EXPLORE INNOVATE EXCEL

DRISHTI

Drishti

Major Project Report

Submitted By:

Neetish Kumar	(18103009)
Ankit Goyal	(18103018)
Bishwash Pokhrel	(18103106)
Jaspreet Singh	(18103107)

Under the Supervision of:

Prof. Poonam Saini

Assistant Professor

Department of Computer Science and Engineering

Punjab Engineering College

(Deemed to be University)

Chandigarh

EXPLORE INNOVATE EXCEL

DECLARATION

We hereby declare that the project work entitled “Drishti” is an authentic record of our own work carried out at Punjab Engineering College (Deemed to be University), as a requirement of Major Project for the award of degree of B. Tech (Computer Science and Engineering), under the guidance of Prof. Poonam Saini (Faculty, Department of Computer Science and Engineering) during August 2021 to May 2022.

Neetish Kumar, 18103009

Ankit Goyal, 18103018

Bishwash Pokhrel, 18103106

Jaspreet Singh, 18103107

CERTIFICATE

This is to certify that the project entitled Drishti by Ankit Goyal, Neetish Kumar, Bishwash Pokhrel & Jaspreet Singh is an authentic record of our work carried out under the supervision of Prof. Poonam Saini, Faculty, Computer Science and Engineering Department, Punjab Engineering College (Deemed to be University), Chandigarh in fulfilment of the requirements as a part of Major Project for the award of 06 credits in semester 8 of the degree of Bachelor of Technology in Computer Science and Engineering.

Certified that the above statement made by the students is correct to the best of my knowledge and belief.

Prof. Poonam Saini
(Faculty Mentor)
Department of Computer Science and Engineering
Punjab Engineering College
(Deemed to be University)
Chandigarh

Ankit Goyal
18103018

Neetish Kumar
18103009

Bishwash Pokhrel
18103106

Jaspreet Singh
18103107

Dated:

ACKNOWLEDGEMENT

We have taken a lot of deliberations in this venture. But it wouldn't have been possible without the help and backing of numerous people. We want to extend our true appreciation and thank them. We take this opportunity to express our profound gratitude and deep regards to our mentor Prof. Poonam Saini for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

This project truly wouldn’t have been possible without her mentorship.

ABSTRACT

The challenges faced by blind people and visually impaired in their everyday lives are not well understood. They confront several visual challenges every day – from reading the denomination on a currency note to reading books for education or any other content in their day-to-day life. There is very limited braille printed books in public libraries and in schools/colleges, which poses a great issue as it puts a constraint on them being educated. Further, the lack of support for them, the limited accessibility to activities and information, the societal stigma and the lack of unemployment, are all factors frequently leading blind or low vision individuals in isolation. We aim to make this world a better place to live in for the blind and the visually impaired. Our project aims to help a visually impaired or a blind person in their everyday life tasks such as Recognizing Currency Notes, reading text from live images, and sharing their live locations with their loved ones.

Figure 1: Visually Impaired People

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Our project aims to help a visually impaired or a blind person in their everyday life tasks such as Recognizing Currency Notes, reading text from live images, and sharing their live locations with their loved ones. Problems faced by visually impaired in performing daily activities are in great number. They are unable to recognize the paper currencies due to similarity of paper texture and size between different categories. Navigation and text reading (book reading, newspaper reading, etc.) are some of the other major problems faced by them. Therefore, a system could be designed that could take in live video input from the person and conveys the corresponding audio output for the visually impaired, which will really help them in their day-to-day chores.

1.1.1 Mobile Application

A mobile application, most referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. The advantage of such an application is that it can be accessed from anywhere and is an anytime application- i.e., it is always available, regardless of the location of the user or other factors. There are many frameworks, languages, tools and paths to create a mobile application. Drishti uses React Native with Expo and JavaScript for the front-end and Flask, a Python framework, for the server-side scripting needs. React Native lets us build both Android as well as IOS apps simultaneously, which basically converts the React Native code to their respective native code.

1.1.2 What is Drishti

We aim to help a visually impaired person by helping them in their everyday activities such as recognizing currency notes, informing their loved ones about their live location through email or SMS and reading out text from a video input, which can be of a book, newspaper or text written on a product.

Thus, we've developed these features by applying Deep learning on it. The tools used for developing the project are Python, React Native with Expo, JavaScript and Flask.

Figure 2: Technology helping visually impaired in studying

- A mobile application incorporating deep learning techniques.
- Mobile App features:
    - Currency Note Recognition
    - Text to Speech:
        - Documents – E.g., Books, Question Papers, etc.
        - Non-Documents – E.g., Grocery Items, Household products, etc.
    - Live location sharing with their loved ones through email and SMS.
- Output: Voice output for guiding the users.

- Easy to use mobile app for the users.
- An additional feature to simulate a wearable device through webcam in which all the mentioned features are incorporated for more convenience and efficiency.

Figure 3: Drishti App

CHAPTER 2

BACKGROUND

2.1 Problem Statement Formulation

Blindness does come with its share of challenges, though, considering most of the world is designed with sighted people in mind. As a result, living with limited vision means blind people must find other ways to handle some things most people take for granted. They confront several visual challenges every day – from reading the denomination on a currency note to reading books for education or any other content in their day-to-day life. There is very limited braille printed books in public libraries and in schools/colleges, which poses a great issue as it puts a constraint on them being educated. Even the ones close to the visually impaired are in constant state of worry for their well-being.

In such a situation, an app which could assist the blind or the visually impaired person would be of great help. There's a need for a system that could take in live video input from the person and conveys the corresponding audio output for the visually impaired, which will really help them in their day-to-day chores.

- **What already exists:**
    - Currency Note Recognition system (for USD only)
    - PDF text to speech
    - Voice assistants for answering questions
- **Existing Problems:**
    - No Currency Note Recognition system (for INR)
    - No text to speech specially designed for blind
    - No application designed to read day to day products for visually impaired
- **Our Contribution:**

- Currency Recognition:
  - Created dataset for Currency Note Recognition for Indian Rupee (INR).
  - Trained Currency note recognition model using Convolutional Neural Networks.
  - Real time predictions giving currency denominations as output.
- Optical Character Recognition:
  - Reading document (dense, well formatted text with consistent background)
  - Reading non-document (sparse, fancy text in no particular order)
  - Created individual pipelines for both the use cases and applied different models for both the use cases
- Live location sharing:
  - A feature that allows the loved ones to consistently monitor the live location of the visually impaired person.

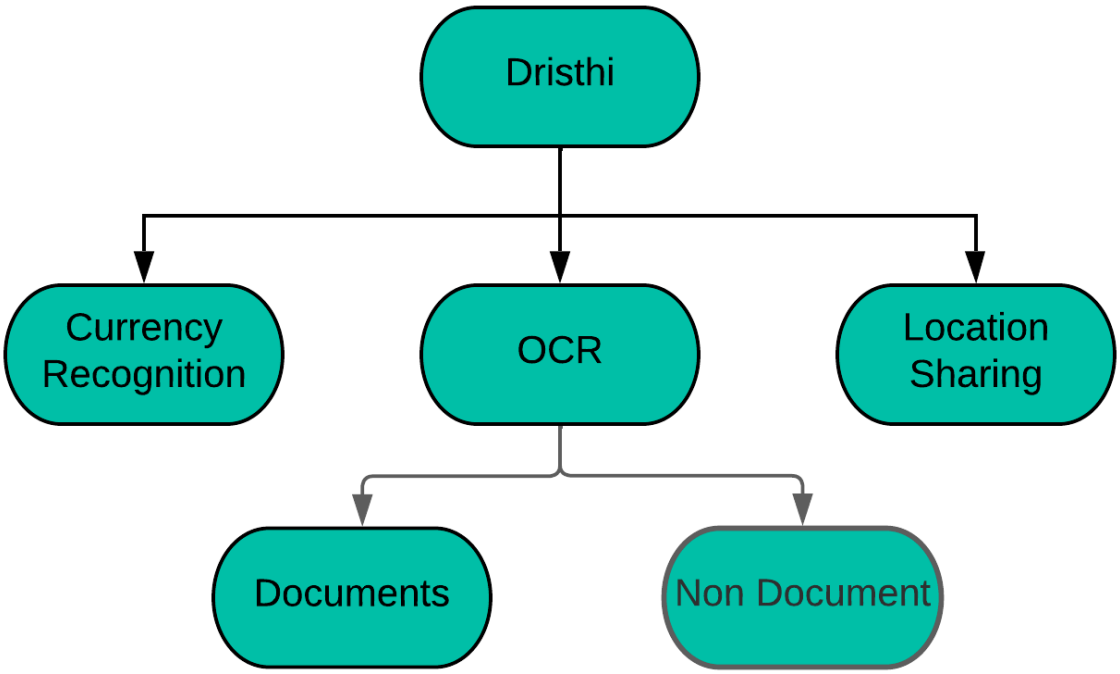


Figure 4: Drishti Features

CHAPTER 3

PROPOSED WORK

(ALGORITHM/MODEL/APPROACH)

3.1 Currency Recognition

Currency Recognition is a feature which takes a few seconds of video as input and detects the denomination of currency in the image with the help of their distinctive patterns through deep learning.

3.1.1 Dataset Preprocessing & Analysis

Indian Currency dataset by Mendeley Data contributed 4000 images, consisting of old ten, twenty, fifty, and hundred rupees notes. Indian Currency Note images dataset 2020 was downloaded from Kaggle and had over 2500 images, consisting of new currency denomination notes of 10, 20, 50, 100, 200, 500, and 2000. Also, it has one additional class of images labelled background to correctly detect images without any currency note. Similarly, India Currency notes contributed over 1000 images with a similar classification structure as the Indian Currency Note images dataset 2020.

The majority of the real-world datasets are highly susceptible to missing, inconsistent, and noisy data due to their heterogeneous origin. The input requirements for a model are specific to its design, thus the datasets require some pre-processing to move to the next phases. Dataset Pre-processing is, therefore, important to improve the overall data quality.

For transforming the dataset into the final state for training the deep learning model, the following four steps are followed:

1. Filtering out irrelevant images from the dataset, for e.g. The old Indian notes with denomination values of 500 and 1000.
2. Correcting the wrong labels where currency denomination notes are not correctly labelled.
3. Merging datasets into one final dataset.
4. Splitting the dataset into training, validation and testing.

