DOCUMENT READING SYSTEM FOR BLIND PEOPLE (“READING EYE”)

19-20-J 17

Final Report of Image & Table Reading Module

IT16079328
W.R.P.Fernando

B.Sc. Special (Honors) Degree in IT Specializing in

Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2020

DOCUMENT READING SYSTEM FOR BLIND PEOPLE (“READING EYE”)

19-20-J 17

W.R.P.Fernando
IT16079328

Dissertation Submitted in Partial Fulfilment of The Requirements for The B.Sc. Special Honors Degree in Information Technology

B.Sc. Special (Honors) Degree in IT Specializing in

Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2020

DECLARATION

I declare that this is my own work and this system requirement specification does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

……………………………………….. ….…...…………….……………

Signature of the supervisor Date

The above candidates are carrying out research for the B.Sc. Special (Hons) degree in SE Dissertation under my supervision.

……………………………………….. ….…...…………….……………

Signature of the supervisor Date

……………………………………….. ….…...…………….……………

Signature of the co-supervisor Date

ABSTRACT

Everyone says reading is important, but why? Reading is so important because it develops our thoughts, gives us endless knowledge and lessons to read while keeping our minds active. Reading books can help us learn, understand and makes us smarter. Besides, it helps to reduce our stress. When we read, we focus on the written word which helps to relieve our mind off the pressures of the day. By pulling our mind away from the stress at hand, we can relax and let the stress melt away. Along with all these benefits, we need reading to complete day-to-day tasks too. From the email that were sent by your boss to a bill for something you purchased from store when you are heading home, you have to read. So, all in all without the ability of reading we are pretty much futile.

For the most part, we take it for granted. As sighted people, we do not think much about it because we have no problems with reading a bill with our own eyes. However, for a blind user, the case is entirely different as it is not an easier task for them to read, even with Braille.

Braille is a system that was introduced centuries ago. Although it is old and probably not compatible with modern man, it has been the most effective method for reading and writing for a blind individual so far. Even so, it is only for text. Because Braille is an alphabet and it only consists of characters like letters, punctuations and so on. Using Braille methods to re-create charts or images or any other graphical content would be extremely difficult or perhaps impossible. Hence, it has some limitations when using with modern books.

Along with Braille system, there are numerous reading systems available in the market that were implemented using current technologies. They too have the same problem just like Braille. These systems are capable of reading only text-based contents and nothing else. So, all of these solutions are far from perfect. Therefore, this research study was mainly focused on developing a perfect reading system for visually impaired people, which is capable of reading more than just sentences and paragraphs. Our system is named as ‘Reading Eye’ because it behaves like a human eye and competent just the same. It can read not only paragraphs, but also charts, images, tables and equations. Plus, it can detect those graphical contents accurately and faster, which will then be used to read by other features. And we have enhanced the security of the application as an overall by implementing secure client-server connection and encryption methods. In addition to that, the mobile application will take less time to read a book or any other document for that matter. It has so many other features which would be extremely useful to blind individuals. Reading Eye will help many individuals to overcome the barriers and enjoy reading as they deserve. However, as the output of the application, a narration of the book or the document will be played by the app itself for the user to listen.

Furthermore, the purpose of this report is to provide answers for the project related questions such as the research problem, the need of addressing that problem, the solution for that problem and the achievements of the project. The report details the main areas related to the project. Under the first section of the report, it details out the problem to be addressed, background context, and the research gap. The second section addresses the literature, methodology and research findings. The third section contains the results and discussions. The last section describes the conclusion of the project.

Keywords: Virtual Reader, Assistive Technology, Computer Vision, Deep Learning, Natural Language Processing, Image processing, Security, Encryption

ACKNOWLEDGEMENT

First of all, I would like to thank the Sri Lanka Institute of Information Technology (SLIIT) for providing the necessary resources to complete the research project successfully.

I extend my sincere gratitude to the Project Supervisors Ms. Suranjini de Silva for providing her invaluable guidance; feedback and time throughout the course of our research project for complete success.

Next, I would like to thank our co-supervisor Dr. Anuradha Jayakody, for his excessive support and guidance to achieve our project goals.

Also, I would like to thank the Project Coordinator Mr. Janaka Wijekoon, who provided the necessary information and deliver the lecture materials to complete the project successfully.

I would also like to thank all the members of the academic staff and those individuals who give their assistance and support.

Last but not least, I would like to thank our parents and family members for their support to make our project a success.

Finally, I would like to thank all the team members of the project for their contribution and effort towards achieving the project goals and objectives.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Background

Reading is more than just a pastime for bookworms. It is a skill everyone uses on a daily basis. A work email, text message, street sign, or even a status on Facebook and Twitter, all require you to read. In addition to these daily chores, many people read to gain additional knowledge, to be entertained, to reduce the stress of the day or even to motivate themselves to move forward in life. Reading helps us to discover ourselves. It broadens our horizons, allowing us to wonder wherever in the world without moving a foot. When you are reading a book, you are living in another world with people who have already become familiar to yourself and feels like you knew them all your life. It is such a wonderful experience to have and only a true reader will understand how beautiful it is. You meet new people, you travel around the world, you experience things you have never before experienced yourself, all just by reading an amazing book. There are so many books in the world, which means you can have that experience again and again. However, sighted people have been fortunate enough to experience and enjoy it tremendously without much trouble. But for the unsighted people, it is not the same. As they cannot perceive with their eyes. But that should not be an obstacle. Instead there should be alternatives to read books for visually impaired people.

Over the years, countless methods were introduced to enable blind people to read and write independently. Out of them, the only thing that succeeded to this day is Braille system. Braille is not a language. It's an alphabetic writing system. It was developed by Louis Braille. He lost his sight because of an accident when he was a child. In 1824, at the age of fifteen he developed the code system and published it. Initially he developed that code for French alphabet as he was a French [1].

Braille system uses raised dots to represent the letters of the print alphabet. It also includes symbols to represent punctuation, mathematics and scientific characters, music, computer notation, and foreign languages [1]. Ever since the introduction of Braille system, blind people were able to read and write without anyone else's support. Braille characters are read by moving one hand or both hands left to right along each line and the index fingers do the reading. The average reading speed of a person who uses Braille is about 125 words per minute, but greater speed can be up to 200 words per minute. The average reading speed of most adults(sighted) is around 200 to 250 words per minute [2].

Undoubtedly, there is a significant difference between the two average reading speeds. It is apparent that the average reading speed of a person who uses Braille is much slower. Of course, it is reasonable. For one thing Braille requires the person to touch and comprehend the words instead of seeing. Therefore, it is a time consuming and tedious task. Although it could be easier for those who are much experienced with it, a newly blind person could find it frustrating. Particularly, for children who are still not good at reading Braille and adults who became blind later in their lives have hard time adjusting to this method to do the reading and writing. It is a common conception that it is much easier to learn a language such as English, French, and others especially when you are a kid. Mostly because, a child absorbs as much as possible and he learns quickly. Later in life we have much trouble learning a new language. The same goes for Braille too. Though Braille is not a language, it still requires some brainpower to learn and years to dedicate. Not to mention lot of memorization as there are so many codes to remember in Braille. Every letter, a number, a punctuation is a group of raised dots. So, you have to remember everything. The older we get, the less we remember.

Besides, Braille requires finger sensitivity. If a person has done hard labor and his fingers are calloused, Braille becomes a bit of a challenge [3]. Hence, as stated earlier a sighted adult who became blind later in his life will be reluctant to learn Braille because he is more familiar with seeing rather than touching.

With the rapid development in technology, various systems have been introduced for blind people to use for reading and writing. Most of these systems were implemented to read text documents. They were not capable of reading tabular data, equations, charts and images. More about these existing systems will be discussed in the literature survey section later on.

When brainstorming ideas for a research, we came across this issue. So, we concentrated on that, thinking what if we could develop a system to read natural images, charts, tables and mathematical equations just like most apps in play store reads text? That was the beginning of this research project. After months of working on that, we finally developed the perfect solution to the above problem of reading.

1.2 Literature Survey

This section is allocated to describe about previous works (existing systems, researches, etc.) that had been done regarding digital talking books. Digital talking books are special kind of electronic books. They are specifically designed for visually impaired people to meet their information access and reading needs. Digital talking books will not work on players that are not designed to play them. Rocket E-Book is a good example for that [4]. They are not compatible with that device. Even so, thanks to the new technologies there are other talking books and reading apps available for the blind people to use. Such as Amazon Kindle which comes with accessibility features, BARD Mobile, Capti Voice, KNBF reader and so on [5],[6]. These applications have screen reading, screen magnification and many useful functionalities that would help visually impaired individuals to read documents. These systems are mostly mobile applications that can be installed in a smart phone. But there are other systems as well like hardware devices which do the same job.

However, they are not perfect systems. There are some major drawbacks in these systems. Mainly, they are designed to read text-based contents. They would at most be suitable to read a novel as novels usually do not have anything but paragraphs. The problem arises when a visually impaired person needs to read a STEM book. Because they have more than just words. They have lot of charts, equations, tables and even graphs. Then he/she cannot use these existing systems to read such books.

Focusing on that issue, there had been a research conducted to develop a mobile application called Schmoozer [7]. The significance of this application is that, it can read graphical contents like images, tables, and equations. It is not capable of reading charts though. It provided solutions to many of the mentioned problems but still has room for improvement.

1.2.1 Region Identification and Reading Text Contents

In Schmoozer, there is a feature to detect graphical contents such as natural images, equations and tables. It detects these contents in a document, extract it and save in a separate folder to read later. It is the first step of the entire process. For this function, image processing methods had been used. When a vision impaired individual uploads an image of a document to the server, the image is cropped and goes through this function. Authors had collected images of mathematical equations, textual contents, tables and natural images and stored them in separate folders to use as a data set for identification purpose. The function can extract features of cropped image using HOG algorithm and suggest the region using SVM. SVM algorithm contains a predefined function called prediction. Authors had collected a set of images of equations, text paragraphs, natural images and tables. This is the dataset they used to correctly identify the region. After detecting the correct type, it will be directed to the relevant folder. However, in our Reading Eye application, we use a completely different method which is faster and more accurate than this one. And these algorithms are kind of outdated as we have modern, state-of-the-art technologies nowadays.

