DOCUMENT ZONE SEGMENTATION, CLASSIFICATION AND CHART INTERPRETATION

Project Id: 2022-024

Priyashan Sandunhetti S. H. S.

IT19187242

B.Sc. (Hons) Degree in Information Technology

(Specialization in Software Engineering)

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2022

DOCUMENT ZONE SEGMENTATION, CLASSIFICATION AND CHART INTERPRETATION

Project Id: 2022-024

Priyashan Sandunhetti S. H. S.

IT19187242

The dissertation was submitted in partial fulfilment of the requirements for the B.Sc. Special Honors degree in Information Technology (Specialization in Software Engineering)

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2022

DECLARATION

I declare that this is my own work and this proposal 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 my 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.

Name	Student ID	Signature
Priyashan Sandunhetti S. H. S.	IT19187242	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor   Date

…………………………..   ………….…………

(Dr. Anuradha Jayakody)

ABSTRACT

Any degree of vision impairment that interferes with a person's everyday life is referred to as a visual impairment, which can range from low vision to blindness. A perceptual, physical, or visual impairment that makes it difficult or impossible to read printed material is known as a print disability. If a person needs alternate access or accessible formats (Braille, audio) to read printed materials, that person is said to be print-disabled. Blindness, physical dexterity issues, learning disabilities, brain injuries, cognitive disorders, and literacy issues are all potential causes of print disability. Due to these issues, millions of people worldwide are unable to understand printed texts. Both the individual's daily life and their academics may be impacted by this. Even though there are tools for translating printed materials into text, most of them are inadequate to help people who cannot read printed materials and lack accessible features. Even when using the available accessible mediums to interpret printed materials there are still many barriers that cannot be overcome by these mediums. For instance, images, mathematics, tables and graphs are hard to interpret using methods like braille. Therefore, we propose a mobile-based solution to help print-disabled people comprehend printed documents that they would otherwise be unable to access. This application will read printed resources like text and paragraphs, mathematical formulas, tables, charts, and photographs using machine learning and image processing. In this particular research component, the main focus is to segment the captured document into different regions like text, figures, tables and mathematics and to interpret the identified charts within the document. Additionally, the program will be built with accessibility features like assistive image capturing and voice guidance that enable those who are print-impaired to use it independently.

Keywords: Assistive technology, Print disability, Vision impairment, Document image analysis, Chart image analysis, Image processing

ACKNOWLEDGEMENT

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Dr Anuradha Jayakody, for the immense support, guidance, and encouragement he provided during the research, as well as for his insightful feedback that helped to make the research project a success. His expertise in the field of assistive technology research was invaluable to this study. I would also like to thank the co-supervisor of this research project, Ms Shashika Lokuliyana, for always being willing to lend a hand. Many folks' perseverance was necessary to complete this dissertation. Throughout the duration of my study endeavour, their invaluable input and time aided me in achieving perfect success. In addition, I'd want to express my gratitude to the CDAP team for assisting us in completing the project successfully. Then, I'd like to offer my heartfelt gratitude to my friends for their unwavering support during this effort. Furthermore, I would like to thank my parents for their patience and time and for providing resources to acquire the needful and for other expenses. A special thanks to the fellow research members who worked alongside us to make this research more effective and successful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
CNN	Convolutional Neural Network
OCR	Optical character recognition
YOLO	You Only Look Once
SSD	Single Shot Detection
UI	User Interface
SVM	Support Vector Machines
CRAFT	Character Region Awareness For Text Detection
NDK	Native Development Kit
API	Application Programming Interface

1. INTRODUCTION

There is a huge percentage of the world population who cannot interpret printed materials due to various reasons. This inaccessibility can be caused by vision impairments, physical dexterity problems, learning and literacy difficulties. This inability to read printed material due to visual, physical or perceptual can be identified as print disabilities. These disabilities can affect individuals' day-to-day life as well as their education and literacy. Even though there are accessible mediums for print-disabled individuals to comprehend printed/written materials by methods like braille, ELIA Frames and audiobooks vast majority of necessary documents, books are not available in most countries. Also, most of the written materials cannot be converted into accessible mediums due to copyright and other various laws. This causes a global reading inequality and there are millions of people affected by this issue [1]. Even when using the available accessible mediums to interpret printed materials there are still many barriers that cannot be overcome by these mediums. For instance, images, mathematics, tables and graphs are hard to interpret using methods like braille. Also, when it comes to materials like legal and personal documents, they are rarely available in accessible mediums, and it is harder to get the assistance of a third party to interpret such documents. For these issues, we propose a solution to overcome the barriers of the print disabled to interpret printed materials. This solution will be implemented to run on users' mobile devices and it will be able to scan and interpret printed material in a vocal form. Furthermore, this solution will be implemented with several accessibility options to cater for every possible user. This system will consist of a mobile application and cloud backend where suitable image processing and computer vision algorithms are stored. The application will also be voice-assisted for better accessibility. In this research component, the author mainly focuses on the document capturing, segmentation and chart image interpretation of the overall research. There will be specific implementations for every mentioned sub-objective.

1.1. Background & Literature Survey

According to current statistics, WHO estimates there are at least 2.2 billion people with near or distance vision impairments [2]. Most of these individuals have difficulties when trying to interpret printed materials. Other than vision-impaired people, many forms of disabilities cause an individual to be print-disabled [3]. Overall, we can conclude that a considerable portion of the global population suffers from one or many types of print disabilities [1]-[3]. Reading is closely related to humans' everyday life and interacts with everyday elements like education, literacy, work, healthcare, justice, political participation and cultural belongings. Also, with reading being the main format of gathering information and communication, it is a necessary skill to survive in most modern societies. As shown in figure 1.1 our survey shows that the vast majority of participants considered reading to be very important in individuals' day-to-day life.

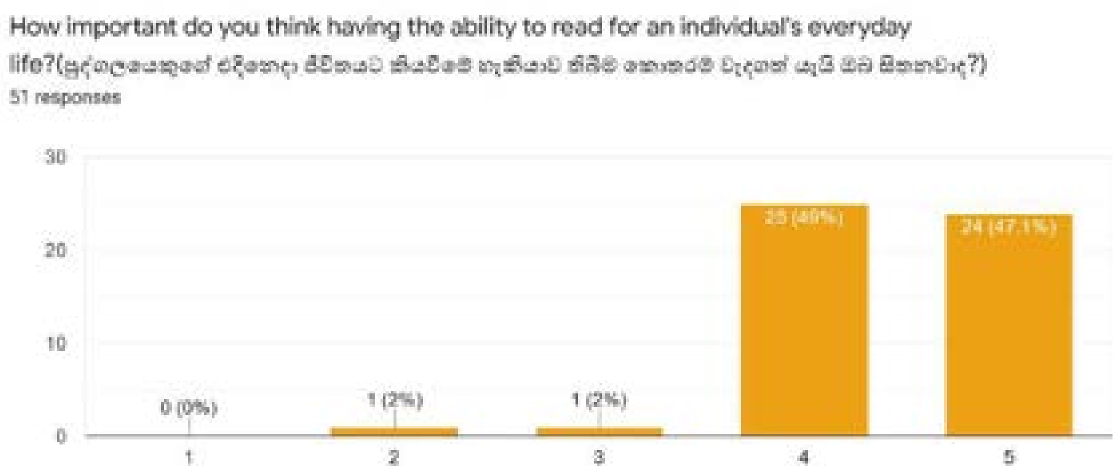


Figure 1.1.1. Survey responses to the importance of reading ability

Even though there are traditional solutions like braille [4] to aid these individuals to interpret printed materials, braille literacy of print-disabled individuals is as low as 10% [5]. Also due to the average cost of a brail book being higher than the normal issue and because of the low availability of braille books, braille cannot be considered as the best solution for print disability. For materials that are not available in accessible formats like braille and print-disabled individuals must have to rely on a third party. This third party can be a human or an assistive tool [6]. When considering another human who can access normal printed material to interpret the printed documents on print-disabled individuals' behalf there can be issues like privacy and mistrust. For personal, legal and confidential documents, a print-disabled individual cannot solely rely on another human being to assist.

With these issues, print disability has a huge impact on an individual's everyday life in many aspects [figure 1.2]. This discriminates against most basic human rights like the right to education, the right to work and even political and justice rights.

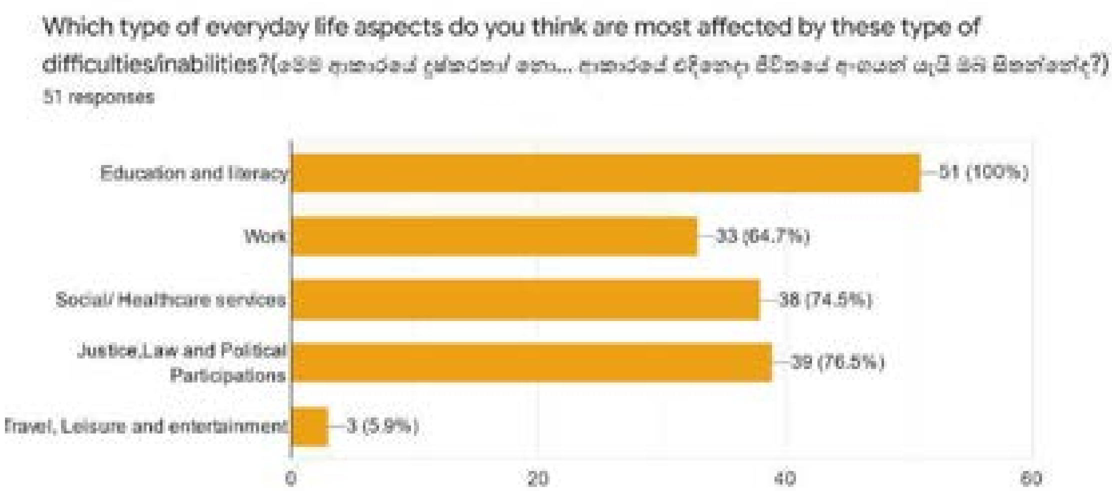


Figure 1.1.2 - Survey results for life aspects affected by reading inability

This causes a large gap between print-disabled individuals and the general population when it comes to reading rights and equality. As shown in figure 1.3 most of the participants in our survey agree that reading should be more accessible and equal for everyone and everyone must have equal rights to access printed materials.

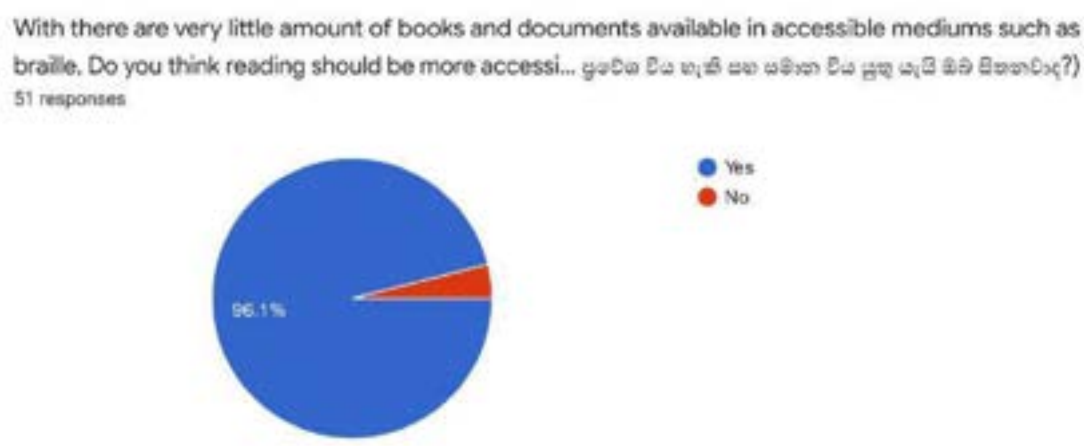


Figure 1.1.3 - Survey results on the reading equality and rights

With these issues in mind, we propose a solution to aid print-disabled individuals to interpret printed materials with better accessibility options to enable the application to be used by themselves. This solution mainly focuses on the Assistive Technology research area and consists of four main modules.

- Document capturing and document zone content classification module
- Chart interpretation module
- Image interpretation module
- Texts and mathematics interpretation module
- Tables interpretation module

This individual proposal mainly focuses on the first module which is the document capturing and document zone content classification module and the graph interpretation module. These modules fall under the computer vision research area and are crucial to the system because all the other algorithms' accuracies and work depend on the classification and it is a necessary and most important module of the system.