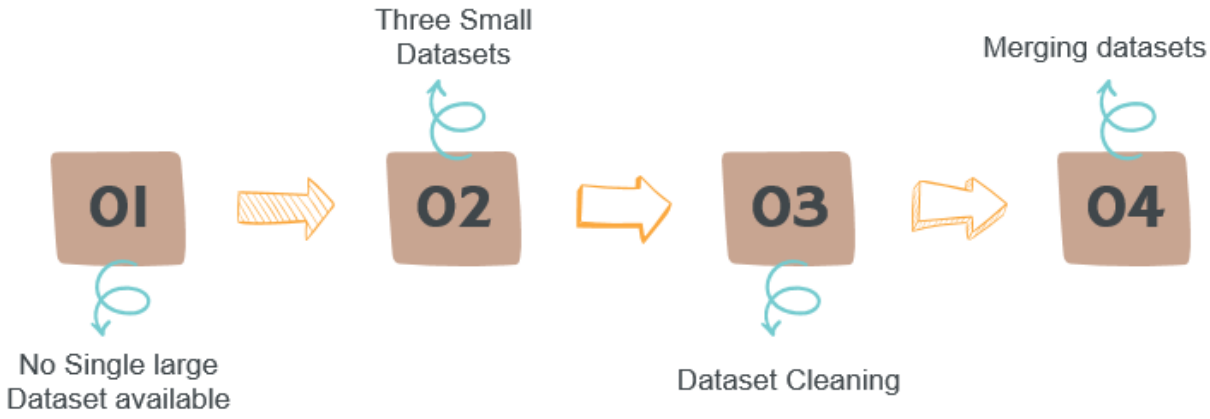


Figure 5: Dataset Preprocessing

Figure 6: Number of Images in Datasets

Datasets had the following distribution of classes of currency denominations:

- 10
- 20
- 50
- 100
- 200
- 500
- 2000
- Background

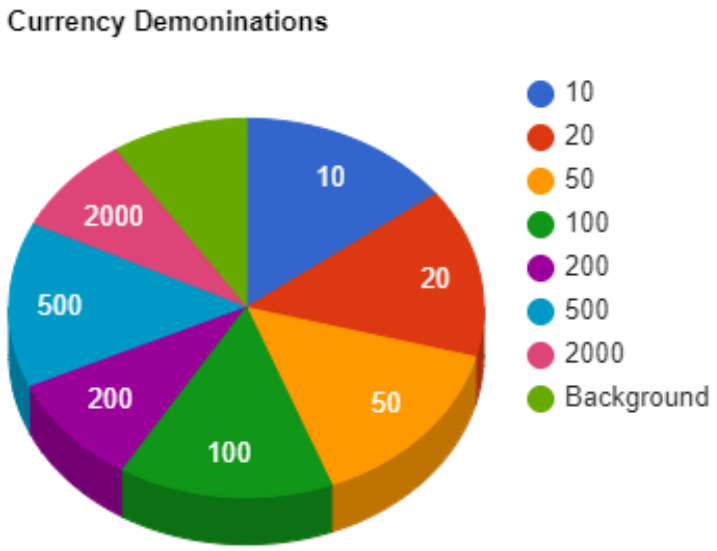


Figure 7: Distribution of number of images for each class

Most of the earlier works have not taken the orientation of the notes into consideration. To accommodate for different orientations, we used Image augmentation. Applying Image augmentation is a really important part of currency recognition since a visually impaired person might hold a currency note in any orientation. So before starting the training phase, we applied the following augmentations to our images while reading the images from the directory:

- Flipping images
- Rotations
- Shifts
- Shear
- Zoom

3.1.2 Model

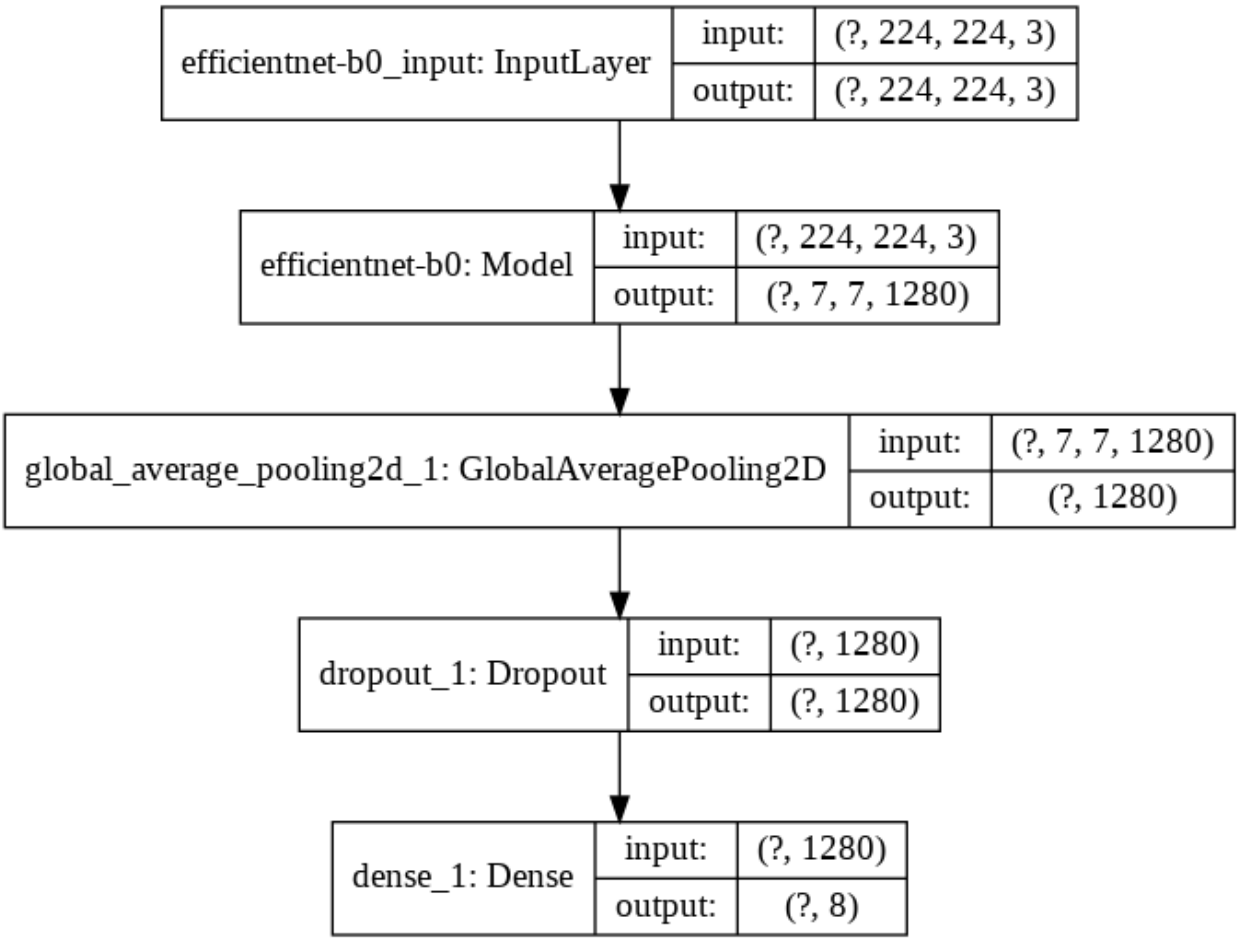


Figure 8: Neural Network Architecture

We have used Convolutional Neural Networks (CNNs) with transfer learning. We performed the following steps:

- Selected a pre-trained base model namely EfficientNet, which is trained over millions of images of the ImageNet dataset.
- We removed the last few layers of the architecture.
- Appended our custom fully connected layers to it.

The following is the structure of our trained neural network:

1. We have used a generic Convolutional Neural network with transfer learning with the following steps:

- Base EfficientNet Input Layer
- EfficientNet Model architecture

2. Along with the baseline structure, the following fully connected layers have been customized:
    - GlobalAveragePooling2D layer -
        - Global Average Pooling layer is often used to replace the fully-connected or densely connected layers in a classifier.
        - The model ends with a convolutional layer that generates as many feature maps as the number of target classes and applies global average pooling to each in order to convert each feature map into one value.
        - As feature maps can recognize certain elements within the input data, the maps in the final layer effectively learn to "recognize" the presence of a particular class in this architecture.
        - Moreover, it reduces overfitting because of the fact that there is no parameter to be learned in the global average pooling layer.
    - Dropout layer -
        - Dropout prevents overfitting and regularizes by randomly cutting the connections (also known as dropping the connection) between neurons in successive layers during training.
    - SoftMax output layer -
        - The SoftMax activation works in a multiclass classification problem.
        - Currency recognition system aims at predicting 8 classes.

3.1.3 Algorithm and Approach

The following steps were followed:

1. Dataset Collection
2. Dataset Pre-processing:
    - Removal of invalid images
    - Image labels corrections
3. Dataset merging

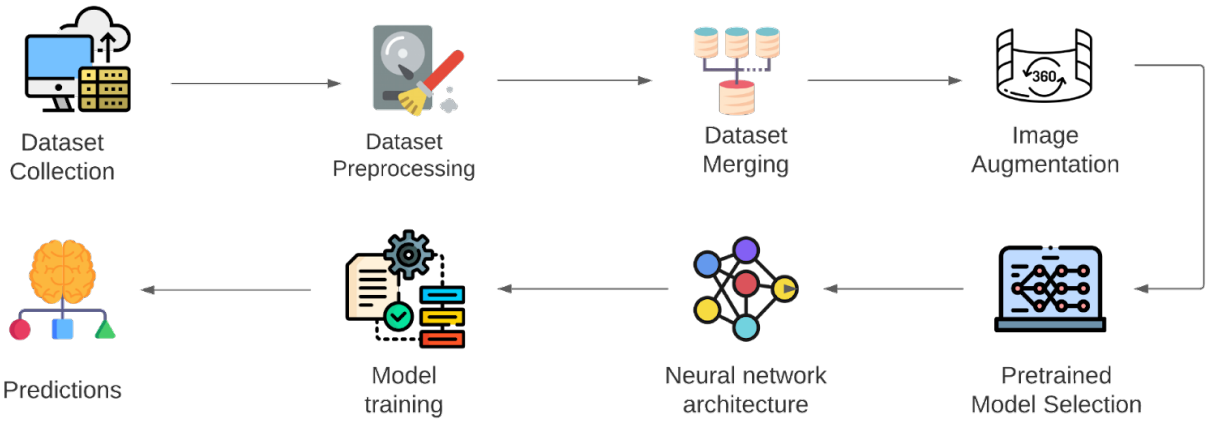


Figure 9: Proposed Pipeline

4. Image augmentation
5. Pretrained model selection
    - For this project, many deep learning models were used as the base model and were trained over the dataset for the currency recognition application.
    - The proposed models for the project are InceptionV3, VGG16, and EfficientNet, all of them trained over the ImageNet dataset.
6. Final Neural Network architecture design
    - Created a suitable Neural Network architecture by appending the last few layers to the base pre-trained model.
    - The layers added were GlobalAveragePooling2D layer, Dropout layer and SoftMax layer.
7. Model training, by tweaking the following hyperparameters:
    - Learning rate: Learning rate is a hyper-parameter that controls the weights of our neural network with respect to the loss gradient. It defines how quickly the neural network updates the concepts it has learned.
    - Optimizer: An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rate. Thus, it helps in reducing the overall loss and improving the accuracy.
8. Predictions

3.2 Optical Character Recognition

Optical character recognition (OCR) technology is a feature for automating data extraction from printed or written text from a scanned document or image file and then converting the text into a machine-readable form. Optical character recognition (OCR) is sometimes referred to as text recognition. An OCR program extracts and repurposes data from scanned documents, camera images and image-only pdfs. OCR software singles out letters on the image, puts them into words and then puts the words into sentences, thus enabling access to original content, which can be easy converted to speech.