1.2.2 Reading Natural Images and Tabular Data

In Schmoozer, the intention of 'Graphical image identification' was to separately recognize and label the images contained in a document. Authors had implemented a strong dataset that contains a lot of images of humans, animals and many other objects with significant behavioral changes. Then they had applied HOG feature to identify unique features in the image and convert to a decimal value to store in a database. The database was trained using SVM algorithm to convert the dataset into the trained dataset. Then the converted decimal values of HOG features were compared with the trained dataset to find the name of the object. After finding the label that matches the objects, a digitized text will be generated describing the graphical image and then it will be saved in a text file to be used later. Though the objective of image component in both systems is similar, the method of implementation is vastly different. As there will be no use of both HOG and SVM algorithms either to extract features or to convert the dataset into a trained dataset.

Table based content identification function in Schmoozer is to recognize the type of data table (whether it is 2-columns or 3-columns) and convert table data into meaningful digitized text. It has implemented with a strong dataset with a huge number of 2 column and 3 column table images. Identification of the table type is quite important in this function. Then authors have applied HOG feature extraction to identify unique features of 2 column and 3 column tables and converted those features into a decimal value to store in a database. The database was trained using SVM algorithm to convert the dataset into the trained dataset. Data inside the table image was read using an in-built OCR function of MATLAB. Finally, the generated digitized text was written to a text file. Two functions were used to generate digitized text for each type of tables. The main drawback of this feature is, it reads only 2/3 column tables hence, more columns cannot be read using this function. Other than that, authors have used programmatically defined methods to generate text. If we want to use this application for other table types, we would have to collect more images of different table types and re-implement these functions to generate text for each table type.

1.2.3 Reading Equations and Charts

As mentioned earlier, Schmoozer contains a feature to read equations. This function separately recognizes characters and symbols in a mathematical equation and convert them into digitized text. Authors had designed mathematical symbols and numbers using Photoshop in bitmap format. All the major mathematical characters were created and stored inside a folder. The image with the equation was preprocessed and then each single character in the equation converted into a binary value. Finally match every character of the input image with the binary value obtained from a previously created template and regenerate the equation as digitized text. Output of this function is a set of single digitized values for single character/symbol in the equation. Then it is written to a text file. The drawback in this method is to design each and every mathematical symbol there it is. Therefore, to make a good output, a lot of characters and symbols will be needed. That is somehow a tedious and time-consuming task.

Schmoozer does not have a functionality to read charts in a document. And we were unable to find other researches that had conducted entirely for reading charts or contained as a feature just like Schmoozer. However, there is one research which does not read but detect charts in a printed document [7]. It detects the chart region using deep learning methodologies such as Mask RCNN. The researches had been able to detect chart regions and classify them in to classes such as bar charts, pie charts, scatterplot charts, box charts, area charts, etc.

1.2.4 Enhance the Security and Data Encryption

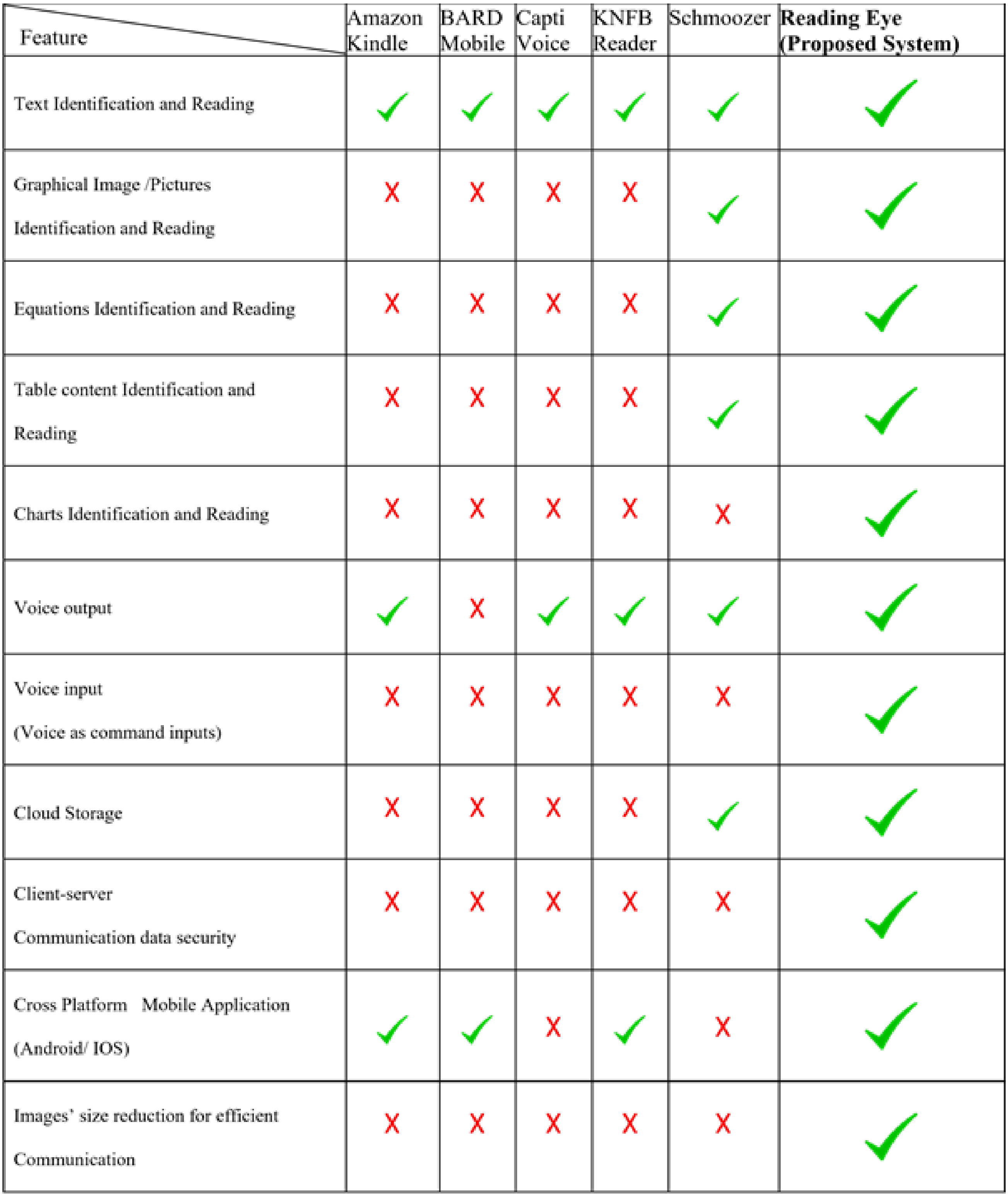
There are many reading applications for visually impaired people in app stores but none of them have any encryption algorithm to secure their communications with servers. Because of that reason, we built that feature in our system.

1.3 Research Gap

There are numerous systems available in the market to assist visually impaired people when reading books and other documents. However, they have some issues with their functionalities, how they operate, and their usability. This section is dedicated to discuss such drawbacks in existing systems that have already developed to assist visually impaired individuals. It will be discussed as a comparison between our system and the existing systems to indicate why the new system was necessary in the first place.

Following an in-depth analysis about existing works, we got a thorough understanding about their strengths and shortcomings. They lack numerous features. Most importantly as mentioned several times in this report, they cannot be used to read graphical contents. So, they are not good with STEM books. Therefore, a visually impaired student would find them hard to use. Naturally, Schmoozer is different from the others we found. Even Schmoozer was developed several years ago. So that, researchers used outdated technology. IT industry moves faster than any other industry in the world, which means new technologies come and existing systems easily become ancient and obsolete. And people invent new frameworks, libraries every single day. So, why not use them to improve these systems? They provide more facilities than older ones anyway. These new technologies are faster and more accurate than older ones. And most importantly, issues with the older ones have been fixed or it is entirely replaced by a new one which is more capable. Our system was built using frameworks like Keras, TensorFlow, OpenCV and many more. They are recent and give extremely good results.

A comparison between Reading Eye and other systems identified in the literature survey is given below.



Feature	Amazon Kindle	BARD Mobile	Capti Voice	KNFB Reader	Schmoozer	**Reading Eye (Proposed System)**
Text Identification and Reading	✓	✓	✓	✓	✓	✓
Graphical Image /Pictures Identification and Reading	✗	✗	✗	✗	✓	✓
Equations Identification and Reading	✗	✗	✗	✗	✓	✓
Table content Identification and Reading	✗	✗	✗	✗	✓	✓
Charts Identification and Reading	✗	✗	✗	✗	✗	✓
Voice output	✓	✗	✓	✓	✓	✓
Voice input (Voice as command inputs)	✗	✗	✗	✗	✗	✓
Cloud Storage	✗	✗	✗	✗	✓	✓
Client-server Communication data security	✗	✗	✗	✗	✗	✓
Cross Platform Mobile Application (Android/ IOS)	✓	✓	✗	✓	✗	✓
Images' size reduction for efficient Communication	✗	✗	✗	✗	✗	✓

Table 1: Comparison of Reading Eye with existing systems.

1.4 Research Problem

Globally, Braille is the primary reading method for blind people to access information and education materials independently. In this system, each character is represented by a combination of one to six raised dots. A dot may be raised at any of the six positions to form 64 possible subsets. Without Braille codes, blind people would have never been able to read or write. Although it is the most effective method so far, there are some problems with the Braille system as well.