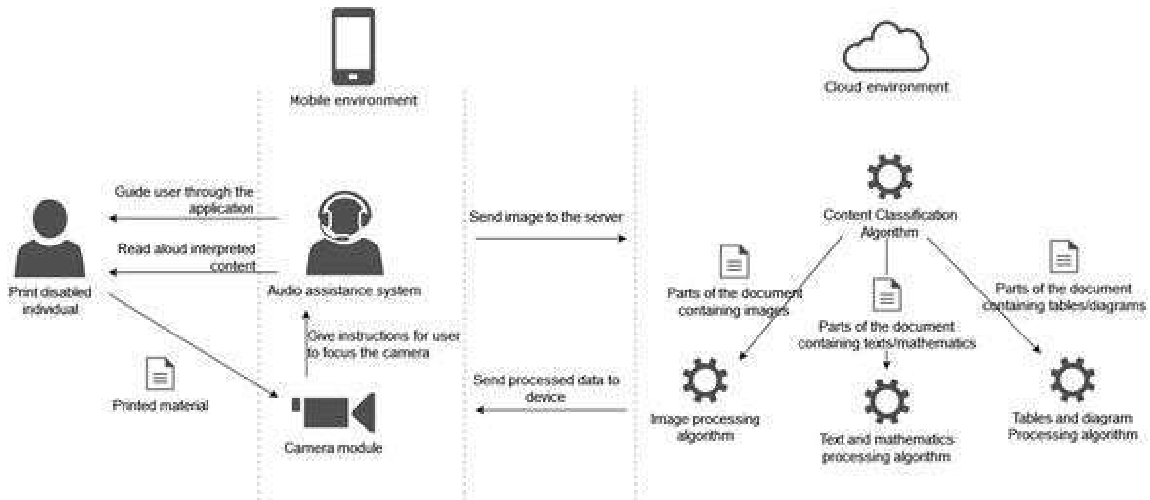


Figure 1.1.4 - Modules of the system

Furthermore, the chart interpretation module is necessary for the system because otherwise, the print-disabled individual won't be able to access and interpret graphs in a general printed material. Even though there are methods like tactile graphics to aid blind and vision-impaired people to read charts, the materials which are available in tactile graphics [10] are rare and useless in day-to-day life.

1.2. Research Gap

When considering the whole system, there are already implemented mobile tools to assist with print disabilities. But most of them are implemented for a limited audience and lack functions like table interpretation and mathematical equation interpretation [6],[7]. Furthermore, most of these solutions do not have the options to guide the user in the material-capturing process and to automatically capture the document when in the focus range. Also, there are promising solutions like wearable devices [8] to aid reading, this type of solution also lacks practicality when used by blind people because tracing in printed lines can be difficult for them. When looking for publicly available solutions in general app stores there are really good applications to interpret captured documents and images, but they lack the functions like interpreting every content (Text, images, tables) within a document at once and also, they include general UI and doesn't have accessibility options for vision-impaired users.

Furthermore, the results of the survey indicate that most of the participants do not know much about available assistive tools [Figure 1.5] and most of them agree that there are no sufficient accessibility options in available tools [Figure 1.6].

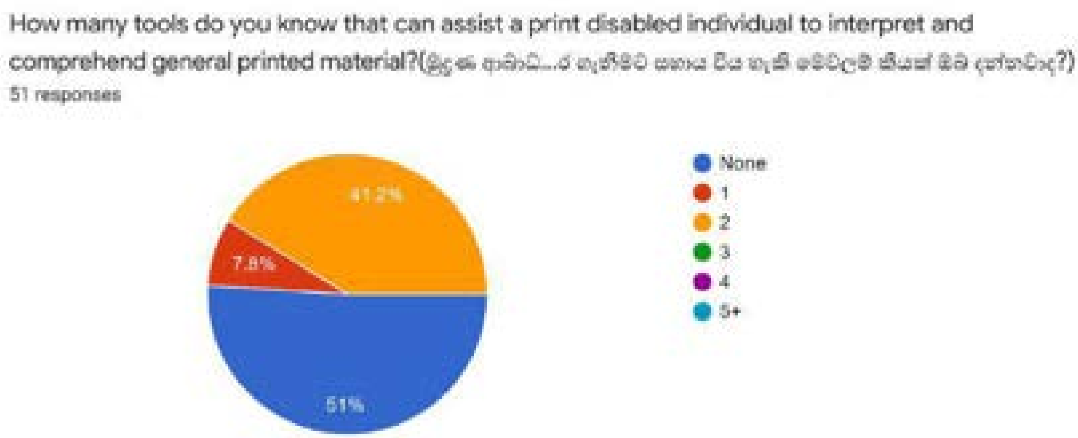


Figure 1.2.1 - Popularity of available assistive tools

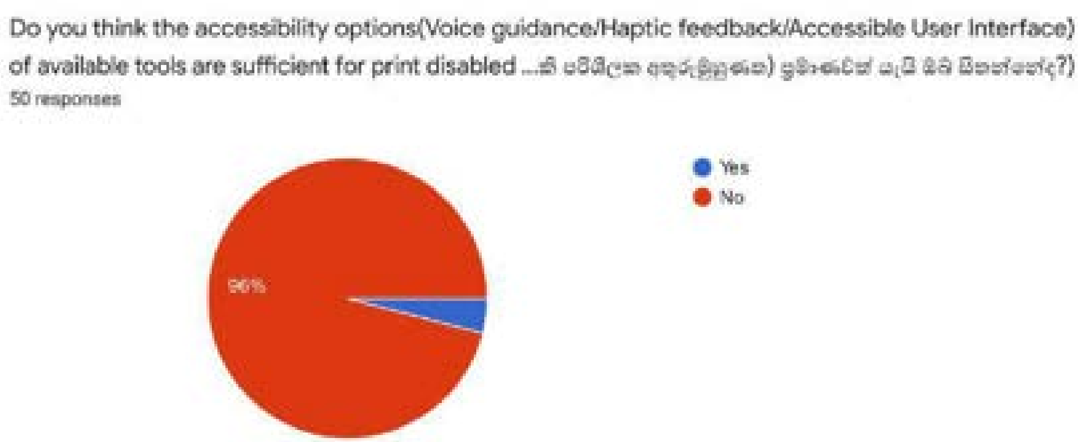


Figure 1.2.2 - Survey on accessibility options of available tools

The individual research component which is the document zone and content classification also already have been researched and many papers can be found on the topic [9]. However, most of them lack the ability to detect and classify all the necessary content types that could be included in a printed material [11],[12]. Andrea Corbelli et al. [13] (Research B) address this issue using the XY-cut algorithm to segment the document and classify the segmented document using heuristic methods for table detection and SVM classifier for other classes. This method is able to classify most of the available but lacks different chart classifications. Ranajit Saha et al. [14] (Research A) have developed a graphical object detection framework that can segment and classify tables, figures and equations separately but lacks the ability to separate charts and graphs from images. Furthermore, as shown in [15] (Research C) document layout analysis can even be done by using one-dimensional convolutional neural networks which are fast and economic in data usage that suits the performance capabilities of mobile devices. Also, this low computational cost means this approach can be implemented in even cloud environments without much cost. But this approach [15] also classifies charts and images under the same class as figures which is not feasible for our kind of document interpretation model. Also, because of the nature of the users that we are implementing the system for, there will be a need to crop out the document from the overall captured image and enhance the document by fixing the perspective issues and adjusting noise and lighting issues to increase the clarity.

Table 1.2.1 - Research gap for document segmentation and classification with existing systems

	Research A	Research B	Research C	Proposing solution
Detect and enhance the document from the overall scanned image (Distortions/Lighting)	No	No	No	Yes
Detect images and graphs separately	No	Yes	No	Yes
Classify text and mathematic expressions separately	No	Yes	No	Yes
Tabular structure detection	Yes	Yes	Yes	Yes
Optimized for mobile and cloud usage	No	No	Yes	Yes

The other part of this research component is the chart decoding module. This part of the research also requires a classification method for different types of charts. Then each chart will be decoded using proper methods and the decoded data will be turned into simple plain English in order to read aloud for print-disabled users as seen in [16] (Research P). As demonstrated in [16], it is possible to classify multiple types of charts with greater accuracy and generate alt-text for each type of chart using suitable algorithms. Furthermore, I'll be comparing this proposed solution in contrast to Research Q [17] and Research R [18]. Also, in every developed solution so far, all the methods only extract values from the tables but do not have the ability to match the values with the chart labels which is crucial for this kind of use case.

Table 1.2.2 - Research gap for chart classification and interpretation with existing systems

	Research P	Research Q	Research R	Proposing solution
Classify different types of charts	Yes	Yes	Yes	Yes
Extract data from bar charts	Yes (Vertical/ Horizontal /Stacked)	Yes (Vertical)	Yes (Vertical/ Horizontal /Stacked)	Yes (Vertical/ Horizontal /Stacked)
Extract data from pie charts	Yes	No	Yes	Yes
Extract data from line charts	No	No	Yes	Yes
Match extracted values into chart labels	No	No	No	Yes
Provide a textual description of chart data in plain English	Yes	No	No	Yes

By reviewing these available approaches, proposing solution can be implemented with many more novel and creative options which will be a hybrid solution for captured document classification and decoding the identified charts within the document.

1.3. Research Problem

According to the world health organization [19], there are 2.2 billion of the global population have near or distant vision impairment. Out of those vision-impaired populations, 300 million individuals can be considered as print-disabled [20]. Print disability is a difficulty or inability to read printed material due to a perceptual, physical or visual disability [21]. These disabilities vary from literacy difficulties to vision impairments and blindness. Learning disabilities like dyslexia can also be considered as print disabilities. Most of these people access documents via methods like braille or in the form of audiobooks. The issue is, there are 1%-7% of books are available in braille format globally [20]. Also, the majority of the documents and books that an individual has to use in day-to-day life are not available in accessible formats. As shown in figure 1.7, the majority of the participants that participated in the survey agree that there is a need for better accessible tool to aid print disabled individuals.

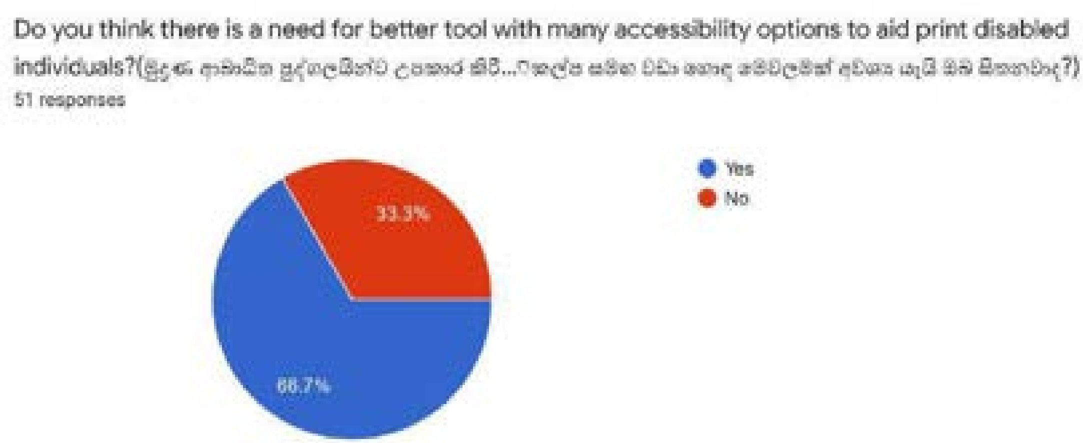


Figure 1.3.1 - Survey results on the need for better tools for print individuals

There are already available mobile tools to assist the reading of the materials but most of them either lack the accessibility options for the vision-impaired users and all of them doesn't convert complex features that can be found in a document such as tabular structures, charts, images and mathematical equations [6]. When it comes to charts, the only way for a print-disabled individual to access a chart in printed material is the tactile methods which are very rare and costly to produce. Even the available tactile charts can be hard to grasp for most individuals.

If we are to implement a mobile smartphone-based solution to address the print disability issue, there must be proper accessibility options in place [Figure 1.8]. In the solution, there is a need to scan the document automatically and the scanning process should also be assisted because most of the print-disabled individuals cannot properly capture a physical document using a smartphone.