We have classified OCR feature into two sub-classification: OCR for Document & OCR for Non-Document.

3.2.1 OCR for Document

We have classified any image with dense text, in a well formatted manner for e.g. horizontally aligned with consistent background is classified as document.

For e.g. Novels, newspapers, books, question paper or any other document.

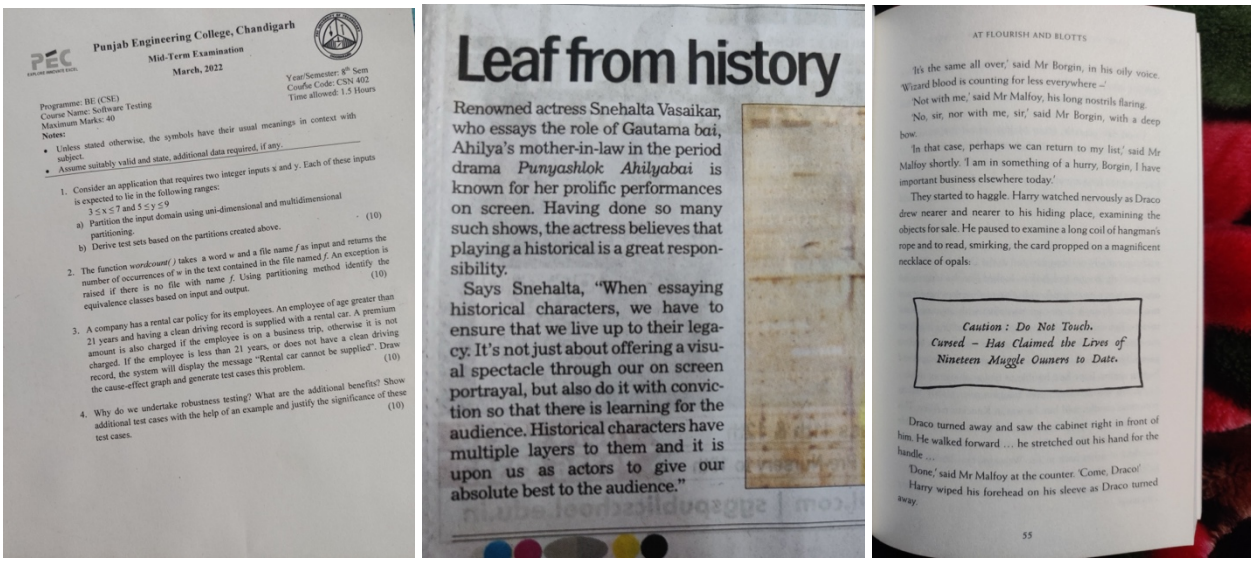


Figure 10: Examples of Document

3.2.1.1 Dataset Preprocessing & Analysis

Initially we wanted to leverage a coco-text dataset with 83k images but we found them to have inconsistent mapping between the bounding box and the image itself. Hence we used the ICDAR dataset with 1k train images and 500 test images. All the 1000 train images had corresponding mappings denoting bounding box i.e. 4 coordinates of a rectangle each in x-y plane starting from top left corner in the clockwise direction.

3.2.1.2 Model

1. Model training

Because of the less amount of dataset, we had to use transfer learning. In order to do so we selected two major models, VGG-16 and ResNet-50. After adding a few layers to customize to our needs, we started training on the available dataset.

2. Loss function

Accuracy cannot be the right metrics to benchmark the performance of the model because our problem statement is not a classification problem and we don't have a discrete output that defines the correctness of the prediction.

Loss function approach:

We decided on maximizing the overlap between the original bounding box and the bounding box created by the trained model. In simple terms our loss function evaluates the area of intersection between original bounding box and that generated by the model and aims to maximize it.
The following is the loss function used to calculate the loss function:

$$L_g = L_{\text{AABB}} + \lambda_\theta L_\theta$$

The following value evaluates the amount of overlap between the original bounding box and the predicted bounded box, and less the overlap, more the loss function. Here IoU stands for Intersection over Union.

$$L_{\text{AABB}} = -\log \text{IoU}(\hat{\mathbf{R}}, \mathbf{R}^*) = -\log \frac{|\hat{\mathbf{R}} \cap \mathbf{R}^*|}{|\hat{\mathbf{R}} \cup \mathbf{R}^*|}$$

The following value considers the angle of alignment of the original bounding box and the predicted bounded box to accommodate for slanted texts.

$$L_\theta(\hat{\theta}, \theta^*) = 1 - \cos(\hat{\theta} - \theta^*)$$

3. **Merging Boxes**

The model creates multiple bounding box around a word. In order to declutter the image and also isolate the words simultaneously, we merge these multiple boxes and create a single bounding around a word. In order to do, the model follows following algorithm.

- First, sort the geometries and start from the topmost.
- Take the next box in the row and find the IOU with the previous
- If IOU > threshold, merge the 2 boxes by taking the weighted average by score otherwise keep it as it is.
- Repeat steps 2 to 3 until all boxes are iterated.

3.2.1.3 Algorithm and Approach

Several problems were encountered while creating OCR for document. In order to overcome these issues we created a pipeline with multiple steps of processing.
The following pipeline was formulated for OCR in document:

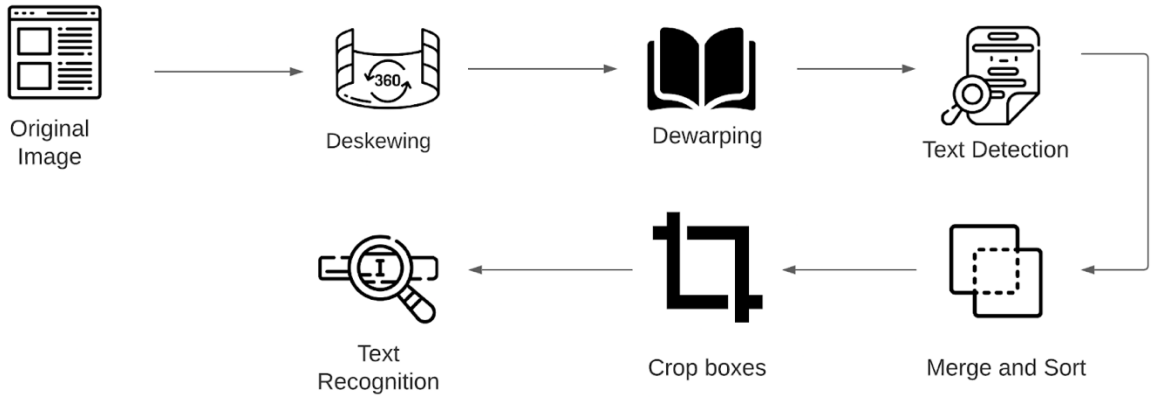
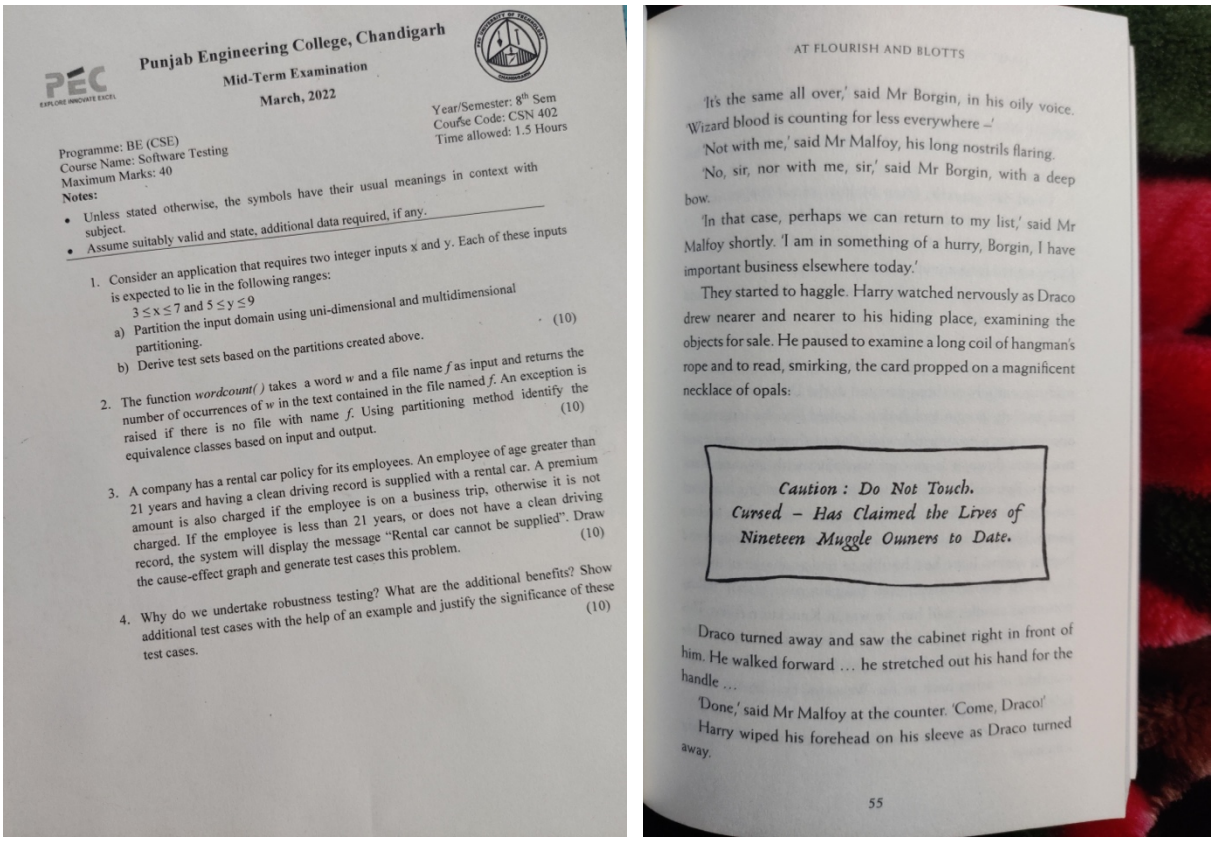


Figure 11: OCR Document Pipeline

We take two images as our running example and apply all the different steps and see their respective outcomes. The images are specifically selected to entertain different corner cases, one being skewed question paper of PEC, and the other being a warped image of Harry Porter and the Chamber of Secrets.