Particularly for blind students, some subjects can be unattainable because of textbooks and exams may not be readily available for those courses in a braille friendly way. Subjects like science, engineering, and mathematics, require advanced codes. They are heavily contained with maps, charts, diagrams, figures and equations that have to be redesigned in order for the braille reading students to feel and understand a concept. It will be even harder for students who enrolled in a biology class. As there should be books which include images of human body, molecules, and cells, for students to refer. Because of the complexity of the contents in that kind of subjects, it is hard to create the same book in braille format.

Not only they are unavailable, but also more expensive than most college textbooks. It will cost up to $15,000 to convert just five chapters of a science book making it even harder to publish a science book in braille format because the conversion is in fact difficult [8].

Another issue is, they take up more physical space than normal printed books. A 1000-page math book could easily be 5000 pages in braille format [8]. If a book is this much larger, having thousands of pages, it will be difficult for a student to use that book. Therefore, it lacks the usability. Not to mention these are visually impaired students we are talking about here. It certainly will be tedious for them to use such gigantic books in their day-to-day lives.

Furthermore, braille reading speed is relatively slower than the normal reading speed. Several researches had been done regarding braille reading speed. They proved Braille method is much slower. According to these facts and findings, it is obvious that there are some major issues with braille method. As it was designed and introduced centuries ago, it may not be compatible with the needs and desires of a person in a modern world. On that account, we strongly believed these issues could be addressed with the help of modern tools and technologies. Speaking of that, currently there are both software solutions and hardware solutions in the market which have developed to assist visually impaired individuals for reading such as DAISY books.

DAISY stands for Digital Accessible Information System [9]. Other than that, there are some hardware devices too. Usually these devices are found in developed countries and they are expensive. Hence, people who cannot afford them cannot use. Especially people in developing countries like Sri Lanka.

As a solution to all these problems, we carried out this research to implement a mobile application, which is easy to use, readily available and most importantly free of charge.

1.5 Research Objectives

The main goal of this research project is to develop, a fully fledge document reading system to read text, images, charts, tables and equations in any kind of printed document accurately making it easier for a blind person to read documents without any hassle. Moreover, Reading Eye will facilitate visually impaired people to read printed documents that are not written using the braille system as well.

The sub-objectives of the project are:

- Reading and detecting text-based contents and region identification of the document images.
- Enhance the security of the collected data, application, and client-server communication and reduce the size of images for highly efficient communication.
- Implement an encryption algorithm unique to the application for securing the communication between the cloud and the application.
- Enforce cloud policies to safeguard and handle the process according to the standards and best practices.
- Reading and detecting graphical Images-based contents and generate descriptions.
- Reading tabular data in a document image and generate descriptions.
- Reading and detecting charts based-contents and generate descriptions.
- Reading and detecting equations-based contents and generate descriptions.
- Enable User-friendly interface & Quick, simple and easy-to-use.
- Provide accessibility features for the intended user base.
- Provide an enjoyable reading experience for the user who is using the application.

CHAPTER 2

Methodology

This section describes about the development process of the entire system in general. Moreover, implementation and testing details of the natural image and table reading component will be discussed in further.

2.1 System Overview

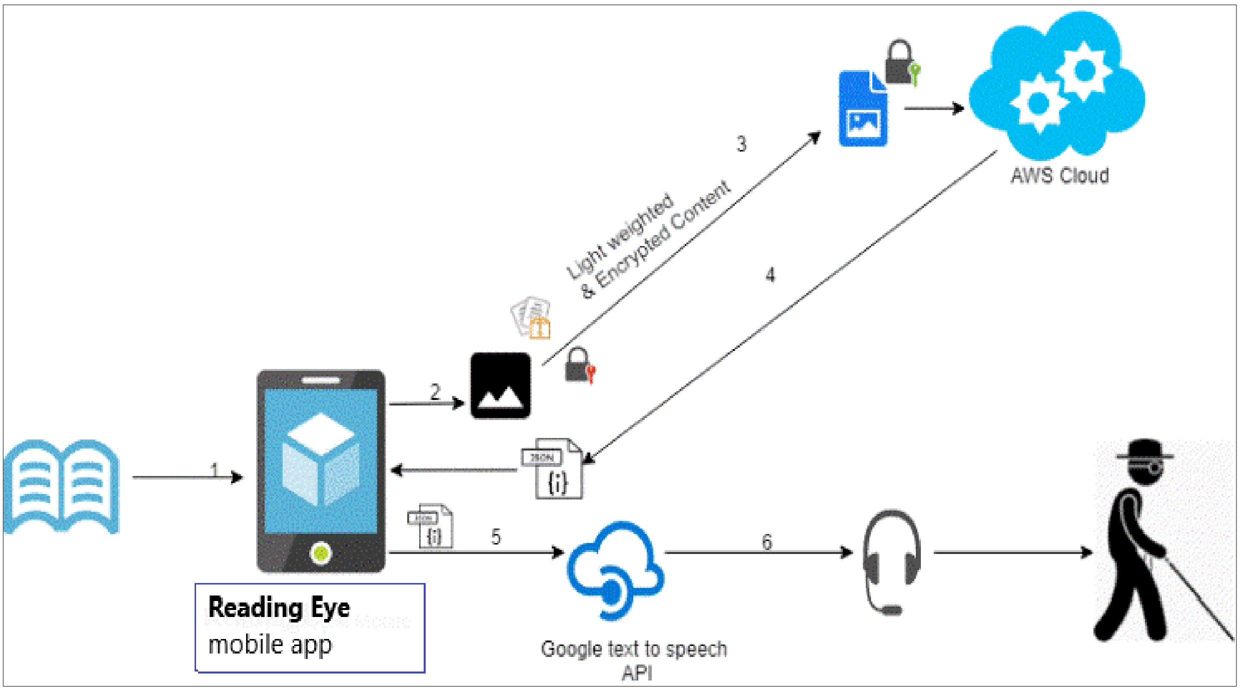


Figure 1: High level architecture diagram.

System overview brings up the most appropriate tools, technologies and software solutions for the implementation phase and flow of the proposed solution. We developed a mobile application that is capable of understanding the contents of document images and reading out aloud for a blind user to understand its content. The application can read text, images, tables, equations and charts. Moreover, security of the application will be enhanced.

When using the application, a user can point the camera of his device to the printed document which he wants to read. Then the camera will capture the photo and the app will set it to correct orientation. If the photo capturing went wrong, the user will be notified by the application (via an audio message). After that the captured image file will be compressed and encrypted using a specific algorithm to ensure the security of the file before uploading it to the cloud server. We are planning to host the application in a cloud server (MS Azure) to do the back-end processing because a mobile phone or tablet do not have such processing capabilities. Thus, in the cloud server all the processing will be handled and only the generated audio file will be sent to the mobile application. Then the mobile application will play the audio file for the user to listen.

Used technologies as follows:

- Python as the programming language
- Keras, TensorFlow – to create deep learning models
- OpenCV library for image processing tasks
- Tesseract OCR
- Android platform to develop the mobile app.
- Postman
- Spyder
- Anaconda Distribution
- MS Azure service for cloud hosting
- Text-to-speech API

2.2 Image Reading Component

Image reading feature is used to generate a textual description about a natural image in a document the user is reading. The feature is expected to identify humans, different sorts of animals and objects and so on. When implementing this feature ImageNet weights were used to correctly identify those said objects in an image [12]. It is an image database which contains thousands of images belonging to various categories. It contains about 1000 classes. Having so many classes in a single dataset is the reason for using ImageNet in this function. So that, it can identify more objects in the image accurately. The objective of this feature is to output a proper description about the image itself so that the visually impaired user can be aware of the contents of that image. After going through a big process, this feature is able to just do that.

As per technologies, both computer vision and natural language processing were used to read the image content and to create a language model to turn the understanding of the image into words in order to generate a textual description. The core of this module is training a deep learning model which is capable of predicting a suitable description for a natural image automatically. Therefore, the most important part is the model. This model consists of two sequential models. First model is image model and the second one is language model. Both do two separate tasks. But before training the model, there are some steps that need to be mentioned here. First one is preprocessing. It is the initial step of the entire training process.

In this feature, we used images and captions that describes the images as input. Prior to training, it is vital to preprocess these input data so that unnecessary elements could be removed and obtain an accurate final model. However, the processing step is different for both input types. For images, it is done not to remove unnecessary elements but to collect features in each image and combine them as a separate dataset. This step kind of speeds up the training.

- **Preprocessing of Images**

In this phase, we use a pre-trained feature extractor. Most commonly used feature extractors InceptionV3, VGG16, VGG19, Mobile Net, Xception and so on. They each give various results. After trying few of them, we finally used InceptionV3 as it gave more accurate results and because it is a lighter model. This pre-trained model uses the ImageNet weights to extract features in an image. It is a convolutional neural network which consists of many layers.