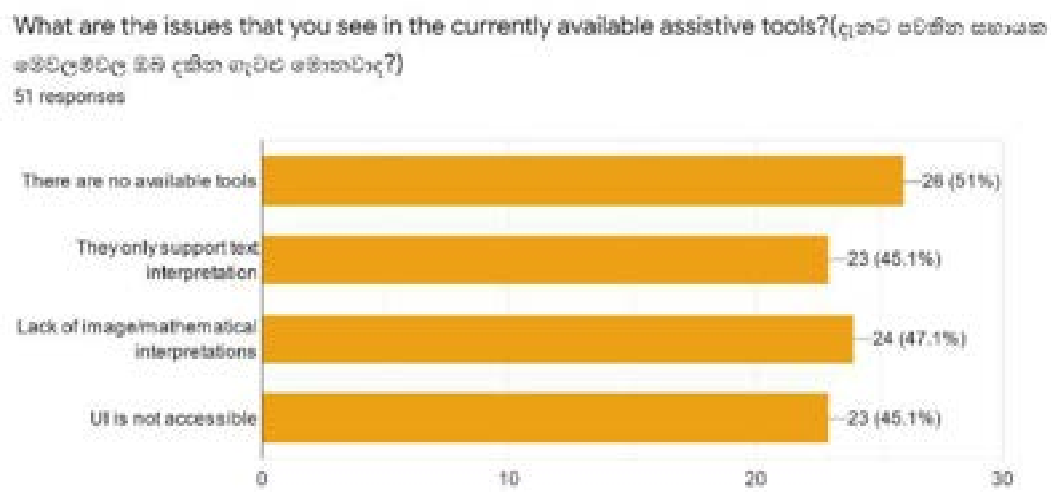


Figure 1.3.2 - Survey results on issues in the current accessibility tools for print disabled

In this proposal, the addressing issues are the assisted document capturing, enhancing, segmenting and classification of the segments and interpreting the identified chart into plain English.

2. RESEARCH OBJECTIVES

2.1. Main Objective

The main objective of this research is to develop assisted document-capturing methods, document content segmentation and classification of the segmented parts and the interpretation of the identified charts within the document.

2.2. Specific Objectives

- To assist the user with the document-capturing process

  The document capturing process must be assisted because most of the users that will be using the system will not be able to capture a physical document using their mobile phone camera properly.

- To crop out and enhance the captured document

  The captured image will include an outside area other than the document and the document will have perspective errors, lighting and noise issues which will affect the systems' accuracy greatly. So, the document has to be cropped out from the overall image and enhanced.

- To segment and classify the regions of the document

  The content in the document has to be segmented and classified in order to interpret the parts of the document using suitable methods.

- To classify the detected charts furthermore into different types of charts

  For each type of chart, there are different methods to extract data. Before extracting the data from the charts, there is a need to classify charts by their type.

- To extract the data from the charts and turn the extracted data into plain English sentences in a meaningful way.

  Finally, the identified charts in the captured document must be interpreted in simple English in order to read aloud to print disabled users.

3. METHODOLOGY

3.1. Methodology

This research component of the project includes several computer vision algorithms and image processing methods. The whole component will be distributed in two environments as client-side in the mobile phone and the classification and interpretation process will be done in the server-side cloud environment. In the mobile environment, there is a real-time computer vision model to aid the capturing process which is developed using OpenCV native libraries because it supports on-device machine learning and performs way better on mobile devices.

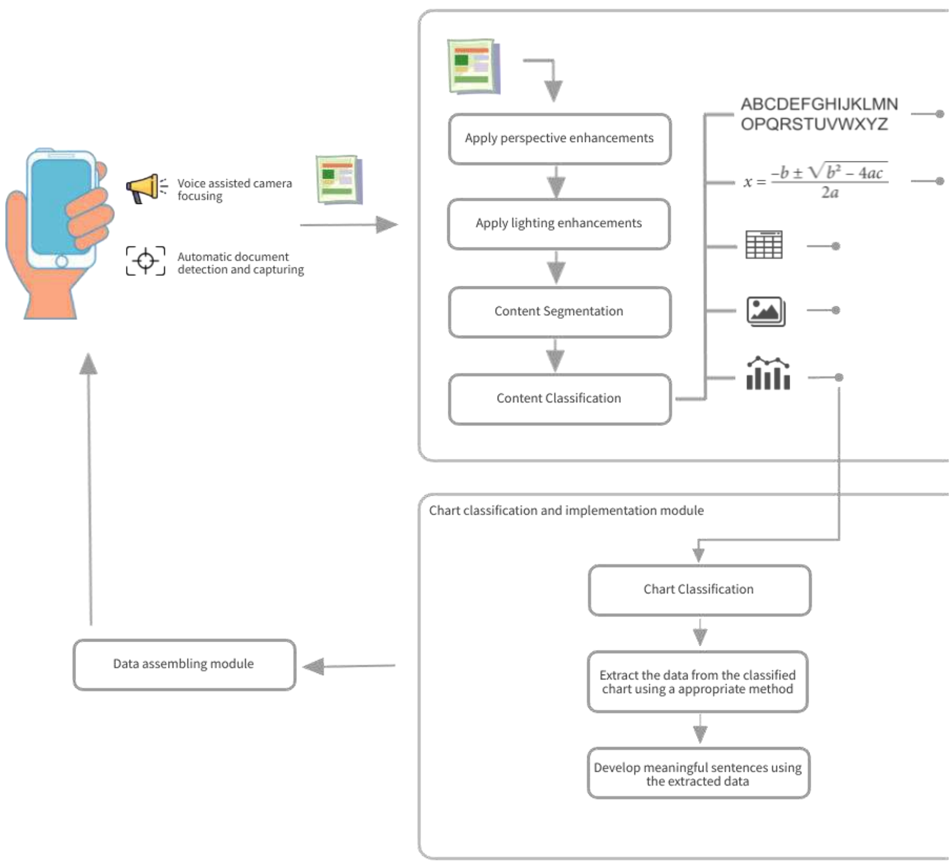


Figure 3.1.1 - Interconnection between implementing components

Then the captured image will be sent to the backend cloud environment to process. First of all, the image will be enhanced in the backend. The image enhancements are using python libraries like OpenCV and Python imaging library (PIL). Then the document segmentation is done by using Deep Learning Layout Detection models with the python library LayoutParser[23] which can be considered as the most recent groundbreaking development in the document layout segmentation research field. This has the capability to implement both the segmentation and classification process in a single model. For this use case, it uses multiple Facebook's Detectron2[26] object detection models pre-trained with several different datasets to identify Text, Tables, Mathematics, Images and Charts. Using multiple Detectron2 models author implemented an ensemble learning method Which is best suited for use cases like this. Also, for the different types of chart classifications, a trained CNN model is used because CNNs are specifically developed to deal with image data and excel at image processing [16]. For the classification processes, trained SVM (Support vector machines) models are widely used as well but for image processing purposes CNNs tend to be more accurate [22]. To classify the different chart types, the CNN that we developed is following the architecture of the VGG-16 convolutional neural network which excels at image-processing tasks.

For the data extraction from the charts, rule-based methods and deep neural networks can be used but for better efficiency and results, hybrid methods combining the advantages of rule-based methods and deep neural network-based methods prove to be more useful [18]. In our case, we use CRAFT PyTorch implementation as the backbone for chart data extraction and built separate rule-based data extraction methods for each chart type. Then to convert extracted data into meaningful sentences, predetermined templates are used for each chart.

3.1.1. System Architecture

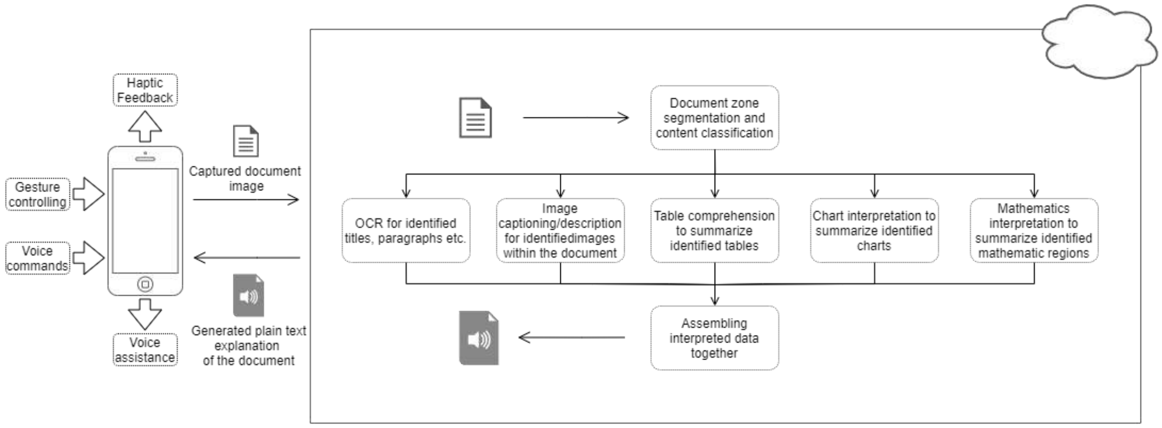


Figure 3.1.2 - High level architecture diagram of the system

The overall system resides in two main layers as a mobile application and a backend Django server. Document capturing and voice assistance is implemented in mobile application and computing heavy image processing components resides in the backend server since the mobile processors cannot handle heavy workloads like CNNs and image processing algorithms.

As shown in figure 3.1.3 first of all, the user will capture the document from the mobile device with assistance. Then the captured document image will be sent to the backend for processing. In the backend first, the document will be segmented into different regions and the segments will be classified. After that, the segmented parts will be sent to relevant algorithms to interpret. In this research component, the author will be implementing the chart interpretation component as well to describe identified charts within the document.

In the chart interpretation component first, there is a chart classification model which classifies charts into different types (Horizon bar, Vertical bar, Pie, Line). Then relevant data will be extracted from the classified chart image using suitable methods. Using a template sentence generation model for each chart type, the explanations will be generated using the extracted data. Finally, the data will be sent to the users' mobile application to output as speech-to-text output.

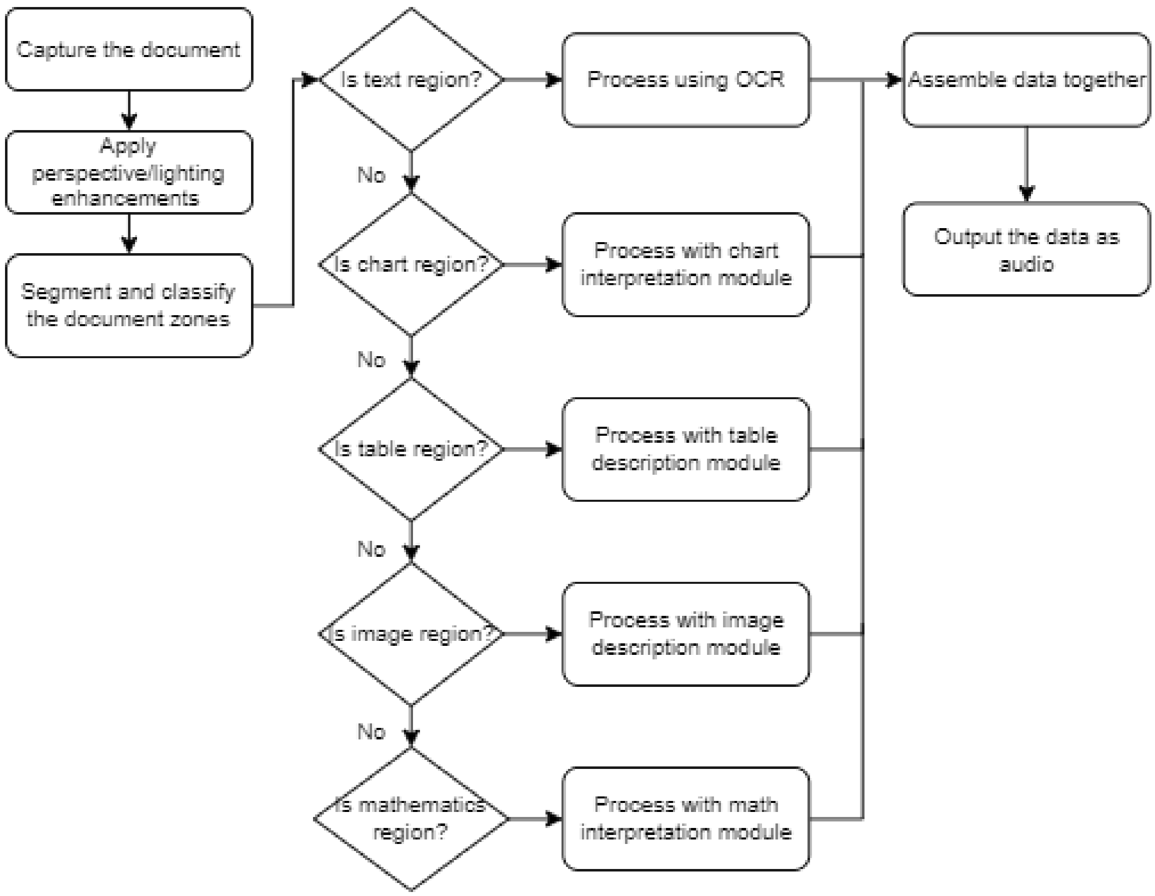


Figure 3.1.3 - Processing flow of the system

3.1.2. Data collection methods

For the automatic capturing there was no need for data collection because the main computer vision method that the author used is edge detection to automatically detect the document which does not require training. Then also for the document segmentation since we implement our method using multiple pre-trained Detectron2 models which are trained using PubLayNet and PRImA layout analysis datasets. For the chart type classification since the author implemented a CNN image classification model from scratch, there was a need for a dataset to train the classifier. We collected different chart images containing about 7000 different chart images from online sources. To maintain consistency, 4000 of those images are used to train the model with each of the chart types (Horizontal bar, Vertical bar, Pie, Line) having around 1000 images in the training dataset.