Punjab Engineering College, Chandigarh

Mid-Term Examination

March, 2022

Programme: BE (CSE)
Course Name: Software Testing
Maximum Marks: 40
Year/Semester: 8th Sem
Course Code: CSN 402
Time allowed: 1.5 Hours

Notes:

- Unless stated otherwise, the symbols have their usual meanings in context with subject.
- Assume suitably valid and state, additional data required, if any.

1. Consider an application that requires two integer inputs x and y. Each of these inputs is expected to lie in the following ranges:
   $3 \le x \le 7$ and $5 \le y \le 9$
   a) Partition the input domain using uni-dimensional and multidimensional partitioning.
   b) Derive test sets based on the partitions created above. (10)
2. The function *wordcount( )* takes a word *w* and a file name *f* as input and returns the number of occurrences of *w* in the text contained in the file named *f*. An exception is raised if there is no file with name *f*. Using partitioning method identify the equivalence classes based on input and output. (10)
3. A company has a rental car policy for its employees. An employee of age greater than 21 years and having a clean driving record is supplied with a rental car. A premium amount is also charged if the employee is on a business trip, otherwise it is not charged. If the employee is less than 21 years, or does not have a clean driving record, the system will display the message "Rental car cannot be supplied". Draw the cause-effect graph and generate test cases this problem. (10)
4. Why do we undertake robustness testing? What are the additional benefits? Show additional test cases with the help of an example and justify the significance of these test cases. (10)

AT FLOURISH AND BLOTTS

'It's the same all over,' said Mr Borgin, in his oily voice. 'Wizard blood is counting for less everywhere –'

'Not with me,' said Mr Malfoy, his long nostrils flaring.

'No, sir, nor with me, sir,' said Mr Borgin, with a deep bow.

'In that case, perhaps we can return to my list,' said Mr Malfoy shortly. 'I am in something of a hurry, Borgin, I have important business elsewhere today.'

They started to haggle. Harry watched nervously as Draco drew nearer and nearer to his hiding place, examining the objects for sale. He paused to examine a long coil of hangman's rope and to read, smirking, the card propped on a magnificent necklace of opals:

Caution : Do Not Touch.
Cursed – Has Claimed the Lives of
Nineteen Muggle Owners to Date.

Draco turned away and saw the cabinet right in front of him. He walked forward … he stretched out his hand for the handle …

'Done,' said Mr Malfoy at the counter. 'Come, Draco!'

Harry wiped his forehead on his sleeve as Draco turned away.

55

Figure 12: Sample images for OCR Document

1. Deskewing:

Scanned documents often become skewed (slanted) during scanning because of misfeeds or other alignment errors. Skew is the amount of rotation necessary to return an image to horizontal and vertical alignment. Skew is measured in degrees. Deskewing is a process whereby skew is removed by rotating an image by the same amount as its skew but in the opposite direction. This results in a horizontally and vertically aligned image where the text runs across the page rather than at an angle.

When an image is not aligned correctly, optical character recognition (OCR) is more difficult and becomes slower and less accurate. Deskewing the documents beforehand can make the OCR process faster and more accurate.

The following algorithm was used in order to deskew an image:

- Detect edges using canny edge detector.
- Create hough line for the most prominent edges.

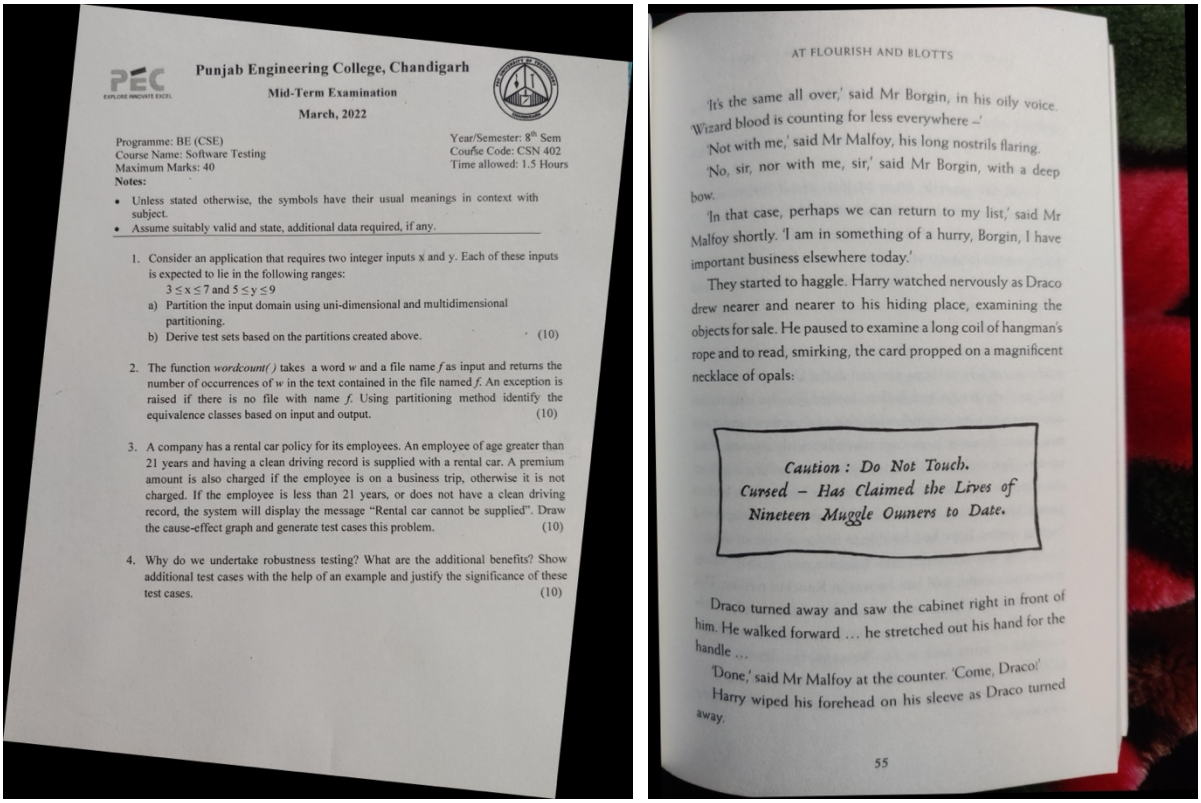


Figure 13: Images after Deskewing

- From the hough line after some accomodation, the average angle gives us the rotation angle.
- By reverse rotating the entire page by the same angle we are essentially aligning hough lines such that they become parallel to the x axis and this the text will be deskewed.
- Since there would be very less amount of text for non doc, the scope of finding average hough line would fail and hence we cannot use the same thing for non doc.

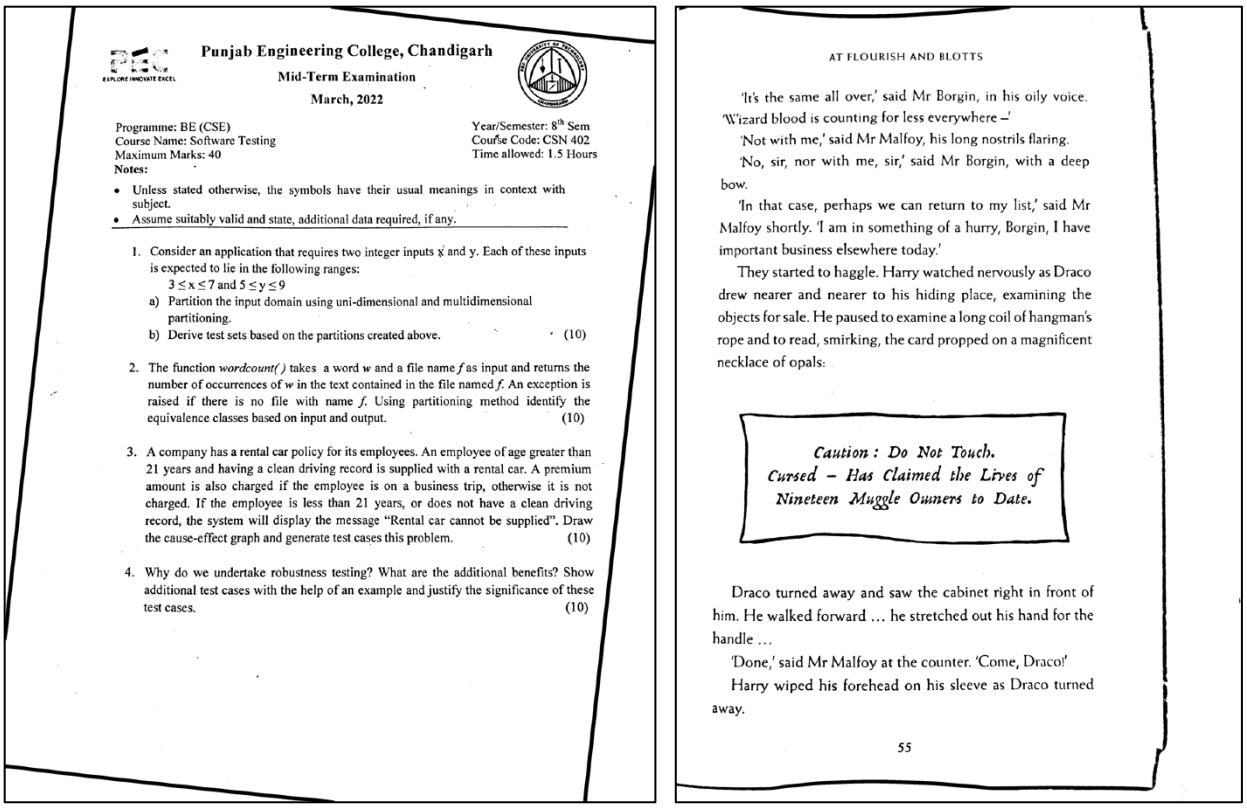
2. Dewarping:

Image warping is a common problem when one scans or photocopies a document page from a thick bound volume, resulting in shading and curved text lines in the spine area of the bound volume. The camera-captured digital documents may be often distorted and warped due to various document surfaces or camera angles.

This will not only impair readability but will also reduce the OCR accuracy. Also, the OCR systems find difficulty in reading such distorted images.

The following algorithm was used in order to dewarp an image:

- Findcontours: Look for regions that look “text-like”, joining all the continuous points (along the boundary), having same colour or intensity.
- Dilation and erosion to connect up horizontally adjacent pixels
- Make points on the horizontal continuous spans
- Then join them creating an MST, thus connecting all the nearly horizontal spans into a single curve line.
- Dense mesh of 3D page points given to cv2.projectPoints and which cv2.remaps it to straight line segments, hence providing dewarping.