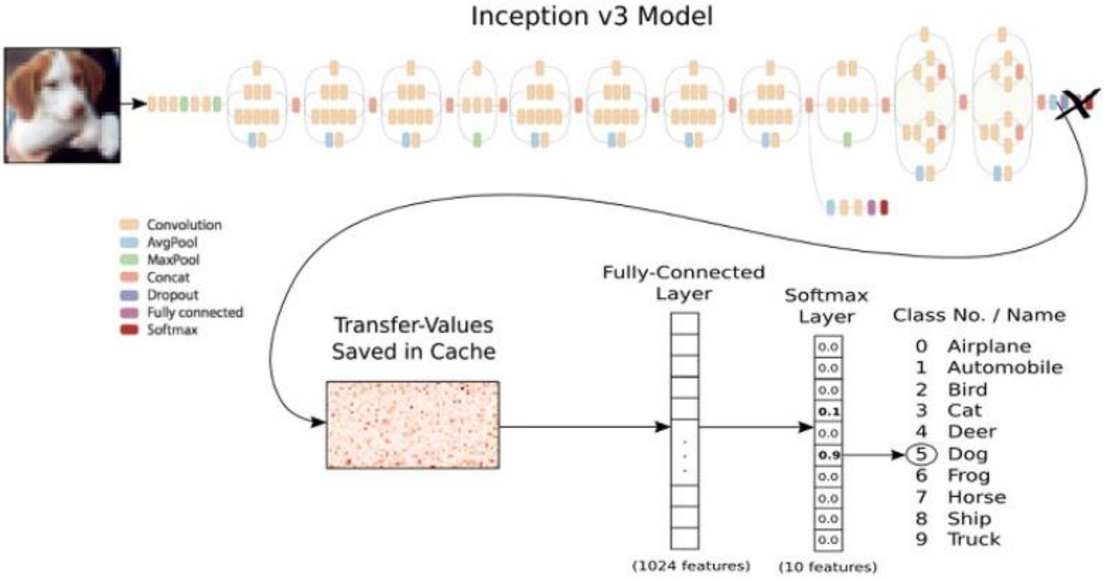


Figure 2: Classifying an object (dog) using InceptionV3 model. [13]

Here is an example to show how to use InceptionV3 model to classify an object to its related class (in this case dog class). However, in our case we do not want to do the classification. We use it to extract features in the images (e.g. animals, people) and use them as input to train our model. InceptionV3 contains many layers. Only the last layer gives us the output (classification). Since we do not want classification, we can remove the last layer of InceptionV3 and compile again, so that it outputs extracted features that we need. Output of newly compiled model is 1 X 1 X 2048 tensor vector. For the image reading component, we used Flickr8K dataset which consists of 6000 images and 30000 captions, each image having 5 captions. These captions describe the image. However, all the images in the dataset are processed using InceptionV3 and extracted features will be combined as one input array. Then it will be used as the input data in the image model we mentioned earlier.

- **Preprocessing of Image Descriptions**

Just like the images, preprocessing of image descriptions also a must to make the final outcome more accurate. In Flickr8k dataset, one image has precisely 5 descriptions (single line though). These descriptions usually contain single letter words, punctuations, and even numbers. Therefore, we need to remove these unnecessary stuffs and clean the text. And we convert all the words to lowercase as well. After cleaning text, we can create a vocabulary out of all the words. Cleaning text helps to create a smaller vocabulary because of unnecessary words have already removed, which result in a smaller model that will train faster. Finally, we save dictionary of image identifiers and descriptions to a new file.

- **Building the model**

After the preprocessing, the next step is building the model. This model is the core of the entire program because it is this model that we use to predict descriptions about images later. This model consists of three parts.

1. **Photo Feature Extractor**
   This is a 48-layer InceptionV3 model trained on the ImageNet dataset. We have already extracted photo features using this pretrained model, so that we can use that features as input for this model. Feature vector expects input photo features to be a vector of 2048 elements. These input vectors are then processed by a Dense layer to create 128 element representation of the photo.

2. **Sequence Processor**

   This part consists of a word embedding layer for handling the text input and a Long Short-Term Memory (LSTM) recurrent neural network layer. RNNs were capable of connecting the previous information with the current task. But it had some shortcomings remembering all the relevant data and then connect that data with the current task. However, as a solution LSTMs came. LSTMs are a special kind of RNNs. These layers can remember information for a long period of time. It is extremely useful when generating descriptions. Because when generating a description, the machine has to find the most suitable word to generate after the previous word. Otherwise the sentence would not make any sense. LSTMs are extremely good at this task and that is reason for using an LSTM in our model too. This part expects input sequences with a pre-defined length. We used 34 words for that. Then the input sequences are fed into an Embedding layer which is followed by an LSTM layer with 128 memory units.

3. **Decoder**

   Both feature extractor (image model) and sequence processor (language model) are implemented as a Sequential model which output a 128 element vectors which will be used as the input in the decoder part to produce descriptions. When building the model, we did some experiments by changing certain layers at certain points to see how it will affect the output. After the merging of above two models, we used two Dense layers instead of LSTMs. However, the accuracy was not satisfying. Relatively LSTMs are much better as they do a better job at predicting the next word in a sentence. Therefore, when using two LSTMs, we got more accurate descriptions for the sampling image. That is the reason for using two LSTMs in the latter part of the model.

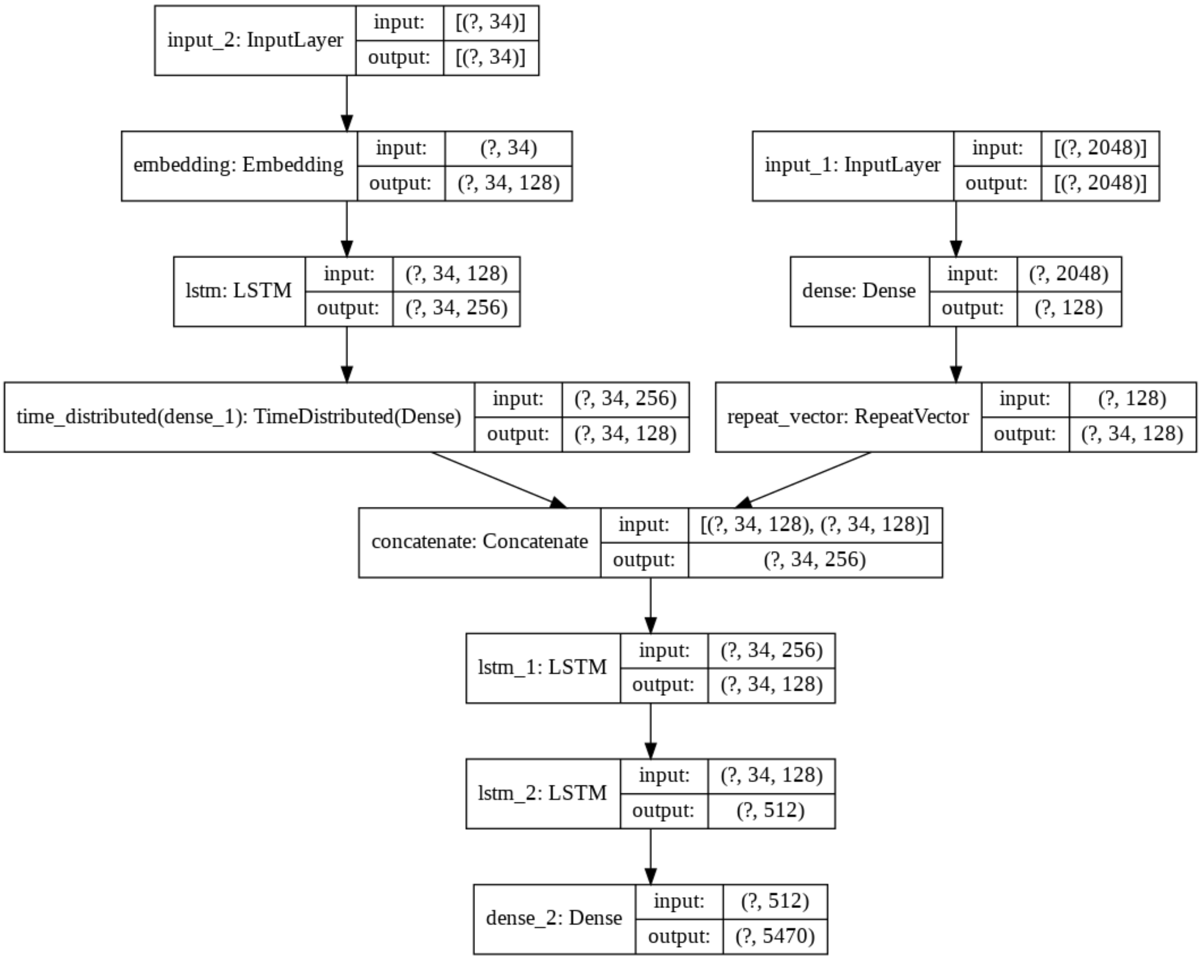
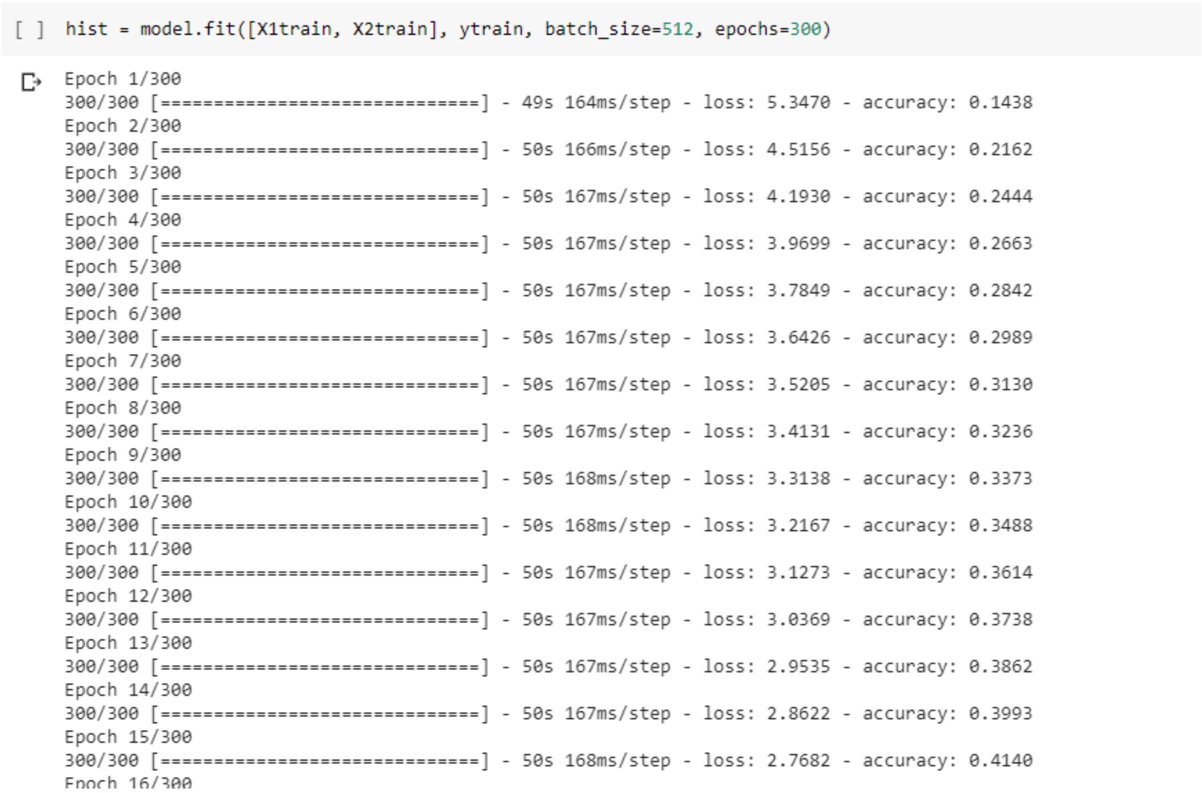


Figure 3: The structure of the trained model.