3.1.3. Tools and Technologies

Tools

- Anaconda Distribution
    - For the early development of machine learning models
- IntelliJ Idea
    - To implement the mobile application
- Google Colab
    - To train the machine learning models on cloud resources
- PyCharm
    - To implement the final Django backend
- Android Studio
    - To simulate the test environment of the mobile application

Technologies

- Layout Parser library
  - A deep learning-based document segmentation toolkit that used for the document segmentation in this implementation
- OpenCV-python
  - The Open-Source Computer Vision Library is used for many image manipulation uses in this implementation
- Native OpenCV for Android with Android NDK
  - Used to implement edge detection model for in-device machine learning. Used for assistive document capturing in this implementation
- Dart
  - Used to implement a mobile application with flutter
- Kotlin
  - Used to write android platform-specific code to implement real-time edge detection
- TensorFlow
  - Used to implement the Chart classification CNN model
- PyTesseract and Tesseract-OCR
  - To identify texts within the chart
- PyTorch
  - To implement chart interpretation models
- Django
  - To implement the backend API for the mobile app to interact with

3.2. Commercialization aspects of the product

The commercialization aspect of the mobile application includes mainly two areas. The marketing aspect of the app and revenue model of the product. There are many marketing/revenue generation methods that can be used in mobile applications but for this application, the options are limited due to the nature of the targeted user base. Following are the marketing and revenue generation strategies that are being used for the product.

- Marketing strategies
    - Use guest blogging to promote the app. Especially audio blogs which the print-disabled users mostly use.
    - There are many vision-impaired social influencers available globally on various platforms like YouTube, Instagram and TikTok. These influencers usually make reviews on accessible applications and devices. Sponsoring their content is a great marketing strategy for this app.
    - Host an event to launch the application and invite journalists with the expertise in the assistive technology area
    - Audiobooks are one of the most popular education mediums with print-disabled users. Posting advertisements within audiobook platforms like audible will improve the user base greatly.
    - Podcasts are one of the most trending entertainment methods in the current decade. Promoting the application through podcasts is a good commercialization strategy for this kind of a app.
- There are many revenue models that can be used in a mobile application. In-app purchases, In-app advertising, Upfront selling (Paid app), Affiliate marketing, Data monetization and Subscription models are some of them. Since this is an application to aid disabled individuals there are social and ethical concerns when it comes to some of the mentioned monetization models. For this application, the

revenue model will be a donation-based model where anyone can donate any amount that they wish which will be used to manage the maintenance and improvement costs of the application.

3.3. Testing and Implementation

3.3.1. Implementation

In this section, the author will discuss about implementations of each implemented module in detail.

- Assisted document capturing

  This module resides in the frontend part of the system within the mobile application. Mobile application is mainly developed with the flutter framework. To identify the document when the user opens the camera author had to use a real-time image processing method. Due to this reason, the image processing component couldn't be moved into the backend because of the network latencies the output will not be real-time. So, the author was restricted to using something lightweight that the mobile phone processing units can handle. TensorFlow lite is a really good solution for this, and it works well with the flutter framework however the technology is not matured yet and running object detection models like YOLO or SSD as TensorFlow lite models will still be heavy for the majority of mobile processors in the current mobile phone market. Ultimately, the decision was to implement document detection with edge detection methods. This can be done with the use of OpenCV native libraries for android. Since with flutter, there is the ability to manipulate native source code to implement and interact with native APIs like camera APIs, the processing will be faster.

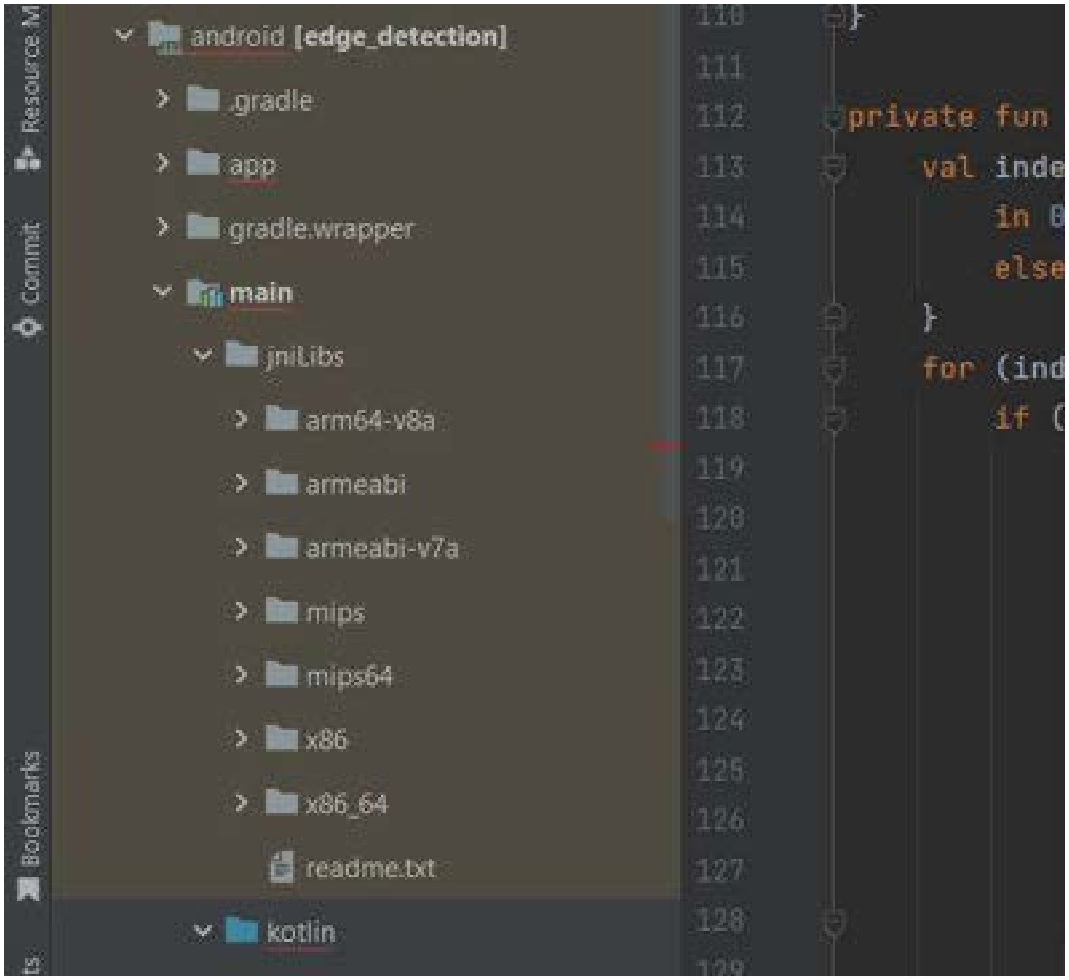


Figure 3.3.1 - Usage of the Open-CV native libraries in the project

The process of edge detection consists of image enhancement, contour detection and deciding the corners using the detected contours. All this processing is applied to the camera image in real-time.

```kotlin
private fun findContours(src: Mat): ArrayList<MatOfPoint> {

    val grayImage: Mat
    val cannedImage: Mat
    val kernel: Mat = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, Size( width: 9.0,  height: 9.0))
    val dilate: Mat
    val size = Size(src.size().width, src.size().height)
    grayImage = Mat(size, CvType.CV_8UC4)
    cannedImage = Mat(size, CvType.CV_8UC1)
    dilate = Mat(size, CvType.CV_8UC1)

    Imgproc.cvtColor(src, grayImage, Imgproc.COLOR_BGR2GRAY)
    Imgproc.GaussianBlur(grayImage, grayImage, Size( width: 5.0,  height: 5.0),  sigmaX: 0.0)
    Imgproc.threshold(grayImage, grayImage,  thresh: 20.0,  maxval: 255.0, Imgproc.THRESH_TRIANGLE)
    Imgproc.Canny(grayImage, cannedImage,  threshold1: 75.0,  threshold2: 200.0)
    Imgproc.dilate(cannedImage, dilate, kernel)
    val contours = ArrayList<MatOfPoint>()
    val hierarchy = Mat()
    Imgproc.findContours(
            dilate,
            contours,
            hierarchy,
            Imgproc.RETR_TREE,
            Imgproc.CHAIN_APPROX_SIMPLE
    )
    contours.sortByDescending { p: MatOfPoint -> Imgproc.contourArea(p) }
    hierarchy.release()
    grayImage.release()
    cannedImage.release()
    kernel.release()
```

Figure 3.3.2 - Contour detection with Open-CV in mobile app

```kotlin
private fun getCorners(contours: ArrayList<MatOfPoint>, size: Size): Corners? {
    val indexTo: Int = when (contours.size) {
        in 0 ≤ .. ≤ 5 -> contours.size - 1
        else -> 4
    }
    for (index in 0 ≤ .. < contours.size) {
        if (index in 0 ≤ .. ≤ indexTo) {
            val c2f = MatOfPoint2f(*contours[index].toArray())
            val peri = Imgproc.arcLength(c2f,  closed: true)
            val approx = MatOfPoint2f()
            Imgproc.approxPolyDP(c2f, approx,  epsilon: 0.02 * peri,  closed: true)
            //val area = Imgproc.contourArea(approx)
            val points = approx.toArray().asList()
            val convex = MatOfPoint()
            approx.convertTo(convex, CvType.CV_32S)
            // select biggest 4 angles polygon
            if (points.size == 4 && Imgproc.isContourConvex(convex)) {
                val foundPoints = sortPoints(points)
                return Corners(foundPoints, size)
            }
        } else {
            return null
        }
    }

    return null
}
```

Figure 3.3.3 - Detecting document corners with detected contours

- Document layout segmentation and content classification

  This component is the first component that the captured document image will go through in the backend server. Implemented in python and mostly used libraries are Open-CV and LayoutParser. The LayoutParser library uses the state-of-the-art object detection library Detectron2 which was developed by Facebook research for the detection and segmentation of different document zones. LayoutParser has separate pre-trained deep learning models with different datasets designed for document layout analysis. In this implementation, the layout analysis is done by the ensemble learning method with multiple pre-trained LayoutParser models. Specifically, the models that were trained with PubLayNet and PRImA layout analysis datasets. The model trained with the PRImA dataset is capable of separately identifying mathematical regions and image regions while the PubLayNet dataset trains models to identify general document areas like texts, titles and tables more accurately.

```python
def segment(filePath):

    publay_model = lp.Detectron2LayoutModel(config_path='models/document_segment/PubLayNet/config.yaml',
                                            model_path='models/document_segment/PubLayNet/model_final.pth',
                                            extra_config=["MODEL.ROI_HEADS.SCORE_THRESH_TEST", 0.8],
                                            label_map={0: "Text", 1: "Title", 2: "List", 3:"Table", 4:"Figure"})

    image = cv2.imread(filePath)
    image = image[..., ::-1]

    layout = publay_model.detect(image)
    result = lp.draw_box(image, layout, box_width=3)

    result.save('media/res1.jpg')

    text_blocks = lp.Layout([b for b in layout if b.type=='Text'])
    figure_blocks = lp.Layout([b for b in layout if b.type=='Figure'])
    table_blocks = lp.Layout([b for b in layout if b.type=='Table'])
    list_blocks = lp.Layout([b for b in layout if b.type=='List'])
    title_blocks = lp.Layout([b for b in layout if b.type=='Title'])
```

Figure 3.3.4 - Usage of LayoutParser and Detectron2 for document segmentation

- Chart classification

  Chart classification is done using a Convolutional neural network which is similar to the architecture of VGG-16 but with fewer parameters. To train this model, the author collected 4000 different chart images from various online sources. The model is trained with an SGD optimizer with a learning rate of 0.001 for 50 epochs.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 conv2d_1 (Conv2D)           (None, 224, 224, 64)      36928

 max_pooling2d (MaxPooling2D  (None, 112, 112, 64)     0
 )

 conv2d_2 (Conv2D)           (None, 112, 112, 128)     73856

 conv2d_3 (Conv2D)           (None, 112, 112, 128)     147584

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 128)      0
 2D)

 conv2d_4 (Conv2D)           (None, 56, 56, 256)       295168

 conv2d_5 (Conv2D)           (None, 56, 56, 256)       590080

 max_pooling2d_2 (MaxPooling  (None, 28, 28, 256)      0
 2D)

 conv2d_6 (Conv2D)           (None, 28, 28, 512)       1180160

 conv2d_7 (Conv2D)           (None, 28, 28, 512)       2359808

 conv2d_8 (Conv2D)           (None, 28, 28, 512)       2359808

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 512)      0
 2D)

 conv2d_9 (Conv2D)           (None, 14, 14, 512)       2359808

 conv2d_10 (Conv2D)          (None, 14, 14, 512)       2359808

 conv2d_11 (Conv2D)          (None, 14, 14, 512)       2359808

 max_pooling2d_4 (MaxPooling  (None, 7, 7, 512)        0
 2D)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 4096)              102764544

 dense_1 (Dense)             (None, 4096)              16781312

 dense_2 (Dense)             (None, 4)                 16388

=================================================================
Total params: 133,686,852
Trainable params: 133,686,852
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.3.5 - Architecture of chart classification Convolutional Neural Network

- Chart Interpretation

For the data extraction from the classified chart images, there are separate algorithms defined for each chart type. All of the chart interpretation models are developed on top of the CRAFT (Character Region Awareness for Text Detection) [24] model. For bar charts and line charts first, we identify the axes by detecting the longest horizontal and vertical lines. Then for the vertical and horizontal bar charts, we used connected component analysis to fit rectangles for each bar in the chart and calculate the values. Then we can identify corresponding labels using the CRAFT model.