Punjab Engineering College, Chandigarh

Mid-Term Examination

March, 2022

Programme: BE (CSE) Year/Semester: 8th Sem
Course Name: Software Testing Course Code: CSN 402
Maximum Marks: 40 Time allowed: 1.5 Hours
Notes:

- Unless stated otherwise, the symbols have their usual meanings in context with subject.
- Assume suitably valid and state, additional data required, if any.

1. Consider an application that requires two integer inputs x and y. Each of these inputs is expected to lie in the following ranges:
   $3 \le x \le 7$ and $5 \le y \le 9$
   a) Partition the input domain using uni-dimensional and multidimensional partitioning.
   b) Derive test sets based on the partitions created above. (10)
2. The function *wordcount()* takes a word *w* and a file name *f* as input and returns the number of occurrences of *w* in the text contained in the file named *f*. An exception is raised if there is no file with name *f*. Using partitioning method identify the equivalence classes based on input and output. (10)
3. A company has a rental car policy for its employees. An employee of age greater than 21 years and having a clean driving record is supplied with a rental car. A premium amount is also charged if the employee is on a business trip, otherwise it is not charged. If the employee is less than 21 years, or does not have a clean driving record, the system will display the message "Rental car cannot be supplied". Draw the cause-effect graph and generate test cases this problem. (10)
4. Why do we undertake robustness testing? What are the additional benefits? Show additional test cases with the help of an example and justify the significance of these test cases. (10)

AT FLOURISH AND BLOTTS

'It's the same all over,' said Mr Borgin, in his oily voice. 'Wizard blood is counting for less everywhere –'

'Not with me,' said Mr Malfoy, his long nostrils flaring.

'No, sir, nor with me, sir,' said Mr Borgin, with a deep bow.

'In that case, perhaps we can return to my list,' said Mr Malfoy shortly. 'I am in something of a hurry, Borgin, I have important business elsewhere today.'

They started to haggle. Harry watched nervously as Draco drew nearer and nearer to his hiding place, examining the objects for sale. He paused to examine a long coil of hangman's rope and to read, smirking, the card propped on a magnificent necklace of opals:

Caution: Do Not Touch.
Cursed – Has Claimed the Lives of
Nineteen Muggle Owners to Date.

Draco turned away and saw the cabinet right in front of him. He walked forward … he stretched out his hand for the handle …

'Done,' said Mr Malfoy at the counter. 'Come, Draco!'

Harry wiped his forehead on his sleeve as Draco turned away.

55

Figure 14: Images after Dewarping

3. Text Detection:

The model identifies words and creates a bounding box around each words. As explained in the section 3.2.1.2, the model identifies and isolates each word and create a bounding box around it.

We can directly perform text recognition after this step. But this poses a huge problem in comprehension of text, since our aim is to not just identify text, but to also lay it out in a meaningly manner in the form of audio output, which brings us to next step.

So, once the text bounding boxes have been identified, we need to combine all the boxes corresponding to a single horizontal span on the page.

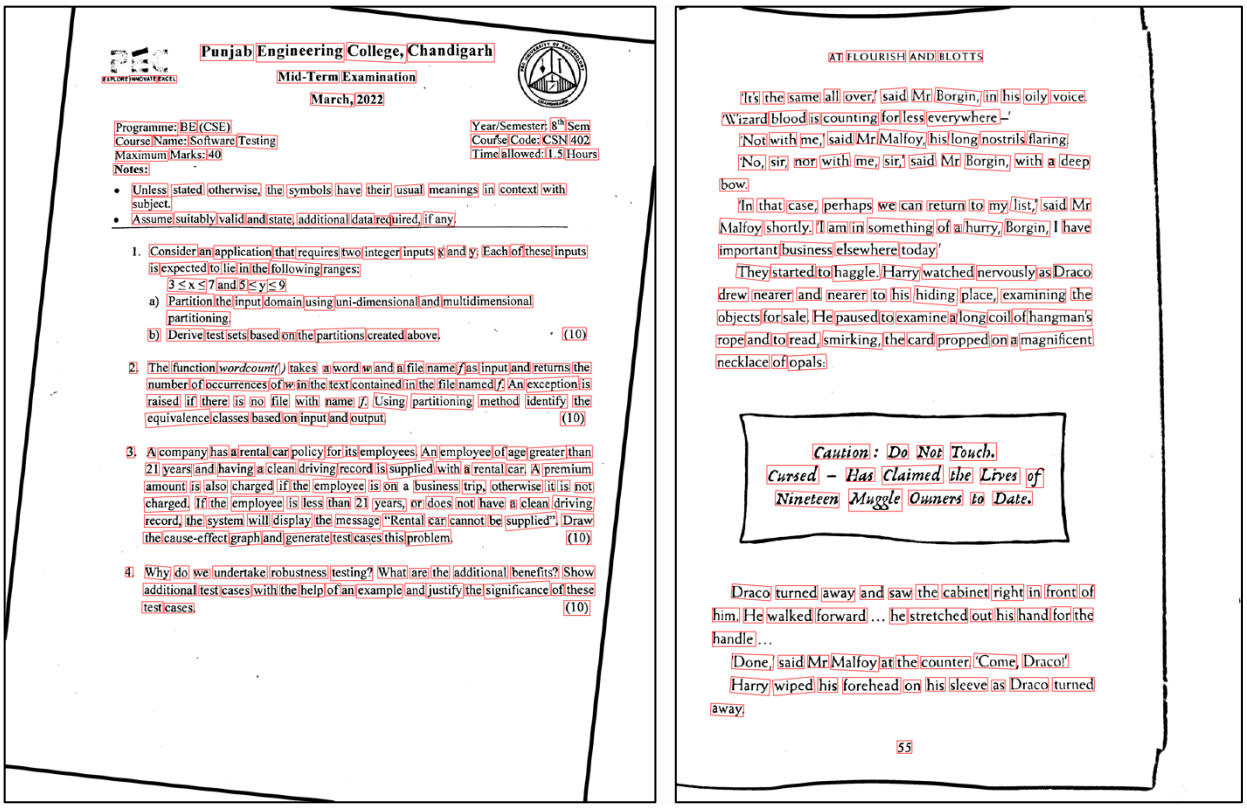


Figure 15: Images after Text Detection

4. Merge and Sort:

Our aim is to create a meaningful audio output and hence preserving the original arrangement of the words is crucial to make the output comprehensible. Since the bounding box are best optimized to highlight words, the boxes generated aren't horizontally aligned and hence sorting the words in proper order will be an issue.

In order to combat this problem we decided to merge the boxes such that the bounding box for words in same line would be merged and hence can be read line by line rather than word by word. Since entire line is read at once this eliminates the problem of sorting words.

Merging and sorting are done in two folds. Firstly, we distinguish which of these boxes fall into the same lines and cluster them together. For this we first sort by midpoint of *y_max* and *y_min* and then we used a condition that checks how far the current centroid is from the running average centroid (centroid of all the previously included boxes), and if it is below a certain threshold we assume that the boxes are in same line.

After this we check through the boxes in the same line which are sorted according to their *x_min* and merge boxes based upon the distance between them (the distance can be calculated by *x_min_curr-x_max_prev*) which are compared with the box height. If it's below a certain threshold, we can safely merge the boxes which would already be sorted according to their *x_min*.

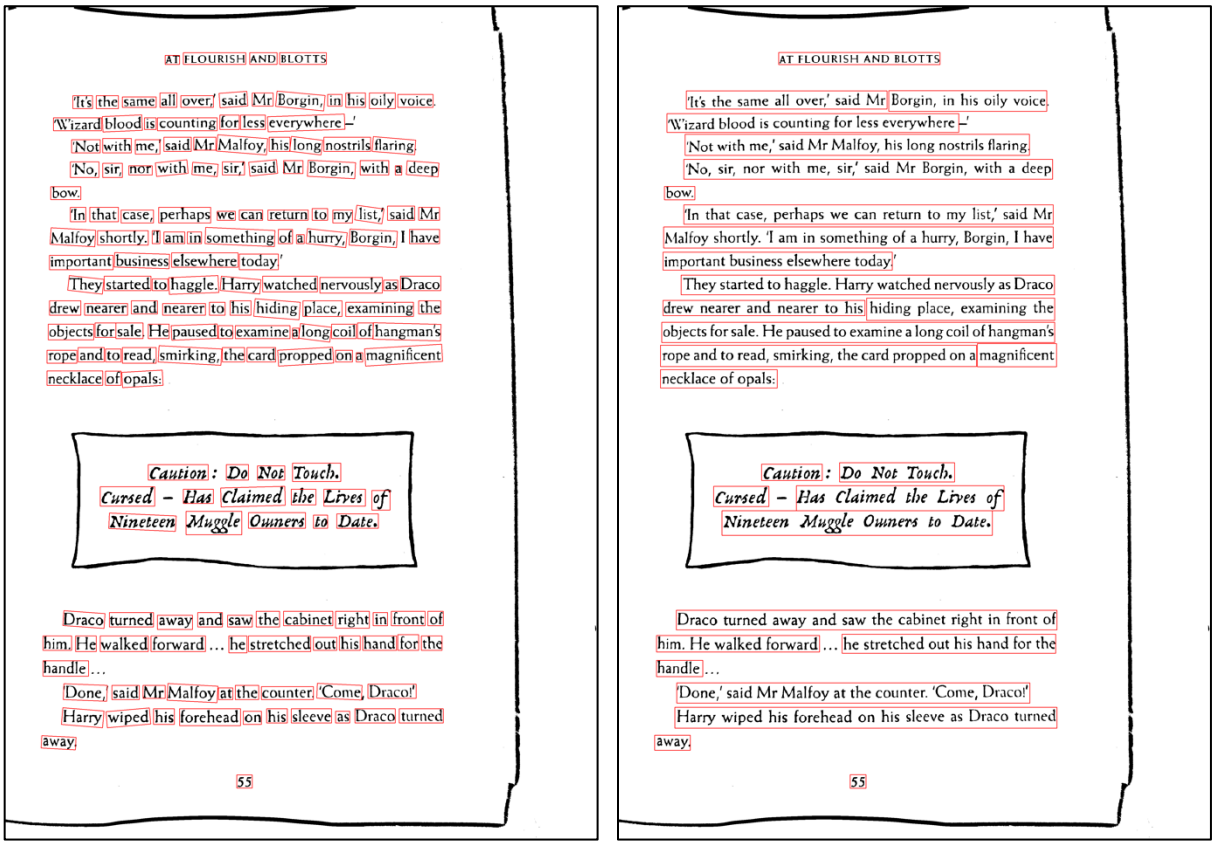


Figure 16: Images after Merging and Sorting

5. Crop Boxes:

The text detection model gives us the coordinates of the bounding box around the text. Using these coordinates we crop the image so that we can isolate the bounding box and pass on individual words to the model, which provides more efficient model performance. Some of the cropped snapshots are shown in the image below.