- **Fitting of the model**

As the final step, we need to compile and train the model with the dataset. The final model was trained using 300 epochs to increase accuracy and get a good final model. The more we train, the better results we get. However, because of some limitations with hardware, the model was trained using only about 2000 images. Because we needed a large amount of memory (RAM) to hold all the data, which we did not have. Therefore, only a subset of the entire dataset was used.



```
[ ]  hist = model.fit([X1train, X2train], ytrain, batch_size=512, epochs=300)

     Epoch 1/300
     300/300 [==============================] - 49s 164ms/step - loss: 5.3470 - accuracy: 0.1438
     Epoch 2/300
     300/300 [==============================] - 50s 166ms/step - loss: 4.5156 - accuracy: 0.2162
     Epoch 3/300
     300/300 [==============================] - 50s 167ms/step - loss: 4.1930 - accuracy: 0.2444
     Epoch 4/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.9699 - accuracy: 0.2663
     Epoch 5/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.7849 - accuracy: 0.2842
     Epoch 6/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.6426 - accuracy: 0.2989
     Epoch 7/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.5205 - accuracy: 0.3130
     Epoch 8/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.4131 - accuracy: 0.3236
     Epoch 9/300
     300/300 [==============================] - 50s 168ms/step - loss: 3.3138 - accuracy: 0.3373
     Epoch 10/300
     300/300 [==============================] - 50s 168ms/step - loss: 3.2167 - accuracy: 0.3488
     Epoch 11/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.1273 - accuracy: 0.3614
     Epoch 12/300
     300/300 [==============================] - 50s 167ms/step - loss: 3.0369 - accuracy: 0.3738
     Epoch 13/300
     300/300 [==============================] - 50s 167ms/step - loss: 2.9535 - accuracy: 0.3862
     Epoch 14/300
     300/300 [==============================] - 50s 167ms/step - loss: 2.8622 - accuracy: 0.3993
     Epoch 15/300
     300/300 [==============================] - 50s 168ms/step - loss: 2.7682 - accuracy: 0.4140
     Epoch 16/300
```

Figure 4: Training of the model.

- **Generating new descriptions**

Though we have a trained model now, we cannot deploy it in a server with the same code to generate new descriptions. When using in a production environment, there is another mechanism. First, we load the model into our application. Then we should upload an image (search the directory for images in our case) to use with the image. Next, the image will be processed by a pretrained model and all the descriptions (saved earlier) will be loaded to the program. Finally, all inputs will be fed to our model and it will generate a suitable description for the given image.

2.3 Table Reading Component

Table data reading feature is used to read contents of the table images. This feature is able to read tables with any number of columns without adhering to 2 or 3 column types. The only constraint is that the table should have horizontal and vertical lines. The technique used here relies on identifying the table structure, so that it must have both vertical lines and horizontal lines. Hence, it does not work on tables with no lines.

Since this component also is required reading text from an image, we used the Tesseract OCR tool to extract characters from the image. When working with OCR, as the first step we need to do the preprocessing. Initially before reading the text, the image must read, converted into grayscale, do thresholding and inverting. The purpose of gray scaling is to provide less information for each pixel mostly because grayscale images are enough for many tasks so there is no need to use more complicated and harder-to-process color images. Automatic thresholding is used to extract useful information encoded into pixels while minimizing background noise.

Detection of boxes in the image is the next step. For that task Morphological operations (erode, reconstruct, dilate are examples for morphological operations) are used. By using OpenCV library morphological operations can be done on the image. Here we define two rectangular kernels with the length based on the width of the image. First kernel is to detect horizontal lines and second kernel to detect vertical lines. After defining kernels, we do morphological operations (erode operation) to detect the vertical and horizontal lines in the table. After detecting lines, they should be saved as separate images like below.

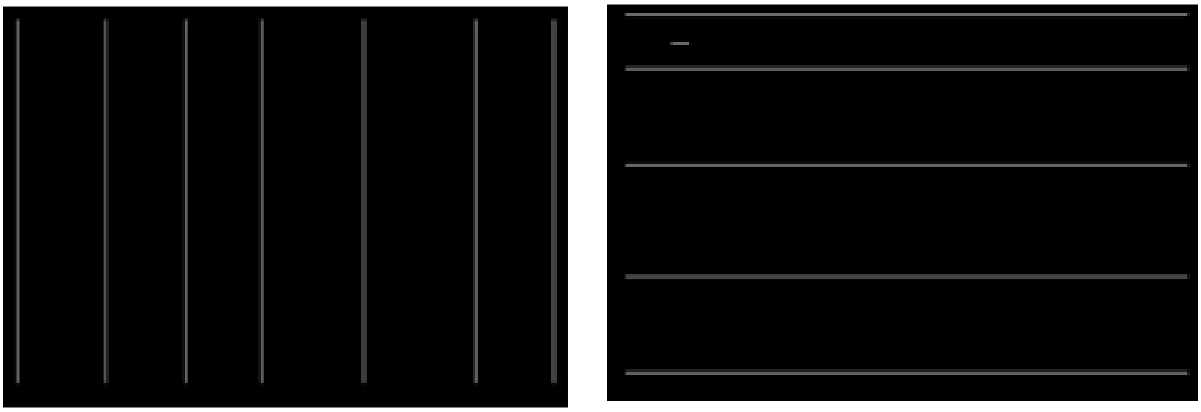


Figure 5: Extraction of vertical and horizontal lines of tables.

As the next step both horizontal and vertical images should be added together and create the final image with entire table structure without unnecessary lines or text. The final image will only contain boxes excluding text in the original table image. This way we can eliminate noise from the image and identify boxes with information precisely.

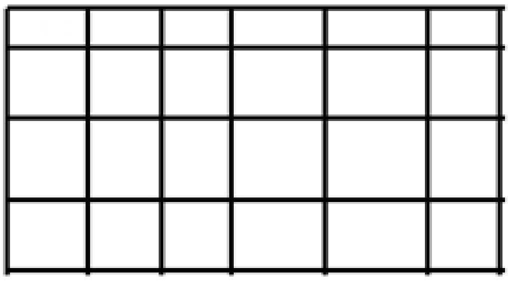


Figure 6: Final structure of the table.

After getting the structure of the table we must find contours in the image. For that we use findContours() method in OpenCV. It finds all the boxes in the image. Then we can sort the boxes from top to bottom. Now loop over all the contours, find the location of all the boxes and crop the part which has a rectangle and save it into the folder. By cropping the box, we can ensure the text inside the box will be read accurately and in order because when reading the table directly by OCR engine it reads the words horizontally regardless of word order (especially in columns). If we directly read images using OCR, there is no way to differentiate columns from rows as all information are read as text line by line. By using this method, not only we can separate columns and rows we can use them to generate meaningful descriptions at the same time. Then the column cells (cropped cells) and row cells should be read iteratively using OpenCV and read the text in the cells using Tesseract OCR engine.

As the final step, we can generate a meaningful textual description using columns values and row values of the table and generate an audio file using Text-to-Speech API for the generated textual description for the user to listen.

2.4 Testing

This section describes the testing process which was performed in order to evaluate the performance of each functionality.

- **Image Reading Component**

To evaluate the skill of this feature we used BLEU score. We evaluated the trained model against a given dataset of photo descriptions and photo features. The actual and predicted descriptions are collected and evaluated collectively using the corpus BLEU score that summarizes how close the generated text is to the expected text.

No. of train images in the dataset: 6000 (only 2000 were actually used to train)

No. of test images: 1000

Epochs: 300 Batch Size: 512 Optimizer: Adam

BLEU Scores on Validation data

BLEU-1: 0.596655	BLEU-3: 0.229676
BLEU-2: 0.342127	BLEU-4: 0.108707

Loss: 0.6002

Accuracy of the model: 0.8495

These are the evaluation results that were obtained for the model after comparing each generated description against all the reference descriptions for the photograph. The higher the BLEU score the better. And it is much better if we could lower the loss value to have a better accuracy. Prediction is more accurate when the loss value is closer to zero. To gain a better accuracy for prediction, it is necessary to train the model using more epochs with a large dataset. To train bigger datasets, it is necessary to have GPUs with RAM at least around 64GB which we lacked while we were doing the research. However, this is the maximum results that we could obtained with the resources we had.

- **Table Reading Component**

The goal of table data reading function is to read a table using OCR and morphological operations. As the result, function generates a description using the read data of the table. To evaluate how good this function is, 35 images were tested using the table reading function in this application and the results are given below. Only tables with border lines were tested as we already stated in the methodology this function does not work well with tables without borders.