For pie charts, we perform gradient analysis to identify the boundaries of the chart. After detecting the boundaries, a calculation is done to calculate the volume of each slice. Then corresponding labels are identified using the Tesseract-OCR [25] engine and the colours of the legend correspond to the slices.

After extracting the data, the plain English sentences are generated using a general template-based sentence generation method.

3.3.2. Testing

The implementation will be tested using different kinds of testing methods such as unit testing, integration testing and user acceptance testing because appropriate testing guarantees that bugs and issues are identified early within the life cycle of the application. After all the testing it should be released. If there are any issues in the testing phase it should be modified before releasing. After the completion and approval of the user testing the mobile application will be released.

- Unit testing

  Following are the test results for the developed assisted document-capturing component.

Table 3.3.1 - Tests of document-capturing component

Test case #	**Test case**	**Result**	**Comments**
001	The camera widget opens from the homepage button	Pass	Works without any issues
002	There is an overlay on the camera screen to show edge detection	Pass	Overlay appears and visible
003	Edge detection detects documents in general scenarios	Pass	Document edges are detected
004	Edge detection detects documents in extremely low-light scenarios	Fail	Does not detect the exact edges properly
005	Edge detection detects documents when there is other document in the background	Fail	Algorithm gets confused with other documents
006	Detected edges are shown through camera overlay real-time	Pass	Edges are show as overlay
007	The image is captured with the capture button	Pass	-
008	The crop page shows up after the capture with the captured image	Pass	-
009	The detected document is automatically selected for cropping	Pass	Final overlay position is selected automatically for cropping
010	The selected region of the image is cropped	Pass	-

	with the crop button		
011	The cropped image is sent to the backend server	Pass	-

Following are the test results for the document segmentation component.

Table 3.3.2 - Tests for document segmentation component

Test Case #	**Test Case**	**Result**	**Comments**
001	The document is successfully uploaded to the algorithm	Pass	The cropped document is uploaded without any issues
002	Image preprocessing makes the image enhanced	Pass	-
003	Detects every major region of the document	Pass	Detects all the zone in the document
004	Detects text regions	Pass	-
005	Detects title regions separately	Pass	-
006	Detects tables	Pass	-
007	Detects figures	Pass	-
008	Detects mathematic regions	Pass	-
009	Output includes all the regions with corresponding coordinates	Pass	Output is sufficient and can be used to segment the document image

Following are the test results for the chart classification component

Table 3.3.3 - Tests for chart classification component

Test Case #	Test Case	Result	Comments
001	The chart image is successfully uploaded to the algorithm	Pass	The cropped chart is uploaded
002	Chart image preprocessing makes the image match the neural network input	Pass	Image is preprocessed and the output matches the 224*224*3 neural network input
003	Successfully loaded the model checkpoint into the neural net	Pass	-
004	Detects pie charts	Pass	Pie charts are detected with good accuracy
005	Detects vertical bar charts	Pass	Does not misclassify with horizontal bar charts and vice versa
006	Detects horizontal bar charts	Pass	-
007	Detects line charts	Pass	-
008	Detects figures without any charts separately	Pass	Images sent through the algorithm without any charts are detected

Finally, for the chart interpretation component, the test results are as below,

Table 3.3.4 - Tests for chart interpretation component

Test Case #	**Test Case**	**Result**	**Comments**
001	Detect all the bars in the vertical bar chart	Pass	All the bars are detected
002	Detect all the bars in the horizontal bar chart	Pass	All the bars are detected
003	Detect all the slices on the pie charts	Fail	In some scenarios, some slices are missing from the detection
004	Detects all the points in the line charts	Fail	Some of the points are missing from the detection
005	Detect labels, and legends in the charts	Pass	All the texts are detected and generated OCR output
006	The output consists of approximate values	Pass	Output has acceptable values
007	The output consists of labels	Pass	Detected label OCR results are outputted
008	Proper sentences are generated with detected values	Pass	Generated sentences are meaningful and useful to the user

- End-user testing

The author tested the application with three different users to complete the end-user testing of the implementation. The personal details of the users are not included in this document due to ethical reasons. User 01 is a general user and does not have any vision impairments. User 02 has the ability to see but cannot read a document without corrective lenses. User 03 corresponds to a blind person who does not have the ability to see.

Table 3.3.5 - Test results of end-user testing

Test Case #	**Test Case**	**User 01**	**User 02**	**User 03**
001	Successfully opens the mobile app	Pass. Opens the app without any issues.	Pass. Opens the app without any issues	Pass. Opens the app with the help of an android voice assistant.
002	Navigates to the camera screen with the button on the homepage	Pass. Navigates to the camera without any issues.	Pass. Navigates to the camera without any issues.	Pass. Navigates to the camera with the help of voice guidance and big buttons.
003	Able to capture a document	Pass. Captures a document without any issues.	Pass. Captures a document without any issues.	Pass. Captures the image with the given button.
004	Able to crop the document	Pass. Crops the document without any issues.	Pass. Able to crop the document.	Pass. Since the detected document is already selected for cropping. Crops without any issues.

- Compatibility testing

Finally, series of compatibility tests have been carried out to ensure that the application is working properly across different platforms with various system specifications. Mobile applications should be of excellent quality and compatible with all hardware, software, operating systems, platforms, and so on, which is achieved by using compatibility testing. Compatibility can be ensured by using compatibility testing, which looks for faults before the product is given to the final user. This testing establishes or validates that the product fits all of the specifications established and agreed upon by both the developer and the end user. Specifically in this kind of an assistive application the compatibility should be high without any major application differences in different platforms.

Table 3.3.6 - Test results of compatibility testing

Device	**OS**	**Version issues**	**Screen resolution issues**	**OpenCV native library support issues**	**UI issues (Content alignment, Overlapping content)**
Xiaomi Pocophone F1	Android 11	Not found	Not found	Not found	Not found
Xiaomi Pocophone X3	Android 13	Not found	Not found	Not found	Not found
Samsung galaxy grand prime	Android 5	Not found	Not found	Not found	Not found

4. RESULTS AND DISCUSSION

This section will provide an analytical summary of the major aspects of the implementation, results, research findings, and concluding discussion, together with the necessary diagrams and screenshots.

4.1. Results

- Results of assistive document capturing

  The main functionality of this component is to detect the document using real-time edge detection and assist the user with capturing and cropping process. This component is implemented in the mobile application and consists of UI's which are catered to be used by visually impaired users. Furthermore, the edge detection model which resides within the camera application successfully detects the document edges given that there is enough lighting within the scene.

Figure 4.1.1 – Real-time edge detection results shown as overlay

The detected corners of the document are automatically selected for cropping in the crop window of the application. Users can do smaller adjustments for the crop or users can make the cropping process to be automatic because the application will automatically detect the exact points to crop the document from the overall image.

Figure 4.1.2 - Detected edges are automatically selected for cropping

After the cropping process, the final image can be shown to the user and will be sent to the backend server for interpretation or the user has the option to re-adjust the cropping of the image.

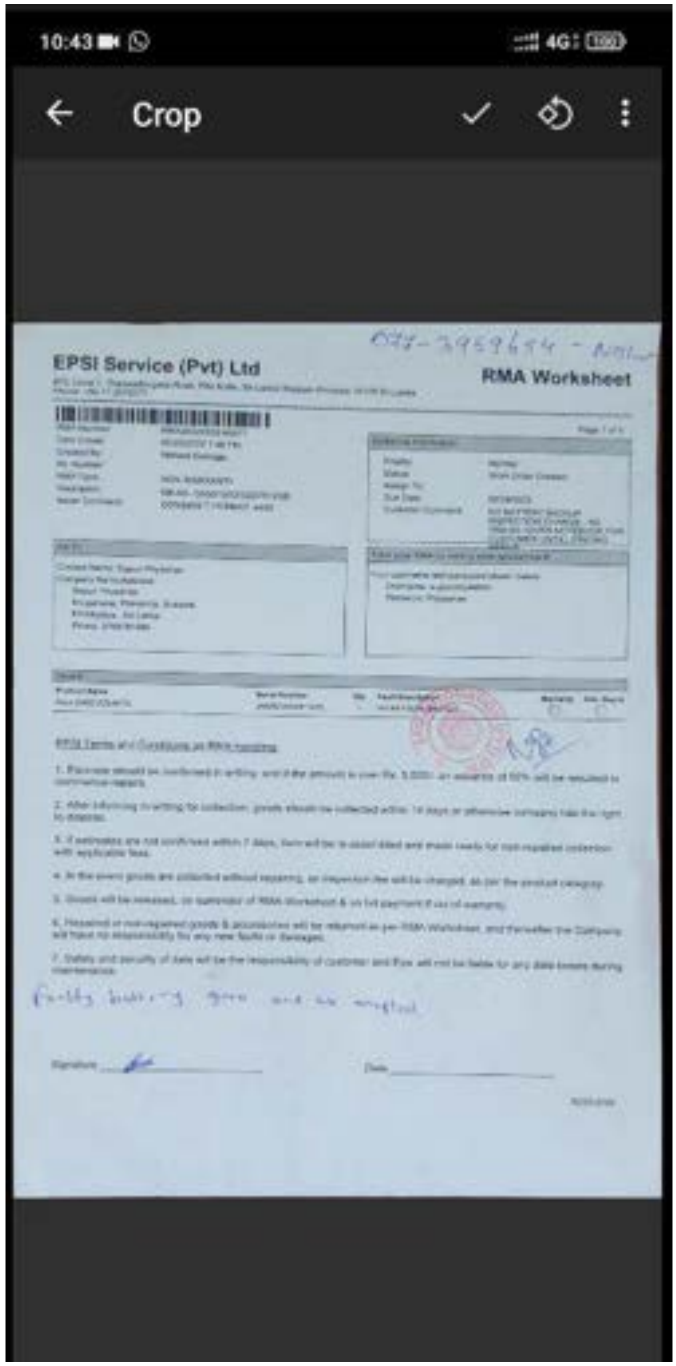


Figure 4.1.3 - Final cropped result

- Results of the document segmentation module

  Since this component is developed on a state-of-the-art object detection model (Detectron2) and pre-trained with huge datasets with hundreds of thousands of images the accuracy of the component is very high. To be specific, the model trained with the PubLayNet dataset includes 360 hundred thousand images and the size is about 102GB. So, the results are highly accurate and the model is very reliable.

Figure 4.1.4 - Results of the document segmentation algorithm

- Results of the chart classification model

  This component consists of a custom-defined modified version of a VGG-16-like convolutional neural network for the classification of the images. The dataset is also collected by the author and the model is trained from scratch. The model has shown ~99% accuracy within the training dataset as well as the validation dataset.