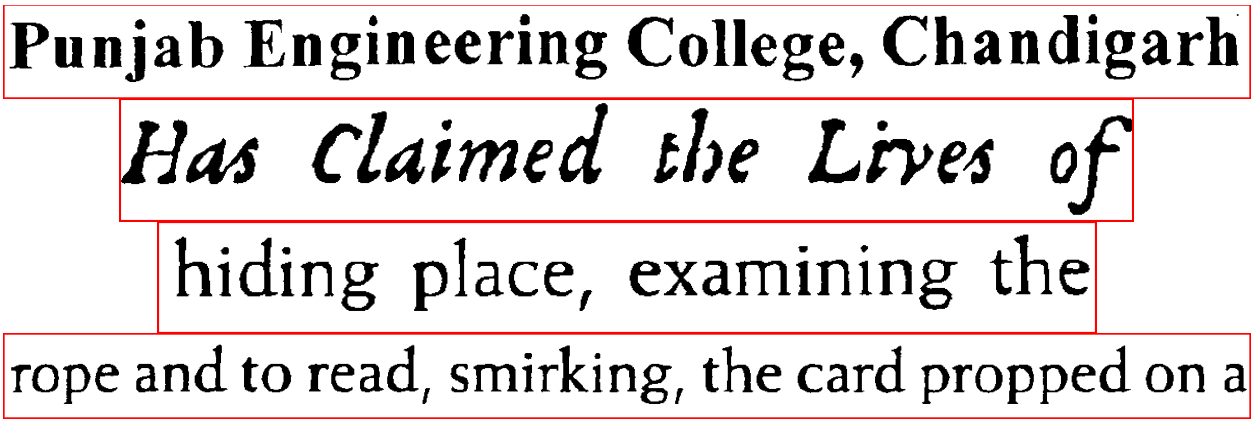


Figure 17: Some sample cropped Images

6. **Text Recognition:**

We used tesseract to recognize the provided images, however tesseract in its default setting didn't perform very well for our use case. Hence in order to improve the performance we used several pre-processing methods. The methods were selected on an iterative basis depending upon its impact on improving the detection accuracy.

- Detail enhancement to make sharper images.
- Converting RGB array to black and white.
- Median blur for noise removal.
- Thresholding is a very popular segmentation technique, used for separating an object considered as a foreground from its background. Used thresholding to create sharper distinctions between the word and the background.

3.2.2 OCR for Non-Document

We have classified any image with sparse, fancy text, in no particular format with any background as non-document.

For e.g. Grocery items, household items, sign boards, name plates, packaged food, etc.

Figure 18: Examples of Non-Document

3.2.2.1 Dataset Preprocessing & Analysis

Initially we wanted to leverage a coco-text dataset with 83k images but we found them to have inconsistent mapping between the bounding box and the image itself. Hence we used the ICDAR dataset with 1k train images and 500 test images. All the 1000 train images had corresponding mappings denoting bounding box i.e. 4 coordinates of a rectangle each in x-y plane starting from top left corner in the clockwise direction.

3.2.2.2 Model

The same text detection model along with the loss functions of OCR documents are also being used for OCR non documents as it performs brilliantly on such non-document images as well. Refer to section 3.2.1.2 for more details about the text detection model.

3.2.2.3 Algorithm and Approach

Several problems were encountered while creating OCR for non-document. In order to overcome these issues we created a pipeline with multiple steps of processing.

The following pipeline was formulated for OCR in non-document:

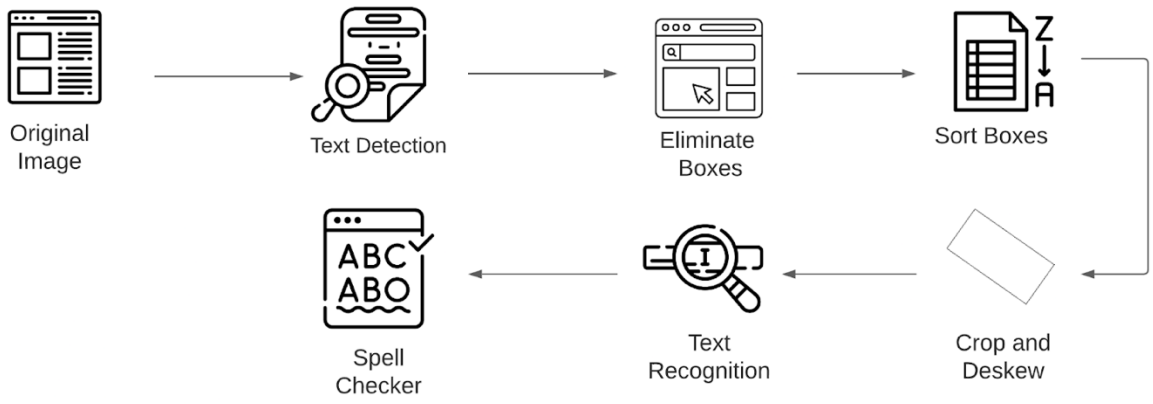


Figure 19: OCR Non-Document Pipeline

We take two images as our running example and apply all the different steps and see their respective outcomes. The images are specifically selected to entertain different corner cases, one being PEC's main entry gate board(sign board), and the other being a packet of Kurkure (packaged food).

Figure 20: Sample Images for OCR Non-Document

1. **Text Detection:**

The model identifies words and creates a bounding box around each word. As explained in the section 3.2.1.2, the model identifies and isolates each word and create a bounding box around it.

In order for the model to speak out relevant information only in a comprehensible order we will other different steps like eliminating boxes, sorting etc which are discussed below in detail.

Figure 21: Images after Text Detection

2. Eliminate boxes:

The problem with text detection was that it was detecting minute boxes which were barely recognizable even for a human eye. We thought that these boxes added unnecessary confusion since the information in such boxes was irrelevant anyways. So to eliminate these boxes we used the ratio of box area to the overall area and if the ratio is below a certain threshold we would simply omit those boxes. Threshold value was selected on an iterative basis and set to 0.001.

Figure 22: Images after Eliminating Boxes

Notice that the text in logo of PEC are eliminated in the first picture where are too small to be readable by human eyes anyways.

3. Sort Boxes:

Since a non-doc image may contain sparse text whose orientation is not always horizontally aligned as opposed to documents. Due to this reason we had to use a different sorting approach for non-documents. However, simply sorting by x and y didn't account for a bit of misalignment during bounding box creation and as a result we would get an incorrect result. Due to this reason we used the sorting function that basically weighs in balance between x and y coordinates of the boxes to find the proper order. The following mathematical equation was formulated to sort the boxes in the desired order.

tolerance_factor=125

*value = ((y_min // tolerance_factor) * tolerance_factor) * cols + x_min*

On the basis of *value* obtained for each box, we assign a position for the boxes. The order of the boxes will be assigned according to their values in the ascending order, resulting in sorted output.

Figure 23: Images after Sorting boxes (with numbers denoting order)

4. Rotating the boxes:

After isolating each word with their bounding box we realised that the orientation of the word played a significant factor in accuracy of recognition. The reason for this was the nature of dataset recognition model was trained on. Since in non-document category words can be on any orientation we decided to rotate the image in such a way that they are always horizontally aligned thus significantly improving the accuracy of recognition.

Following are the all possible cases that we entertained and accordingly they would be rotated. First the program would find the slope of each of the boxes with respect of their horizontal axis and then will simply rotate those boxes with the same angle in opposite direction thus making the new box parallel to horizontal axis

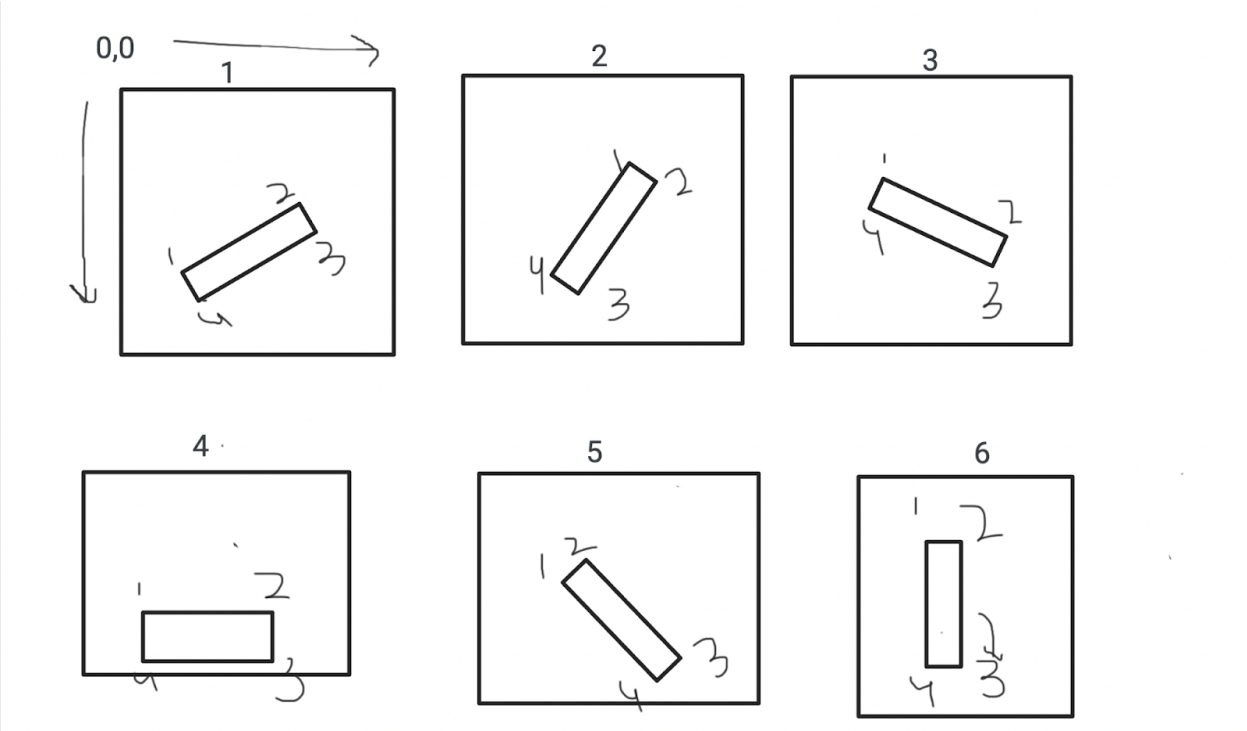


Figure 24: All Possible cases of inclination of bounding box

Following are the output before and after the rotation of image:

Figure 25: Examples of Image Rotation

5. Text Recognition:

Initially we tried using tesseract for the recognition of non-documents given the reliability of tesseract and also to maintain consistency with the OCR documents. However upon using it we ran into following major problems

1. Word by word,
2. Fancy text

Despite several of our efforts to run the non doc prediction in tesseract we were facing two major problems. First, tesseract's performance saw a massive degrade in case of fancy fonts which is more often the case in non-documents. Secondly the performance of tesseract on a single word, despite specifically tuning the model for it, was not up to our satisfaction. We used a model from *Clova AI* which was specifically trained with fancy text that specialized in prediction of individual words. We found that the performance of this model was much better than that of tesseract. This model combined with all the steps of our pipeline gave us desirable results.