Table Type	*Number of Inputs*	*Number of Correct Outputs*
2 column tables with lines	10	10
3 column tables with lines	10	10
4 or more column tables with lines	10	13

Table 2: Accuracy of Table Reading Component.

By using these table, we could finally measure the accuracy of the function.

Average Accuracy (AC) = Number of correct inputs / Number of total inputs * 100%

= 33 / 35 * 100%

= 94.3%

Above calculation shows that the table reading function achieves an accuracy around 94% which is a good accuracy. Hence, we can assume this function provides best autonomous, accurate document reading service to visually impaired individuals.

Moreover, output of each functionality is given in Chapter 3 for further understanding.

2.5 Commercialization of the Product

Systems to read graphical contents such as images, charts, tables and equations almost do not exist in the current market. Because of that, visually impaired people find it very difficult read documents with such contents. But using this application, they can easily read documents with these graphical contents.

Target customer base,

- Visually impaired people
- Students
- People interested in reading audio books
- Researchers

There are so many benefits that users can get through our application. Some of them are,

- User can use the app anywhere with an uninterrupted internet connection.
- No gender limitation.
- No age limitation.

CHAPTER 3

Results and Discussions

3.1 Results

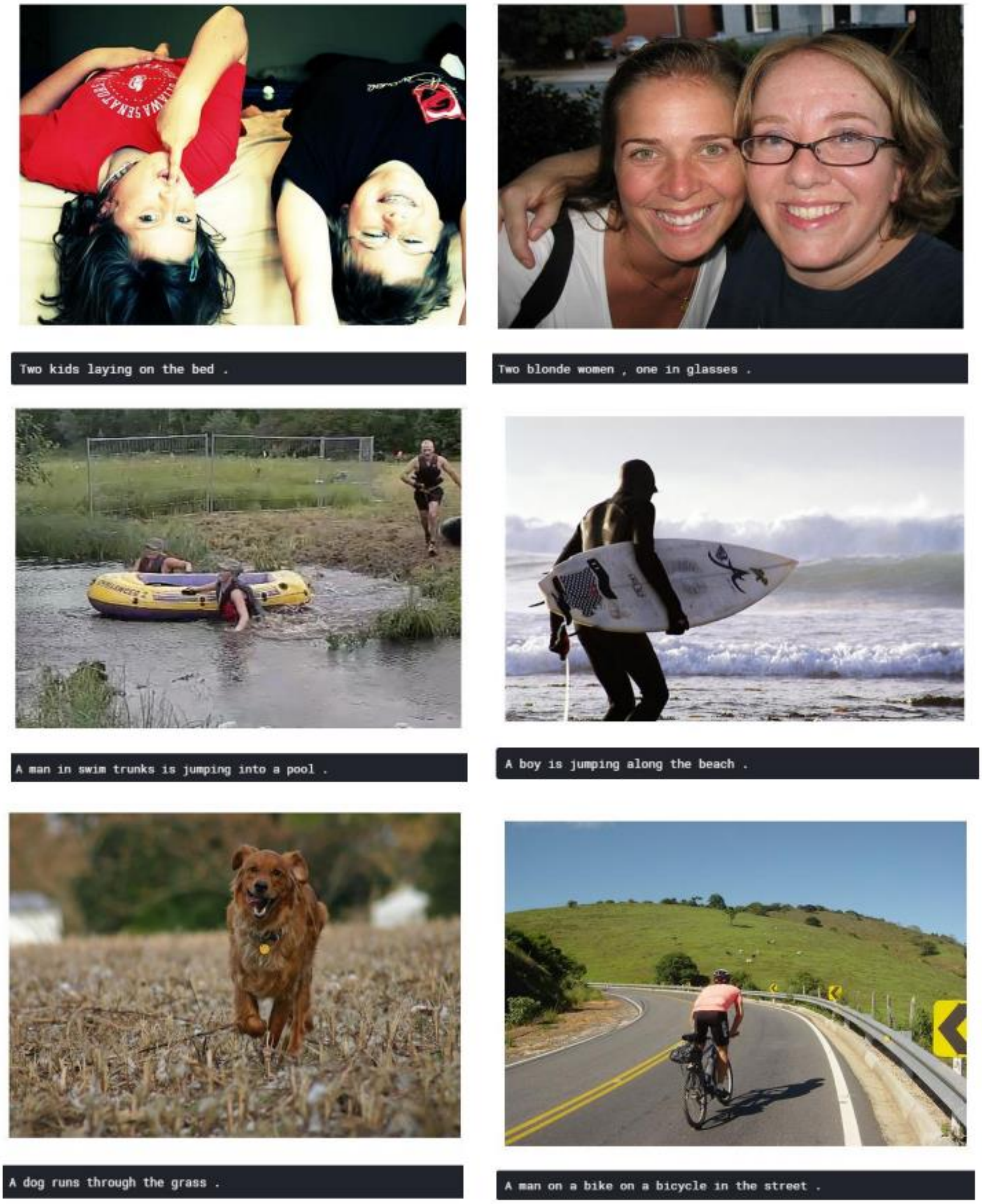
In this section, we discuss about all the results obtained from the implemented image and table reading features of the application and accuracy of them. Though we get a final audio file as the outcome of the entire project, we would like to show output result of each functionality. Hence, the results are as follows,

- **Image Reading Component**

As mentioned in the methodology section, a deep learning model was developed and used to predict a suitable description for a sample image in the image reading component. It gives good and reasonable results for the natural images. Here, we have to keep in mind that, this functionality is entirely artificial intelligence which means, we do not explicitly instruct the machine to what it should gives as the output for a given image. Machine learns by itself without human intervention. Therefore, it could give some weird results sometimes. But most of the times, it gives good results. It identifies objects accurately most of the times, but somehow fails to understand what is happening in the image. Naturally, these are some drawbacks of the machine learning part.

However, these are the results we got for image component.

- New descriptions for images in the Flickr8K dataset.



Two kids laying on the bed .

Two blonde women , one in glasses .

A man in swim trunks is jumping into a pool .

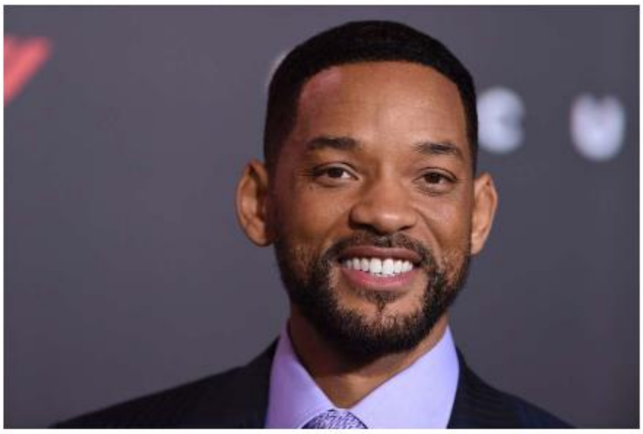
A boy is jumping along the beach .

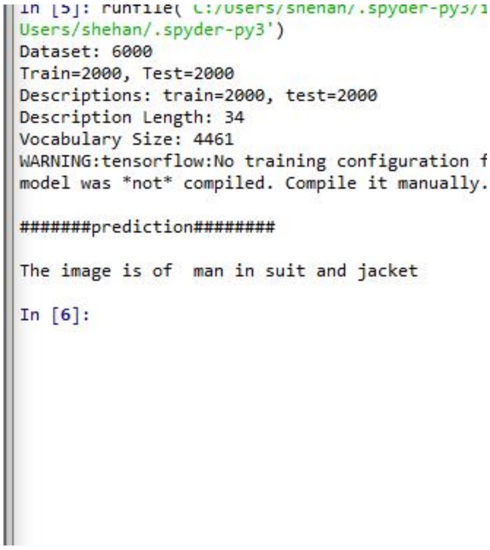
A dog runs through the grass .

A man on a bike on a bicycle in the street .

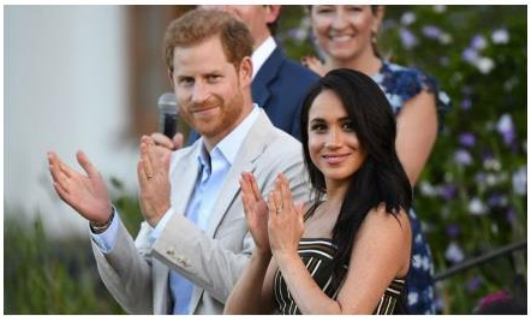
Figure 7: Set of tested images with generated descriptions.