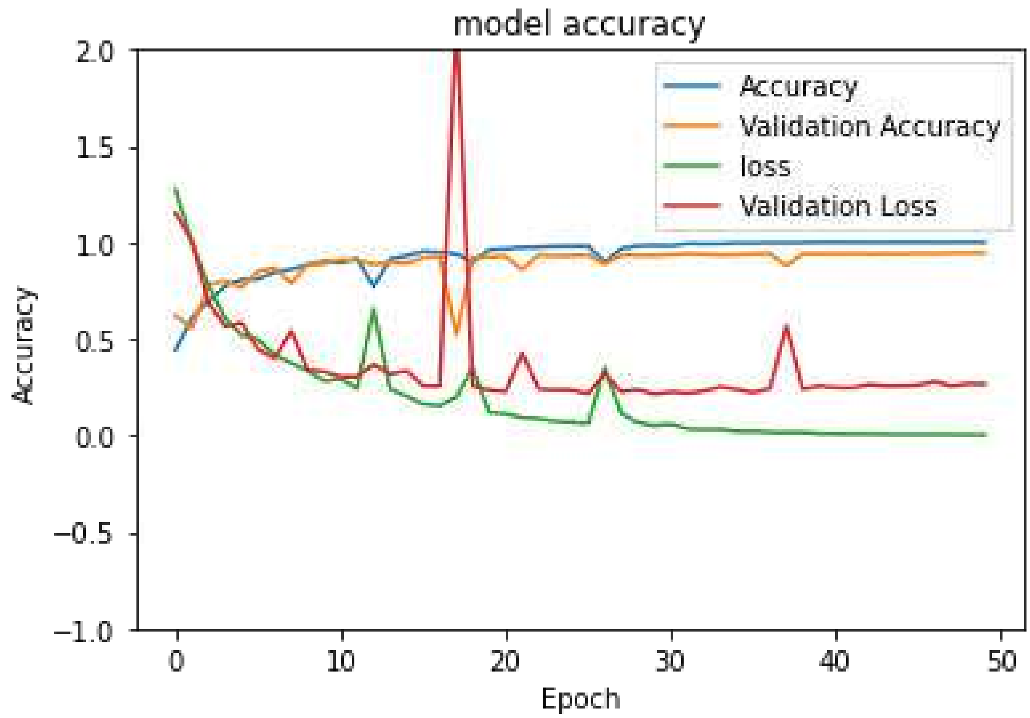


Figure 4.1.5 - Training and validation accuracy of chart classification model

Furthermore, the confusion matrix for the model results for the testing (consists of 1000 chart images) dataset shows that pie charts are classified with ~99% accuracy while horizontal bar charts are classified with ~100% accuracy. Also, the vertical bar charts and line charts can be classified with ~96% accuracy with the trained model.

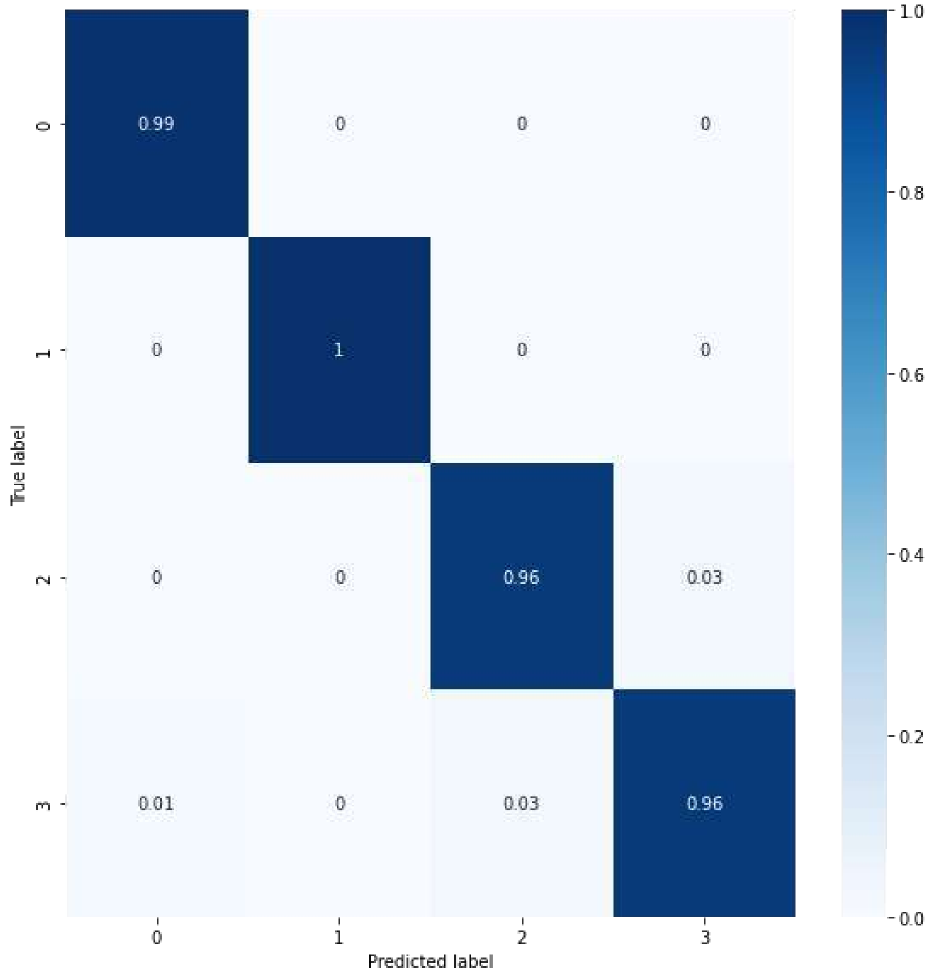
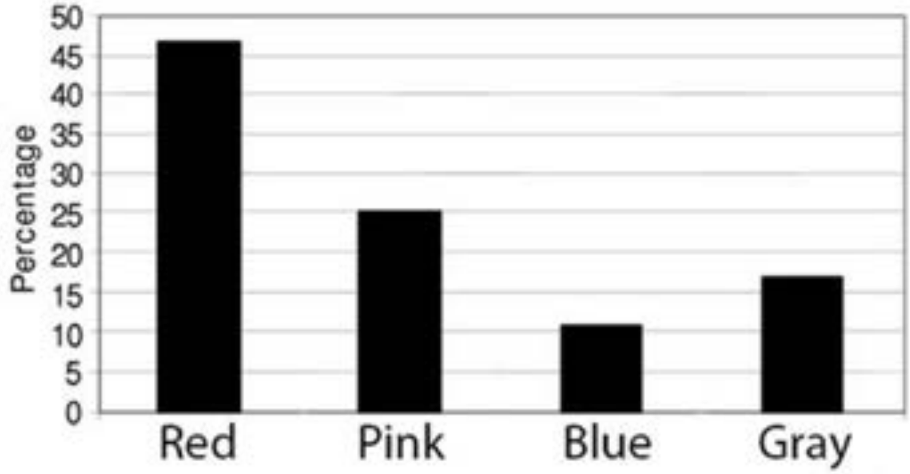


Figure 4.1.6 - Confusion matrix of trained CNN chart classifier

- Results of interpreting charts

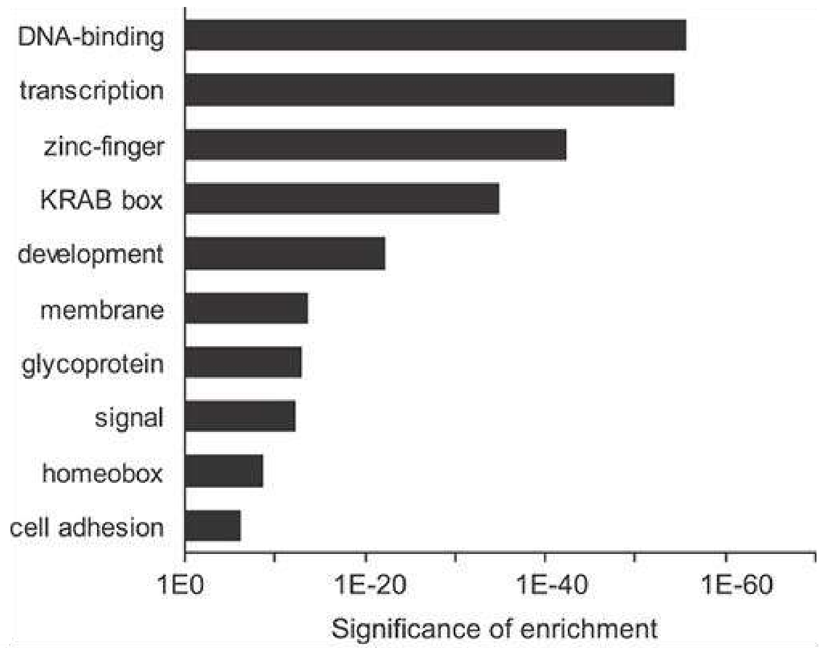
  This component has four separate interpretation methods for each chart type. For horizontal and vertical bar charts, it detects the bars and detects the values and labels as well. The accuracy of the OCR results of the images are vary and depends on the OCR module we used called CRAFT-OCR. However, the used OCR engine is highly accurate and reliable for this use case.




```
{
    "response": "There is a Vertical bar chart within the
        document. There are 4 bars in the given bar chart.
        Red has the maximum value and Blue has the minimum
        value. The bar Red has a value around 50. The bar
        Pink has a value around 25. The bar Blue has a value
        around 10. The bar Gray has a value around 15. ",
    "type": "Vertical bar chart"
}
```


Figure 4.1.7 - Result of vertical bar chart interpretation

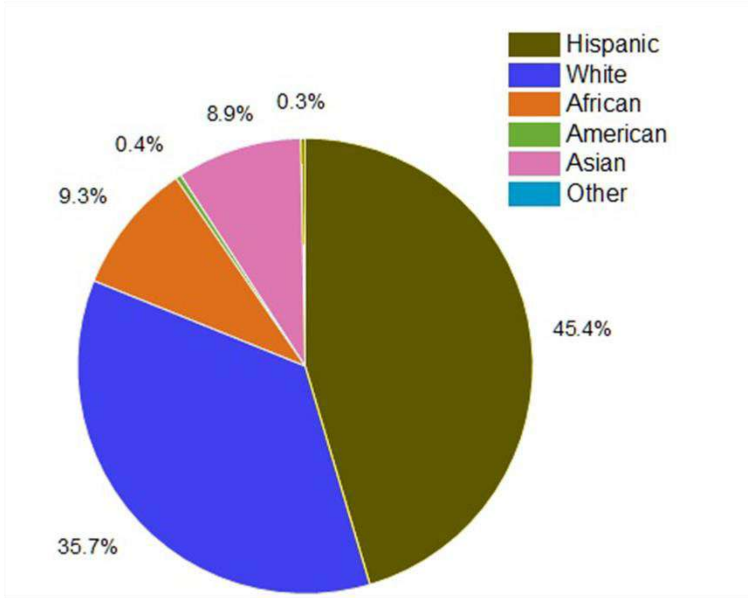
For all the bar charts, the bar count detection is highly accurate, and the output is given as a range value rather than an exact value to maintain consistency.




```
{
    "response": "There is a Horizontal bar chart within the document. There
        are 10 bars in the given horizontal bar chart. zinc-finger has the
        maximum value and DOX has the minimum value. The bar adhesion has a
        value around 1E-20. The bar signal has a value around 1E-20. The bar
        signal has a value around 1E-20. The bar signal has a value around
        1E-20. The bar membrane has a value around 1E-20. The bar DOX has a
        value around 1E-20. The bar DOX has a value around 1E-40. The bar
        DOX has a value around 1E-40. The bar zinc-finger has a value around
        1E-60. The bar transcription has a value around 1E-60. ",
    "type": "Horizontal bar chart"
}
```