6. Spell Checker:

In case of some text (usually the ones with unusual fonts), the model would give predictions that would be off by a few characters, resulting in incomprehensible words. In order to mitigate this

we used a spell checker specifically for the words with lower confidence so that any misread words would be fixed.

3.3 Live Location Sharing

We realized that for the loved ones of the visually impaired it was crucial to constantly keep on monitoring them so that if any emergency befalls, they can reach as soon as possible and help them. Furthermore, we felt that it provided a significant level of assurance simply by being updated to their live location.

Sharing live location has its advantages as: It allows you to keep your family and friends informed about your location in real time.

Live location feature works in the following manner:

- Whenever the user enters the location sharing screen, he/she gets the option to add guardian. The name, email and number of the person is added, which will be used for communicating the live location of the visually impaired person.
- Add atleast one guardian to use the live location sharing feature.
- Once the gaurdians are added, you can use the toggle button on the top right which instructs the backend to deliver the live location to all the guardians in the list through email and SMS.
- The background task in the app keeps updating the live location of the person after every few seconds.
- To access the live location, open the link received via mail or SMS.
- The link will redirect to the last updated location on backend.

CHAPTER 4

IMPLEMENTATION DETAILS

4.1 Mobile Application

4.1.1 Splash Screen

The mobile app first opens the splash screen, which is the first screen when the user opens the app. Drishti is seen written on this page, along with an image denoting visually impaired people.

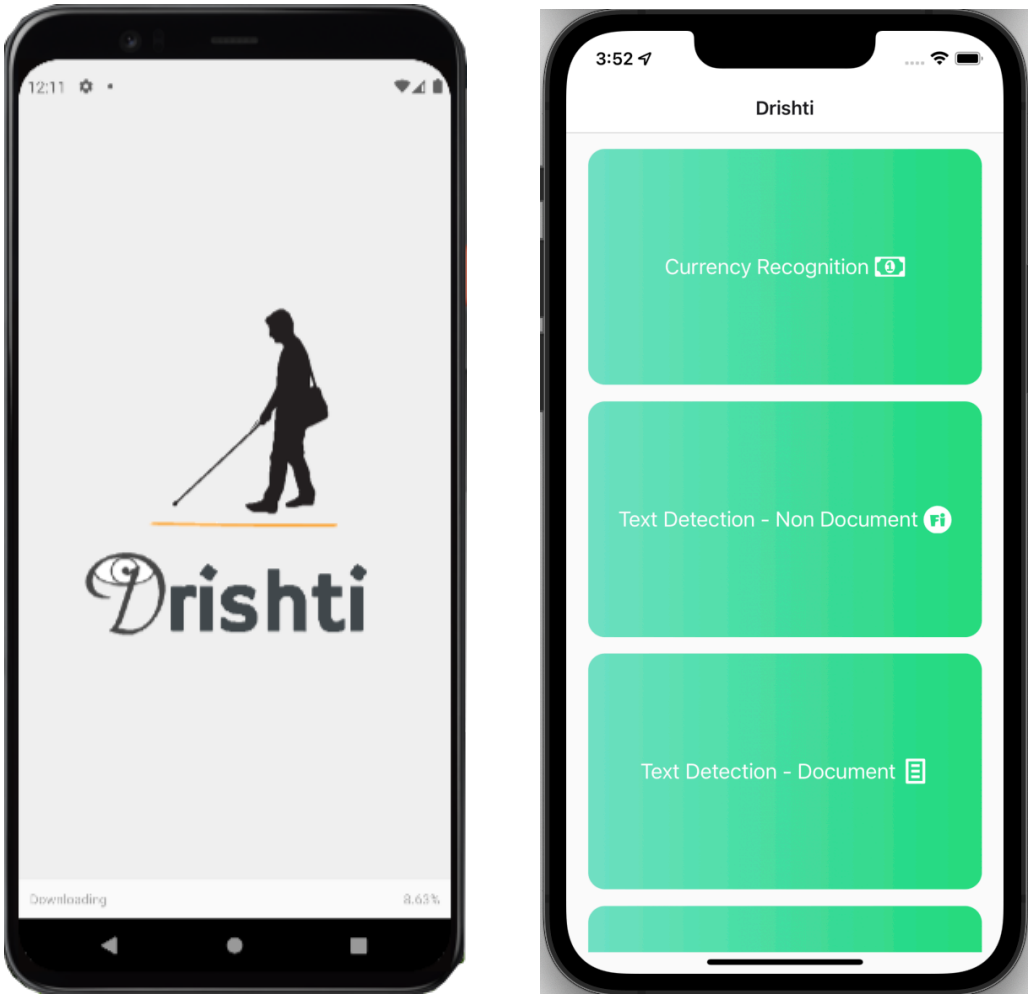


Figure 26: Drishti Splash Screen and Homepage

4.1.2 Drishti Homepage

This page contains cards for various features of our application. If the app doesn't have camera permissions, it requests for the same from the user. Long pressing any card will give audio output of the feature, which is helpful for a visually impaired person for navigating through the application. The page currently looks like as shown in below figure.

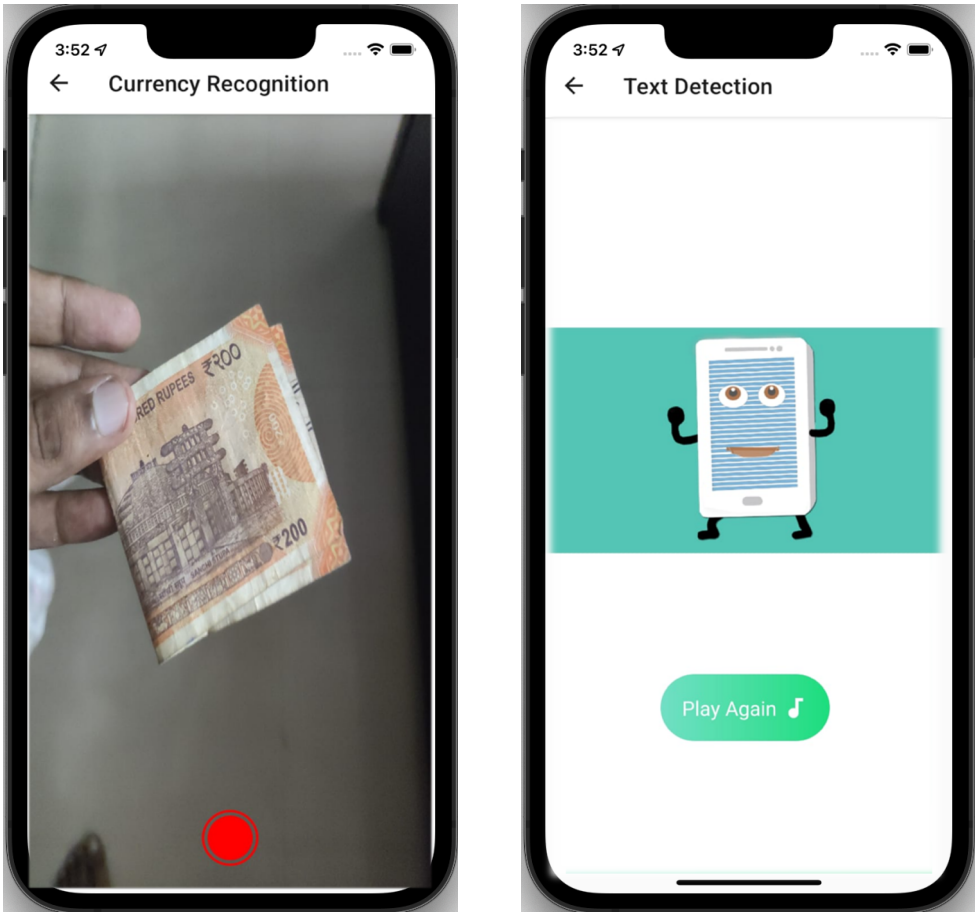


Figure 27: Currency Recognition and it's audio output

4.1.3 Currency Recognition

The camera screen will open, which will have a capture button to it. Once the button the camera will begin recording a 3 second long video which is then sent to server. The server separates the video into several frames and each frame is individually identified. Based on majority output the final currency is recognized and sent back to user.

4.1.4 Output Screen

Once the output is obtained from the server the user will be guided to an output screen where they will get the information in speech form. There is also a replay button incase the user misses to hear it well the first time.

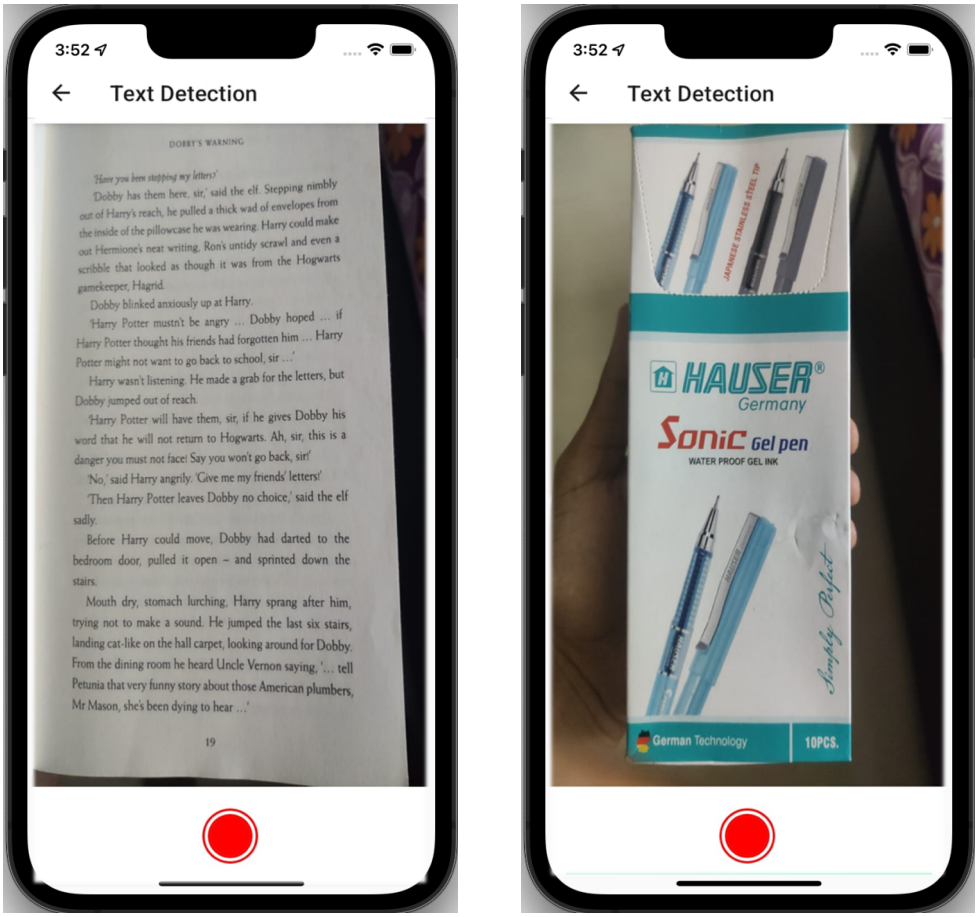


Figure 28: Text Detection - Document and Non-Document

4.1.5 Text Detection Document

On pressing the card for text-document, the user will be guided to a camera screen where they can capture the picture. The picture is sent to the server, in which the OCR for document pipeline runs. The recognized text is sent back to the app in string format which the app converts into speech.