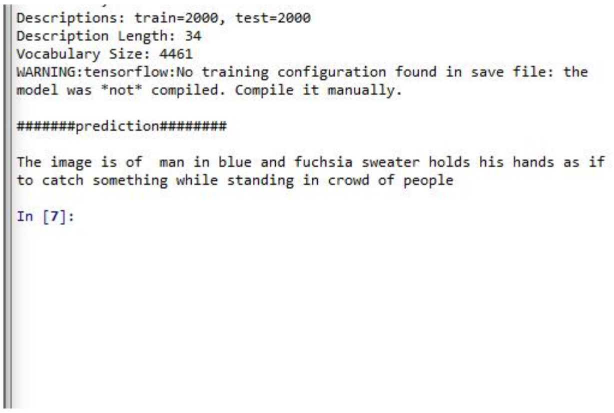
- Descriptions for new images



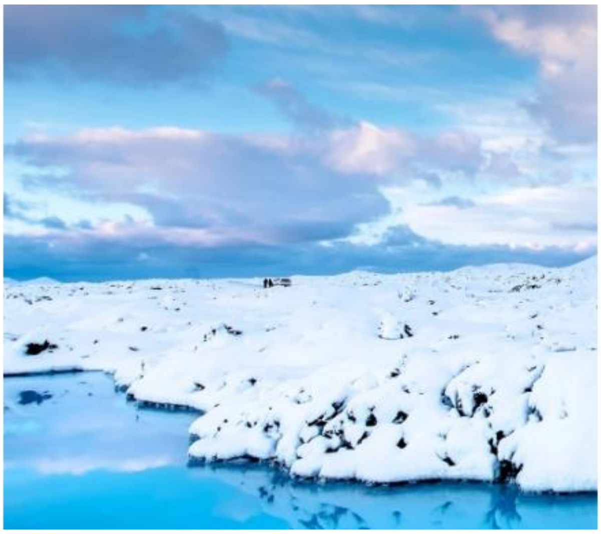


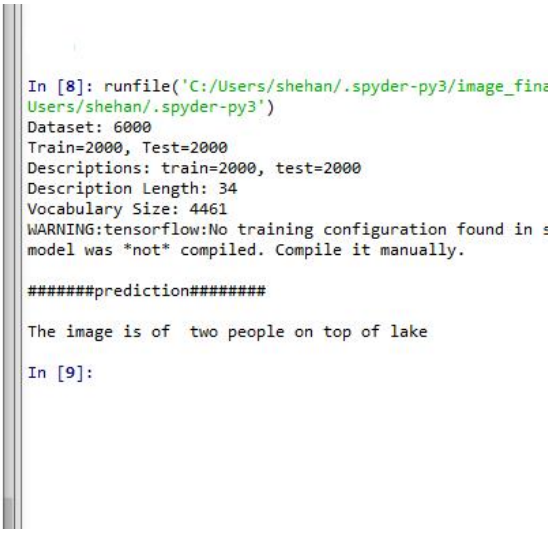
```
In [5]: runfile('C:/Users/shehan/.spyder-py3/i
Users/shehan/.spyder-py3')
Dataset: 6000
Train=2000, Test=2000
Descriptions: train=2000, test=2000
Description Length: 34
Vocabulary Size: 4461
WARNING:tensorflow:No training configuration f
model was *not* compiled. Compile it manually.

#######prediction########

The image is of  man in suit and jacket

In [6]:
```





```
Descriptions: train=2000, test=2000
Description Length: 34
Vocabulary Size: 4461
WARNING:tensorflow:No training configuration found in save file: the
model was *not* compiled. Compile it manually.

#######prediction########

The image is of  man in blue and fuchsia sweater holds his hands as if
to catch something while standing in crowd of people

In [7]:
```





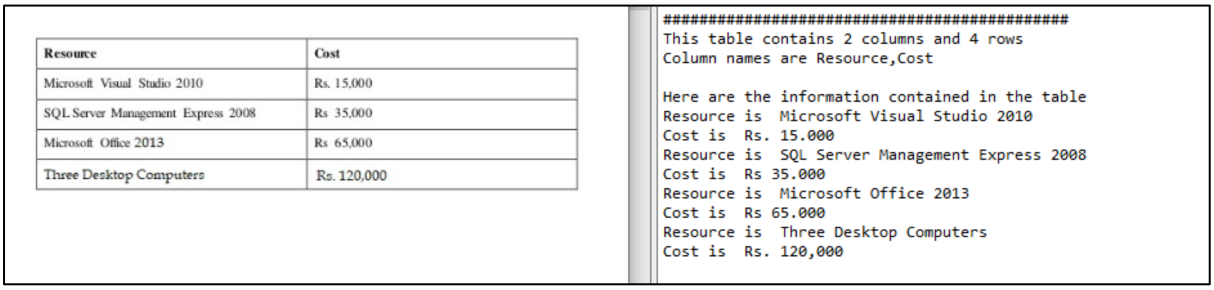
```
In [8]: runfile('C:/Users/shehan/.spyder-py3/image_fina
Users/shehan/.spyder-py3')
Dataset: 6000
Train=2000, Test=2000
Descriptions: train=2000, test=2000
Description Length: 34
Vocabulary Size: 4461
WARNING:tensorflow:No training configuration found in s
model was *not* compiled. Compile it manually.

#######prediction########

The image is of  two people on top of lake

In [9]:
```

Figure 8: Set of new images with generated descriptions.

- **Table Reading Component**

These are the results that got from the table reading component. This function can read tables with any number of columns. Examples are given below with the generated results.

2 – Column Table

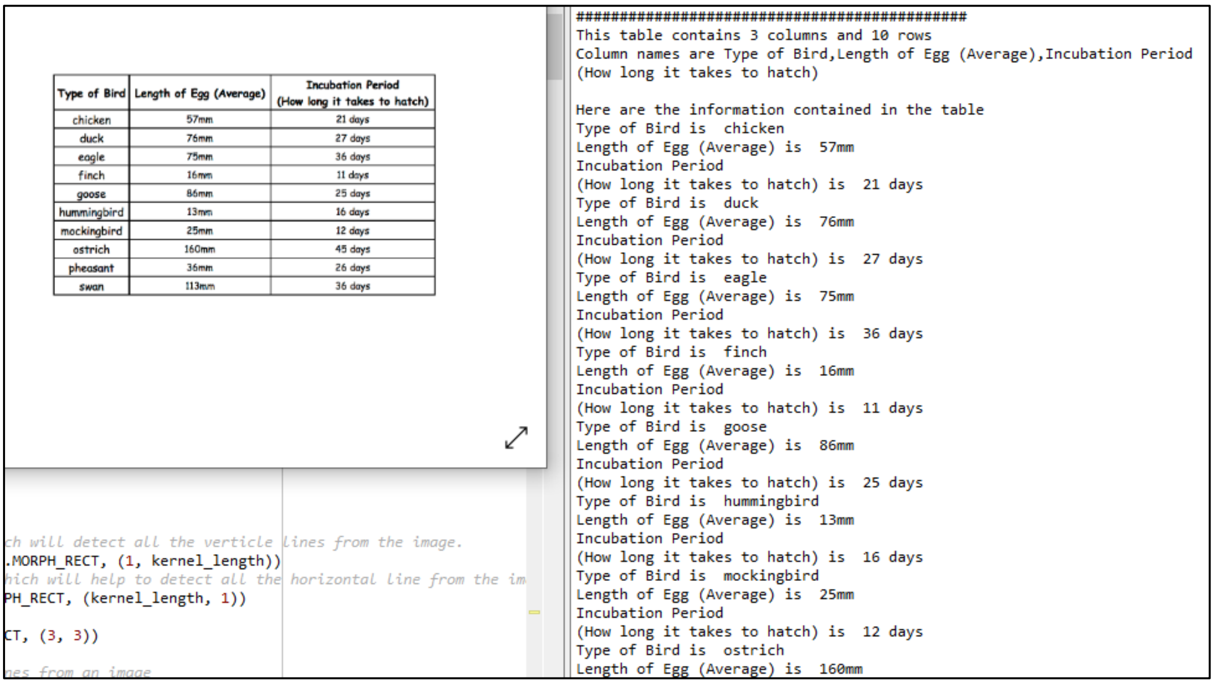


Resource	Cost
Microsoft Visual Studio 2010	Rs. 15,000
SQL Server Management Express 2008	Rs 35,000
Microsoft Office 2013	Rs 65,000
Three Desktop Computers	Rs. 120,000

```
#############################################
This table contains 2 columns and 4 rows
Column names are Resource,Cost

Here are the information contained in the table
Resource is  Microsoft Visual Studio 2010
Cost is  Rs. 15.000
Resource is  SQL Server Management Express 2008
Cost is  Rs 35.000
Resource is  Microsoft Office 2013
Cost is  Rs 65.000
Resource is  Three Desktop Computers
Cost is  Rs. 120,000
```

Figure 9: Generated description of a 2-column table.