Figure 4.1.8 - Result of horizontal bar chart interpretation

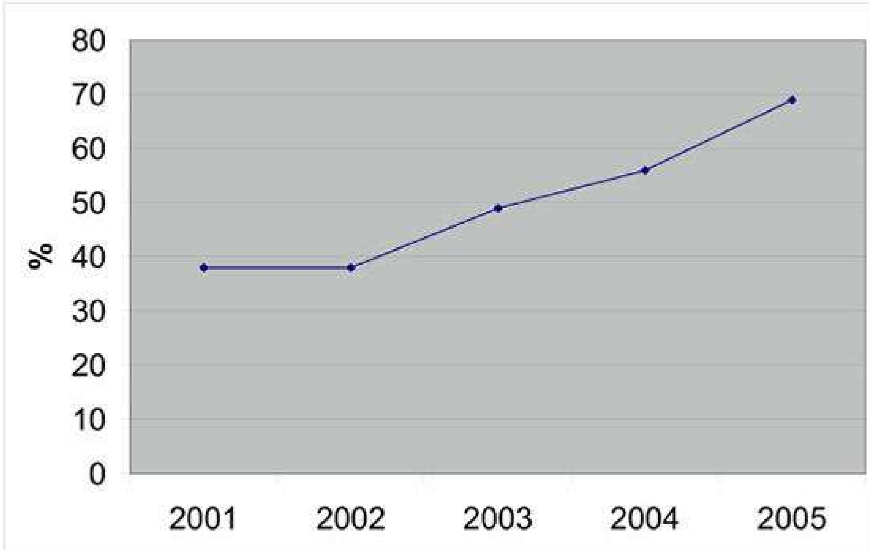
Furthermore, for the pie charts, we use a python dictionary consisting of every possible RGB value to cross-check against the chart legend and the pie chart. We take the values by calculating the area of the slices rather than just reading the text in the chart. So even pie charts without any numerical values are supported. This method also achieved great results. However, the most complicated part of this type of chart image processing is to match the corresponding label to the calculated value. So, there are some minor issues that were seen when we were testing the algorithm where values matched the wrong labels.




```
{
    "response": "There is a Pie Chart shown in the given figure. There are
        4 slices in the given pie chart. Slice White has the maximum value
        and the slice African has the minimum value. The slice White has an
        approximate value of 34.73. The slice African has an approximate
        value of 8.24. The slice Asian has an approximate value of 8.08.
        The slice White has an approximate value of 29.93. ",
    "type": "Pie Chart"
}
```


Figure 4.1.9 - Result of pie chart interpretation

Since line charts do not have easily recognizable features like bars or slices it is hard to detect the exact values of the charts. The smaller points and thin lines in the line charts are hard to detect and hard to differentiate from other features (axes, legend) within the chart. Out of all four of the chart types of line chart has the least reliable output but the output is accurate enough to generate a meaningful sentence for our use case.




```
{
    "response": "There is a Line chart within the document. There are 5 points
        in the given line chart. Point 2005 has the maximum value and 2001 has
        the minimum value. The point 2001 has a value around 70. The point 2002
        has a value around 70. The point 2003 has a value around 70. The point
        2004 has a value around 70. The point 2005 has a value around 70. ",
    "type": "Line chart"
}
```


Figure 4.1.10 - Result of line chart interpretation

Overall, the chart interpretation module gives highly accurate results compared to currently available studies and applications. Since this is done with mostly rule-based methods, the performance of the component is also high. The final generated explanations of the charts are insightful and useful for print-disabled users.

4.2. Research Findings

The main outcome of this study was to develop and mobile application with various images processing techniques to assist print-disabled individuals to assist in their day-to-day life by reading printed materials. To implement such a complex image processing/AI task most of the currently available state-of-the-art image processing/deep learning techniques have been used. The outcome of the research reveals that currently available popular image processing/deep learning techniques like convolutional neural networks (VGG-16), OCR techniques (CRAFT-OCR) and object detection models (Detectron2) are reliable and can be applied to implement various applications in many study areas. However, to interpret/annotate chart images there is no readily available and reliable technique that can be used for an application like this. Also, there are very good datasets for the task of document image analysis like PubLayNet and Prima layout analysis dataset with a huge number of records but there are no proper datasets for chart image annotation. This is one of the main reasons that we had to implement rule-based chart image analysis methods from scratch. Furthermore, when we were testing with digital computer-generated images the accuracy of the techniques was very high but when we are uploading a photo which is captured from a smartphone camera there are some inconsistencies like OCR not detecting every text region and missing some. So, there are some improvements that can be done to these image processing techniques to get better uses from them.

From the aspect of print disability and assistive technologies, as the author discussed in the literature review, there are no proper methods for print-disabled individuals to interpret printed charts or documents in general. Most of the available applications lack practicality or do not have enough features or accessibility options. As shown in the survey results most of the users prefer having better accessibility options and most of them agree that the currently available applications do not have enough features and accessibility options.

4.3. Discussion

This chapter covers in detail the many methods we've identified for resolving difficulties in specific modules. In addition, this chapter analyzes the techniques and explains why we selected a specific approach to address the issue.

- Achievement of objectives

  All the proposed objectives are properly met with the implementation of this research component. The main objective of developing assisted document capturing methods, document content segmentation and classification of the segmented parts and the interpretation of the identified charts within the document is completed and yielded great results on every developed component.

- Limitations and future work

  As with any study with a good research gap, this study has some limitations that can be overcome with future research. As mentioned in the research findings as well, the quality of the smartphone camera is the main limitation of this study. Since everything heavily depends on the quality of the camera, results with different mobile devices can vary. Furthermore, the author is working on improvements to support much more complex chart types like stacked bar charts, scatter plots and box plots. In some instances, the edge detection component which is used to detect documents mistakenly detects other objects with the same shape as documents. To improve this one can include an object detection model in the mobile application to identify only documents.

- End-user feedback and suggestions

  When testing the implementation with end-users there were a few suggestions that they would like to see implemented. Also, the current limitation of smartphone cameras was an issue for them as well. To overcome this, some of them suggested developing a separate device with a high-quality camera with better accessibility features. There were some suggestions to extend the implementation to support for diagrams as well. Overall, the users are satisfied with the implementation and were glad to assist with user acceptance testing.

4.4. Summary of student contribution

Student: Priyashan Sandunhetti S. H. S. (IT19187242)

Research component: Document Zone Segmentation, Classification and Chart Interpretation

Tasks:

- Developed the mobile application camera with real-time edge detection methods
- Implemented the Django backend to expose services like document segmentation, chart classification and interpretation to be used by the mobile application
- Implemented document segmentation algorithm and added it to the Django backend
- Implemented the CNN to classify chart images
- Collected images of charts and created a dataset to train the classification algorithm
- Trained the chart classification neural network and added it to the backend
- Implemented four different algorithms to extract data from chart images (Vertical bar charts, Horizontal bar charts, Pie charts and line charts) and to generate sentences using extracted data.

5. CONCLUSION

A huge portion of the global population is unable to comprehend printed documents properly due to a variety of problems. A print disability prohibits a person from acquiring information from printed material in a conventional manner, necessitating the adoption of alternative access techniques. These impairments can affect a person's daily life as well as their education and literacy. As a solution to this issue, we've developed a mobile solution that can scan and understand the content of printed documents using a variety of accessibility options to accommodate all possible users. Our solution is implemented with a voice assistant to help the user collect documents and navigate the program for greater accessibility. The user interfaces are created with appropriate fonts and buttons to facilitate the operation of the application by visually challenged users. This application focuses mostly on digitizing and reading aloud the captured image of printed documents or books. Text, mathematical expressions, table data, chart data, and image description are all supported by the provided solution.

In this particular research component, the main objectives are to assist the user with the capturing process, to segment the captured document image, to classify identified charts into chart types and to extract and explain the data within the charts. These objectives are implemented with various image processing techniques. Especially deep neural networks (Convolutional), Character Region Awareness, Optical Character Recognition, Object detection with deep neural networks and Open-CV library. The objectives were achieved within the given timeframe and the algorithms achieved great results compared to currently available applications and past studies. Also, the study covers many major research gaps in the assistive technology research field as well as chart data extraction and document image analysis research areas.

The tests for the component have been carried out with various users as well as unit and performance testing were done on various devices. There are some improvements made according to the end user feedback. There are some limitations with the implemented solution like smartphone camera quality and the support for complex charts like stacked bar charts, scatter plots and box plots which can be overcome with future research.

REFERENCES

[1] W.-J. Chang, L.-B. Chen, C.-H. Hsu, J.-H. Chen, T.-C. Yang, and C.-P. Lin, "MedGlasses: A wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients," IEEE Access, vol. 8, pp. 17013–17024, 2020.

[2] "Blindness and vision impairment," Who.int. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. [Accessed: 11-Feb-2022].

[3] Southern Cross University, "Students with a print disability - Southern Cross University," Edu.au. [Online]. Available: https://www.scu.edu.au/copyright/for-students/students-with-a-print-disability/. [Accessed: 11-Feb-2022].

[4] K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon, and V. Polezhai, "Braille character recognition based on neural networks," in 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 2018.

[5] A. Graves, "Braille literacy statistics research study: History and politics of the 'braille reader statistic': A summary of AFB leadership conference session on education," J. Vis. Impair. Blind, vol. 112, no. 3, pp. 328–331, 2018.

[6] N. D. U. Gamage, K. W. C. Jayadewa, and J. A. D. C. A. Jayakody, "Document reader for vision impaired elementary school children to identify printed images," in 2019 International Conference on Advancements in Computing (ICAC), 2019.

[7] S. Muralidharan, D. Venkatesh, J. Pritmen, R. Purushothaman, S. J. Anusuya, and V. Saravanaperumal, "Reading Aid for Visually Impaired People," International Journal of Advance Research, vol. 4, no. 2, 2018.

[8] R. Shilkrot, J. Huber, W. Meng Ee, P. Maes, and S. C. Nanayakkara, "FingerReader: A wearable device to explore printed text on the go," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15, 2015.

[9] T. A. Tran, K. Oh, I.-S. Na, G.-S. Lee, H.-J. Yang, and S.-H. Kim, “A robust system for document layout analysis using multilevel homogeneity structure,” Expert Syst. Appl., vol. 85, pp. 99–113, 2017.

[10] C. Jayant, M. Renzelmann, D. Wen, S. Krisnandi, R. Ladner, and D. Comden, “Automated tactile graphics translation: In the field,” in Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility - Assets ’07, 2007.

[11] R. Saha, A. Mondal, and C. V. Jawahar, “Graphical object detection in document images,” in 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 51–58.

[12] D. Augusto Borges Oliveira and M. Palhares Viana, “Fast CNN-Based Document Layout Analysis,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1173–1180.

[13] A. Corbelli, L. Baraldi, F. Balducci, C. Grana, and R. Cucchiara, “Layout analysis and content classification in digitized books,” in Communications in Computer and Information Science, Cham: Springer International Publishing, 2017, pp. 153–165.

[14] R. Saha, A. Mondal, and C. V. Jawahar, “Graphical object detection in document images,” in 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 51–58.

[15] D. Augusto Borges Oliveira and M. Palhares Viana, “Fast CNN-Based Document Layout Analysis,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1173–1180.

[16] A. Balaji, T. Ramanathan, and V. Sonathi, “Chart-Text: A fully automated chart image descriptor,” arXiv [cs.CV], 2018.

[17] W. Dai, M. Wang, Z. Niu, and J. Zhang, “Chart decoder: Generating textual and numeric information from chart images automatically,” J. Vis. Lang. Comput., vol. 48, pp. 101–109, 2018.

[18] J. Luo, Z. Li, J. Wang, and C.-Y. Lin, “ChartOCR: Data extraction from charts images via a deep hybrid framework,” in 2021 IEEE Winter Conference on Applications

of Computer Vision (WACV), 2021.

[19] "Blindness and vision impairment," Who.int. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. [Accessed: 31-Jan-2022].

[20] B. Okpeh, "Towards ending the information poverty among persons with print disabilities: A proposed draft (amendment) bill for disability diversification of the national library of Nigeria," SSRN Electron. J., 2021.

[21] "What is print disability," Visionaustralia.org. [Online]. Available: https://www.visionaustralia.org/services/print-accessibility/what-is-print-disability. [Accessed: 31-Jan-2022].

[22] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith, and N. Kumar, "Image Classification using SVM and CNN," in 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020.

