaired," p. 17, 2017.
4. "https://pdf.abbyy.com/learning-center/what-is-ocr/," *optical character reader*, 2020.
5. Wojciech Zaremba, Ilya Sutskever, Oriol Vinyals"Recurrent Neural Network Regularization.," *2017*.