3 – Column Table



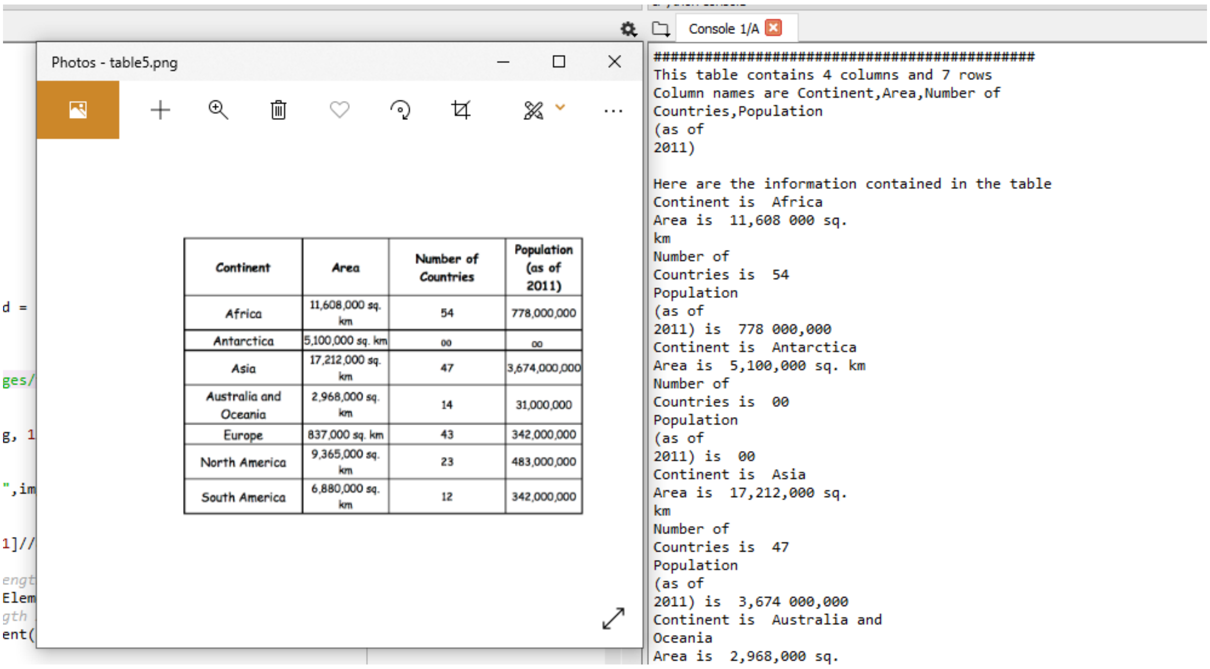
Type of Bird	Length of Egg (Average)	Incubation Period (How long it takes to hatch)
chicken	57mm	21 days
duck	76mm	27 days
eagle	75mm	36 days
finch	16mm	11 days
goose	86mm	25 days
hummingbird	13mm	16 days
mockingbird	25mm	12 days
ostrich	160mm	45 days
pheasant	36mm	26 days
swan	113mm	36 days

```
ch will detect all the verticle lines from the image.
.MORPH_RECT, (1, kernel_length))
hich will help to detect all the horizontal line from the im
PH_RECT, (kernel_length, 1))

CT, (3, 3))

nes from an image
```

```
#############################################
This table contains 3 columns and 10 rows
Column names are Type of Bird,Length of Egg (Average),Incubation Period
(How long it takes to hatch)

Here are the information contained in the table
Type of Bird is  chicken
Length of Egg (Average) is  57mm
Incubation Period
(How long it takes to hatch) is  21 days
Type of Bird is  duck
Length of Egg (Average) is  76mm
Incubation Period
(How long it takes to hatch) is  27 days
Type of Bird is  eagle
Length of Egg (Average) is  75mm
Incubation Period
(How long it takes to hatch) is  36 days
Type of Bird is  finch
Length of Egg (Average) is  16mm
Incubation Period
(How long it takes to hatch) is  11 days
Type of Bird is  goose
Length of Egg (Average) is  86mm
Incubation Period
(How long it takes to hatch) is  25 days
Type of Bird is  hummingbird
Length of Egg (Average) is  13mm
Incubation Period
(How long it takes to hatch) is  16 days
Type of Bird is  mockingbird
Length of Egg (Average) is  25mm
Incubation Period
(How long it takes to hatch) is  12 days
Type of Bird is  ostrich
Length of Egg (Average) is  160mm
```

Figure 10: Generated description of a 3-column table.

4 – Column Table



Continent	Area	Number of Countries	Population (as of 2011)
Africa	11,608,000 sq. km	54	778,000,000
Antarctica	5,100,000 sq. km	00	00
Asia	17,212,000 sq. km	47	3,674,000,000
Australia and Oceania	2,968,000 sq. km	14	31,000,000
Europe	837,000 sq. km	43	342,000,000
North America	9,365,000 sq. km	23	483,000,000
South America	6,880,000 sq. km	12	342,000,000

```
#############################################
This table contains 4 columns and 7 rows
Column names are Continent,Area,Number of
Countries,Population
(as of
2011)

Here are the information contained in the table
Continent is  Africa
Area is  11,608 000 sq.
km
Number of
Countries is  54
Population
(as of
2011) is  778 000,000
Continent is  Antarctica
Area is  5,100,000 sq. km
Number of
Countries is  00
Population
(as of
2011) is  00
Continent is  Asia
Area is  17,212,000 sq.
km
Number of
Countries is  47
Population
(as of
2011) is  3,674 000,000
Continent is  Australia and
Oceania
Area is  2,968,000 sq.
```

Figure 11: Generated description of a 4-column table.

Above mentioned results are some the tested table images. Other column types also can be used as well.

3.2 Research Findings

We developed the application to make it useful to the visually impaired people specifically students who are studying scientific subjects and other people who are interested in listening to audio books as a hobby. But the main focus was on visually impaired people. For others, it is a side benefit. Almost all the systems currently available in the market cannot read graphical contents. Most of them read only text documents. Nothing to read charts or images. According to the literature survey researchers were unable to find any similar mobile application which can read all the mentioned contents in a document. This is one of the most significant findings of the research project.

When developing the system, it was very difficult for us to find a suitable dataset for image reading tasks. All the available datasets were not fully descriptive of the image and, improvements of dataset are still necessary to describe the images in more detail.

And the training was harder to do in the local pc. Instead, we had to use online platforms like Google Colab and Kaggle for the training purposes as they provided GPUs. And during the research, we realized more training was necessary in order to increase the accuracy of image reading component.

Even with all the limitations, we got pretty decent results for these functionalities.

3.3 Discussion

We strongly believe our Reading Eye application will be a valuable system to visually impaired people. Specially, this will be hugely beneficial to students who are studying science and mathematics. This is an open-source and real-time application. So, users can easily access and get the benefits from this application. We developed it successfully and it will be really helpful for visually impaired people in near future. They can use it for free without spending any money.

For the time being we were able to develop only the Android app. But in the future, we intend to develop the iOS application as well with the hope of attracting more users.

3.4 Conclusion

Considering the digital talking books and other document reading systems available in the market, most of them are focused only on text reading functionality and do not read other types of graphical contents such as images, tables, charts and equations. Also, other than digital talking books, other systems are not implemented solely for visually impaired people. Considering the need for a device with multiple reading capabilities which is specifically made for visually impaired people, that is more comfortable, low cost, portable and efficient the study focused to cover them all with a document reading system which can read not only text but also other graphical contents in a document. As for image reading part, usage of deep learning was immensely effective to make reading more accurate and faster. Because of that, we gave the system the ability to think itself and generate an output. Having quality images improves the accuracy of the table reading function. Moreover, the application process consists of region detection and separation, secure encryption and decryption of the original document images, region analysis and content reading, conversion to textual description of the read content and lastly conversion the description to the audio narration. As said earlier, the purpose of the system development was to help visually impaired people to read documents easily and most importantly without anyone else’s support. As a further benefit, it limits the need to use Braille formatted books as it can read many types of graphical contents in a document. With all these features, Reading Eye will be a better solution for visually impaired people to read the document contents and fulfill their knowledge in a better way.

3.5 Future Works

As for future works, we would like to propose some suggestions that will carry this research forward. First of all, we would like to mention that we came across some issues and limitations when developing the application, specifically image and table reading component. In image component, we were unable to find a suitable public dataset that describes an image in more details. For example, a dataset that each image consists of lengthy paragraphs instead of one-line sentences, because all the public datasets available for this task had only small descriptions about each image. Therefore, it would be really useful to create a dataset with lengthy paragraphs for each image. Other than that, it would be good to train the model with all the images and descriptions to get a more accurate model (with the necessary hardware). And using a larger dataset to train the model will improve the accuracy of the model, generating more accurate and sophisticated descriptions about images. As for table component, we suggest to implement the feature with capabilities of reading tables that are having no horizontal and vertical border lines.

Moreover, it would be good to implement the chart function with the ability to read more chart types like line charts, area charts, scatter plots, etc. And include a new feature to describe graphs which are more common science and engineering books. When adding these new features, region identification should automatically expand to accommodate these changes by identifying new graphical contents. Furthermore, we would like to suggest to expand this research to include more complex equations for read. These suggestions will definitely improve this research further more.

REFERENCES

[1]"Braille", En.wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/Braille. [Accessed: 07- Aug- 2019].

[2]"NLS Factsheet: About Braille - National Library Service for the Blind and Print Disabled (NLS) | Library of Congress", National Library Service for the Blind and Print Disabled (NLS) | Library of Congress. [Online]. Available: https://www.loc.gov/nls/resources/blindness-and-vision-impairment/braille-information/about-braille/. [Accessed: 09- Aug- 2019].

[3] Quora.com. [Online]. Available: https://www.quora.com/How-long-does-it-take-to-learn-Braille-and-does-it-get-harder-as-you-get-older. [Accessed: 10- Aug- 2019].

[4]"A New Look for the Book: Overview of Digital Talking Book Technology | AccessWorld |American Foundation for the Blind", Afb.org. [Online]. Available: https://www.afb.org/aw/2/3/15033. [Accessed: 07- Aug- 2019].

[5] "Accessible Mobile Apps | American Foundation for the Blind", Afb.org, 2019. [Online]. Available: https://www.afb.org/blindness-and-low-vision/using-technology/ [Accessed: 10- Aug- 2019].

[6] Play.google.com. [Online]. Available: https://play.google.com/store.

[7] J. Amara, P. Kaur, M. Owonibi and B. Bouaziz, "Convolutional Neural Network Based Chart ImageClassification.."Pdfs.semanticscholar.org, 2019. [Accessed: 23- Aug- 2019].

[8]"Braille versions of textbooks help blind college students succeed -Marketplace", Marketplace, 2019. [Online]. Available: https://www.marketplace.org/2017/10/12/braille-versions-textbooks-help-blind-college-students-succeed/. [Accessed: 09-Aug-2019].

[9]"DAISY: What Is it and Why Use it?", Nfb.org. [Online]. Available: https://www.nfb.org/sites/www.nfb.org/files/images/nfb/publications/bm/bm11/bm1102/bm110210.htm. [Accessed: 10- Aug- 2019].

[10] A. Gilani, S. Qasim, I. Malik and F. Shafait, "Table Detection using Deep Learning." [Accessed 10 August 2019].

[11] "Image Detection, Recognition, and Classification with Machine Learning", Azati.ai, 2019. [Online]. Available: https://azati.ai/image-detection-recognition-and-classification-with-machine-learning/. [Accessed: 11- Aug- 2019].

[12]"ImageNet", Image-net.org. [Online]. Available: http://www.image-net.org/. [Accessed: 04- Sep- 2019].

[13]"Inception V3 Deep Convolutional Architecture for Classifying Acute Myeloid/Lymphoblastic Leukemia", Software.intel.com. [Online]. Available: https://software.intel.com/en-us/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic. [Accessed: 04- May- 2020].