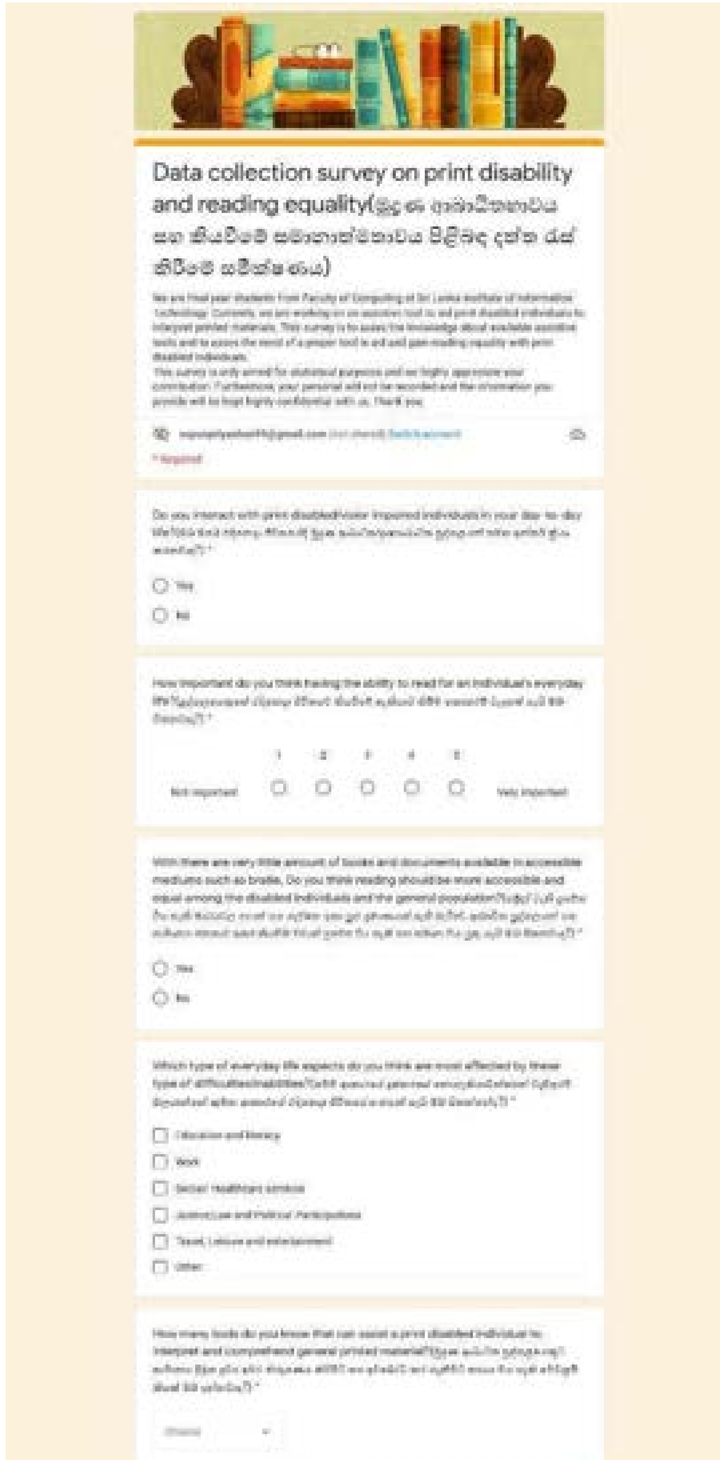
[23] Z. Shen, R. Zhang, M. Dell, B. C. G. Lee, J. Carlson, and W. Li, "LayoutParser: A unified toolkit for deep learning based document image analysis," 2021.

[24] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," arXiv [cs.CV], 2019.

[25] S. Saoji, R. Singh, A. Eqbal, και B. Vidyapeeth, 'TEXT RECOGINATION AND DETECTION FROM IMAGES USING PYTESSERACT', Journal of Interdisciplinary Cycle Research, τ. XIII, σσ. 1674–1679, 08 2021.

[26] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, 'Detectron2', 2019. [Digital edition]. Available at: https://github.com/facebookresearch/detectron2.

APPENDICES


Data collection survey on print disability and reading equality(මුද්‍රණ ආබාධිතභාවය සහ කියවීමේ සමානාත්මතාවය පිළිබඳ දත්ත රැස් කිරීමේ සමීක්ෂණය)

Yes

No

1 2 3 4 5

Not important Very important

Yes

No

Education and literacy

Work

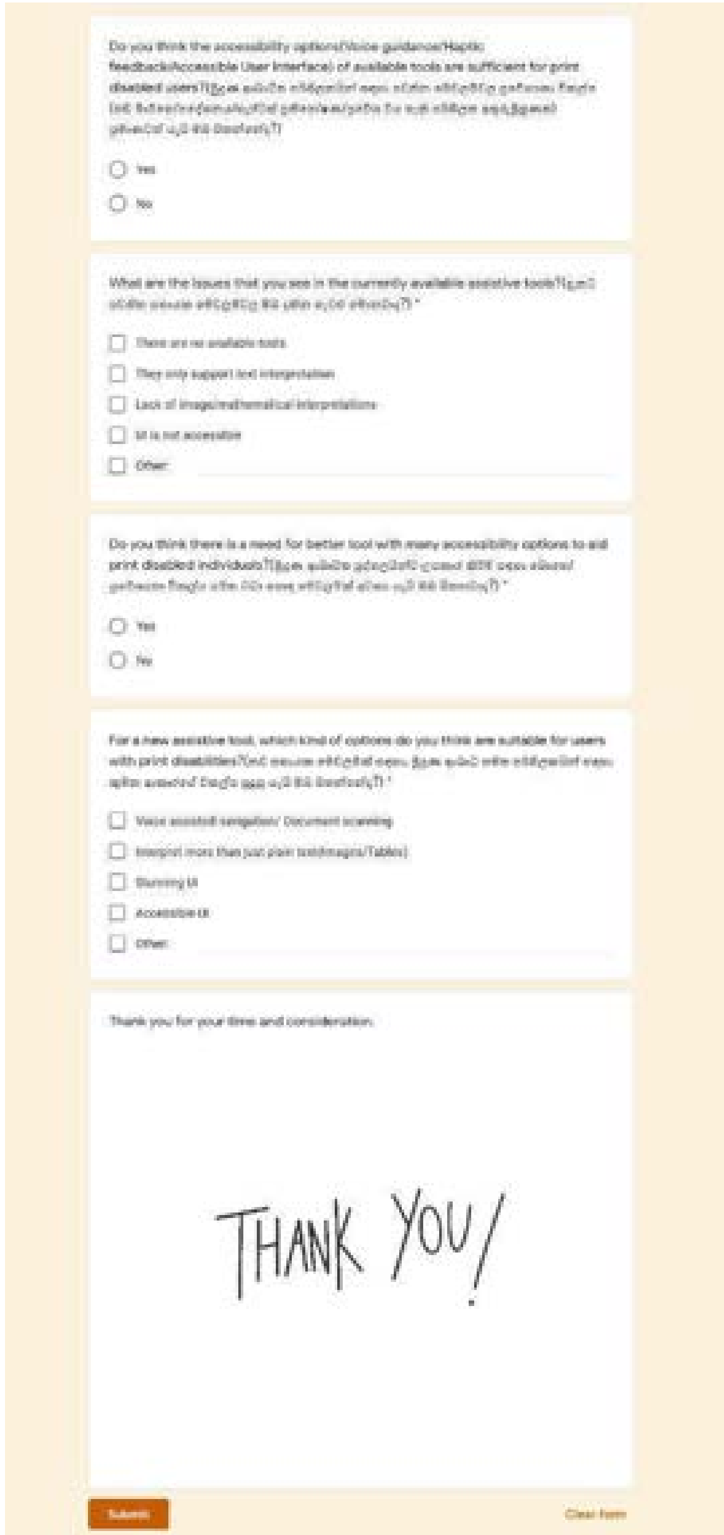
Social/ Healthcare services

Justice/Law and Political Participation

Travel, Leisure and entertainment

Other

Appendix A: Online Survey I



Do you think the accessibility options(Voice guidance/Haptic feedback/Accessible User Interface) of available tools are sufficient for print disabled users?

- Yes
- No

What are the issues that you see in the currently available assistive tools? *

- There are no available tools
- They only support text interpretation
- Lack of image/mathematical interpretations
- UI is not accessible
- Other:

Do you think there is a need for better tool with many accessibility options to aid print disabled individuals? *

- Yes
- No

For a new assistive tool, which kind of options do you think are suitable for users with print disabilities? *

- Voice assisted navigation/ Document scanning
- Interpret more than just plain text(Images/Tables)
- Stunning UI
- Accessible UI
- Other:

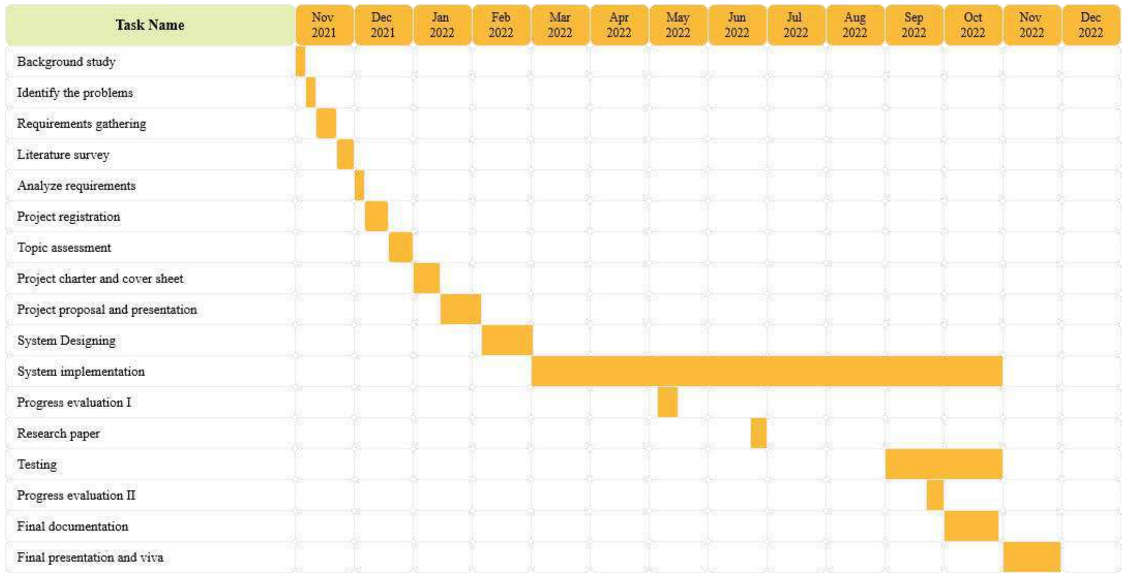
Thank you for your time and consideration.

THANK YOU!

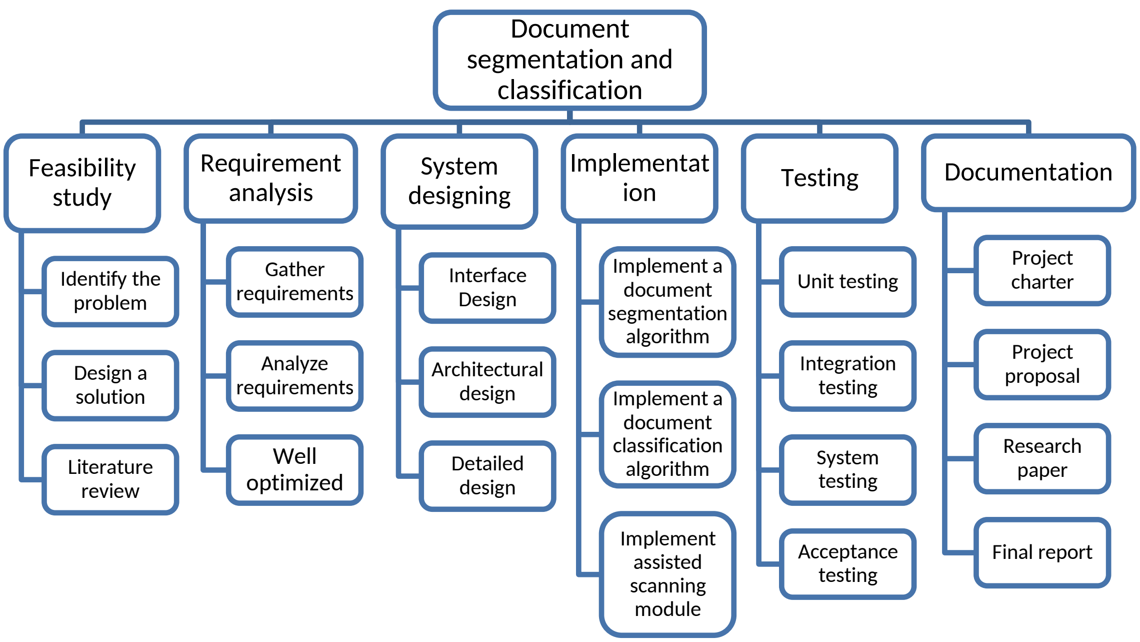
Submit

Clear form

Appendix B: Online Survey II



Appendix C: Gantt chart



Appendix D: Work breakdown structure

Thesis 19187242

ORIGINALITY REPORT

5%	5%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sri Lanka Institute of Information Technology Student Paper	2%
2	www.coursehero.com Internet Source	<1%
3	printdisability.org Internet Source	<1%
4	eprints.utm.my Internet Source	<1%
5	dl.lib.mrt.ac.lk Internet Source	<1%
6	open.uct.ac.za Internet Source	<1%
7	jsrad.org Internet Source	<1%
8	Submitted to University of Central Lancashire Student Paper	<1%

Appendix E: Plagiarism report