4.1.6 Text Detection Non-Document

On pressing the card for text-non-document, the user will be guided to a camera screen where they can capture the picture. The picture is sent to the server, in which the OCR for non-document pipeline runs. The list of words recognized are consolidated into string and sent back to the app which converts into an audio format

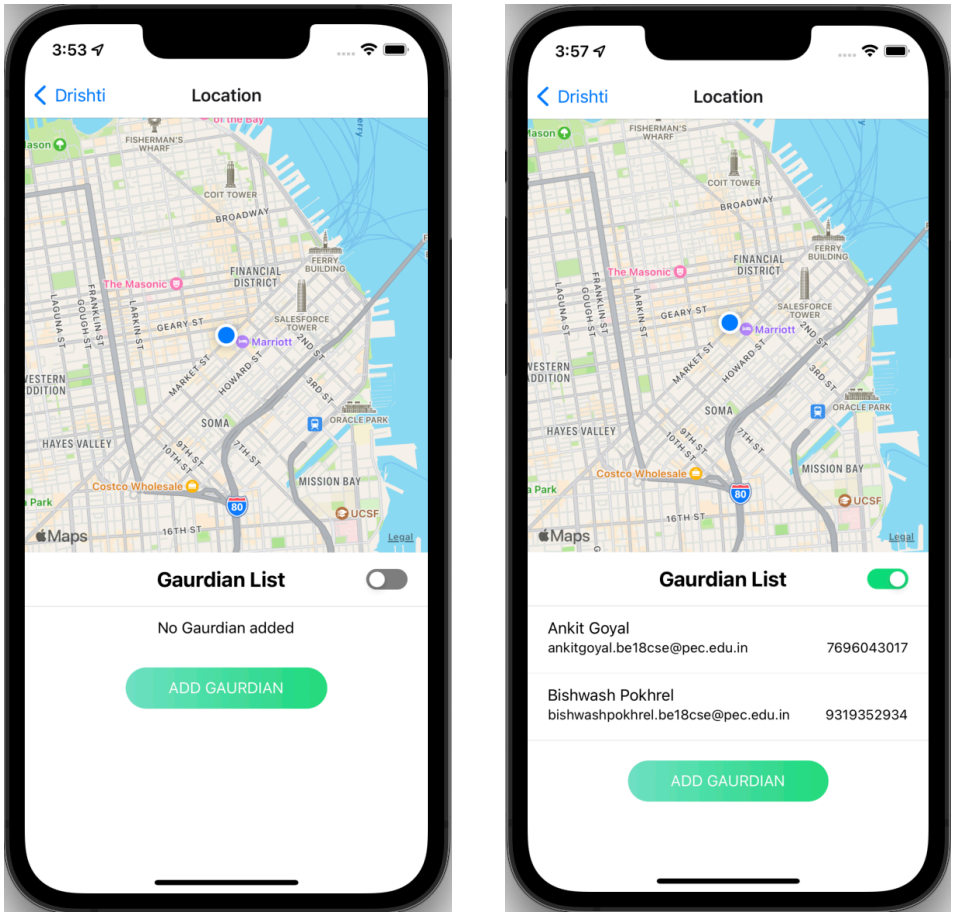


Figure 29: Live Location Sharing

4.1.7 Live Location Sharing

Whenever the user enters the location sharing screen, he/she gets the option to add guardian. The name, email and number of the person is added, which will be used for communicating the live location of the visually impaired person. Add at least one guardian to use the live location sharing feature.

Once the gaurdians are added, you can use the toggle button on the top right which instructs the backend to deliver the live location to all the guardians in the list through email and SMS. The background task in the app keeps updating the live location of the person after every few seconds. To access the live location, open the link received via mail or SMS. The link will redirect to the last updated location on backend.

The mail received after toggling the button looks like the following. By using the link given in the email, the user is redirected to the latest updated location of his/her loved ones.

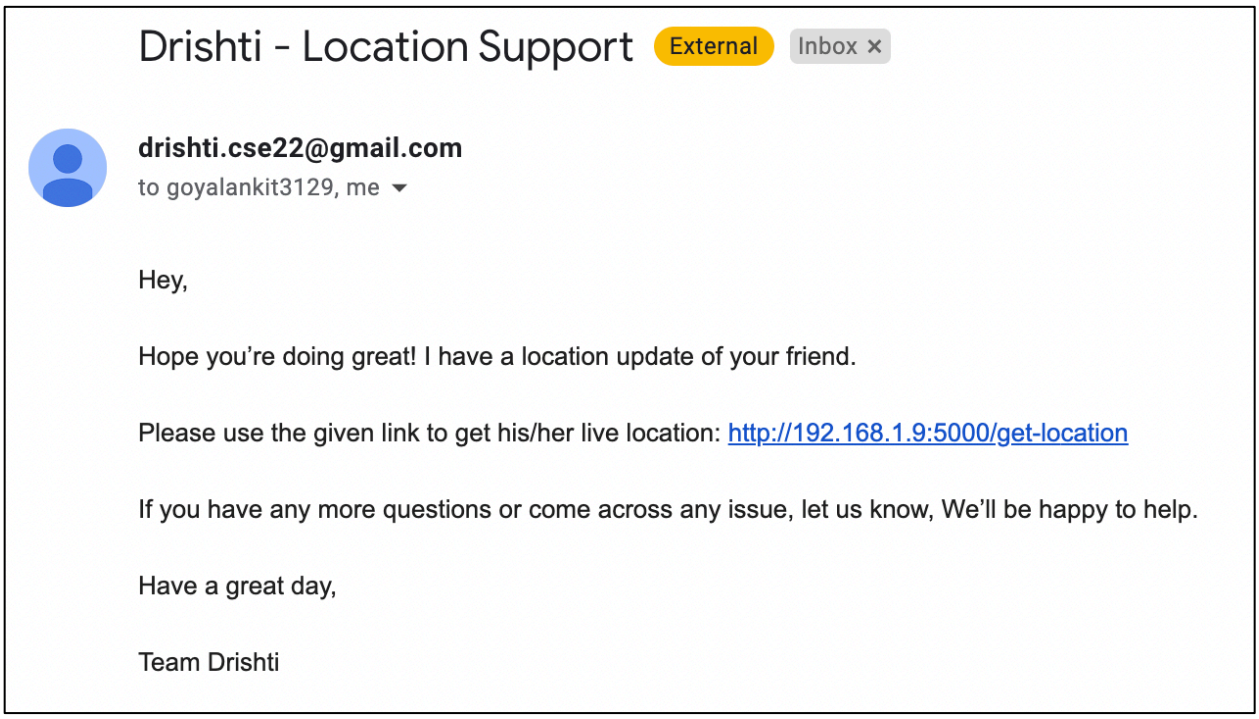


Figure 30: Live Location Sharing Mail

4.2 Mobile Application

4.2.1 React Native

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. In other words: web developers can now write mobile applications that look and feel truly "native," all from the comfort of a JavaScript library that we already know and love. Plus, because most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS. React Native currently supports both iOS and Android and has the potential to expand to future platforms as well.

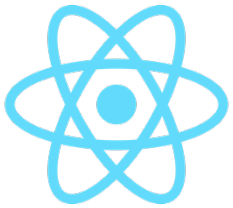


Figure 31: React Native

4.2.2 Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools.

Figure 32: Flask

4.2.3 Tensorflow

TensorFlow is a Python library for fast numerical computing created and released by Google.
It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an open-source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open-source license. The API is nominally for the Python

programming language, although there is access to the underlying C++ API.

Figure 33: TensorFlow

4.2.4 Tesseract

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License. Originally developed by Hewlett-Packard as proprietary software in the 1980s, it was released as open source in 2005 and development has been sponsored by Google since 2006.

Tesseract has unicode (UTF-8) support, and can recognize more than 100 languages "out of the box". Tesseract supports various output formats: plain text, hOCR (HTML), PDF, invisible-text-only PDF, TSV and ALTO (the last one - since version 4.1.0).

Figure 34: Tesseract by Google

4.2.5 Sublime Text

Sublime Text is a shareware cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically, community-built and maintained under free-software licenses.

Figure 35: Sublime Text

4.2.6 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, statistical modeling, data visualization, machine learning, and much more. We used Jupyter Notebook for writing code, compiling it and getting the results in modern fashion.

Figure 36: Jupyter Notebook

4.2.7 Atom

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control, developed by GitHub. Atom is a desktop application built using web technologies. Most of the extending packages have free software licenses and are community-built and maintained. Atom is based on Electron (formerly known as Atom Shell), a framework that enables cross-platform desktop applications using Chromium and Node.js. It is written in CoffeeScript and Less.

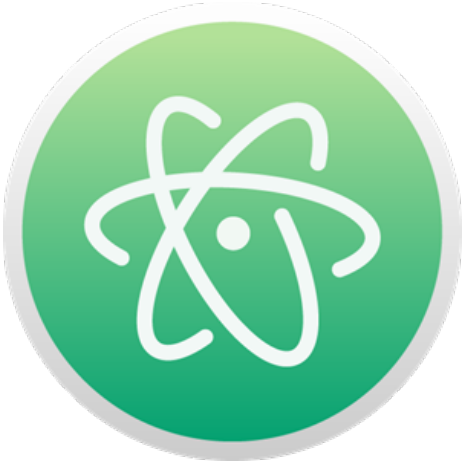


Figure 37: Atom

CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS

5.1.1 Currency Note Recognition

The following is the structure of our trained neural network:

- Base EfficientNet Input Layer
- EfficientNet Model architecture
- Final custom Fully Connected layers

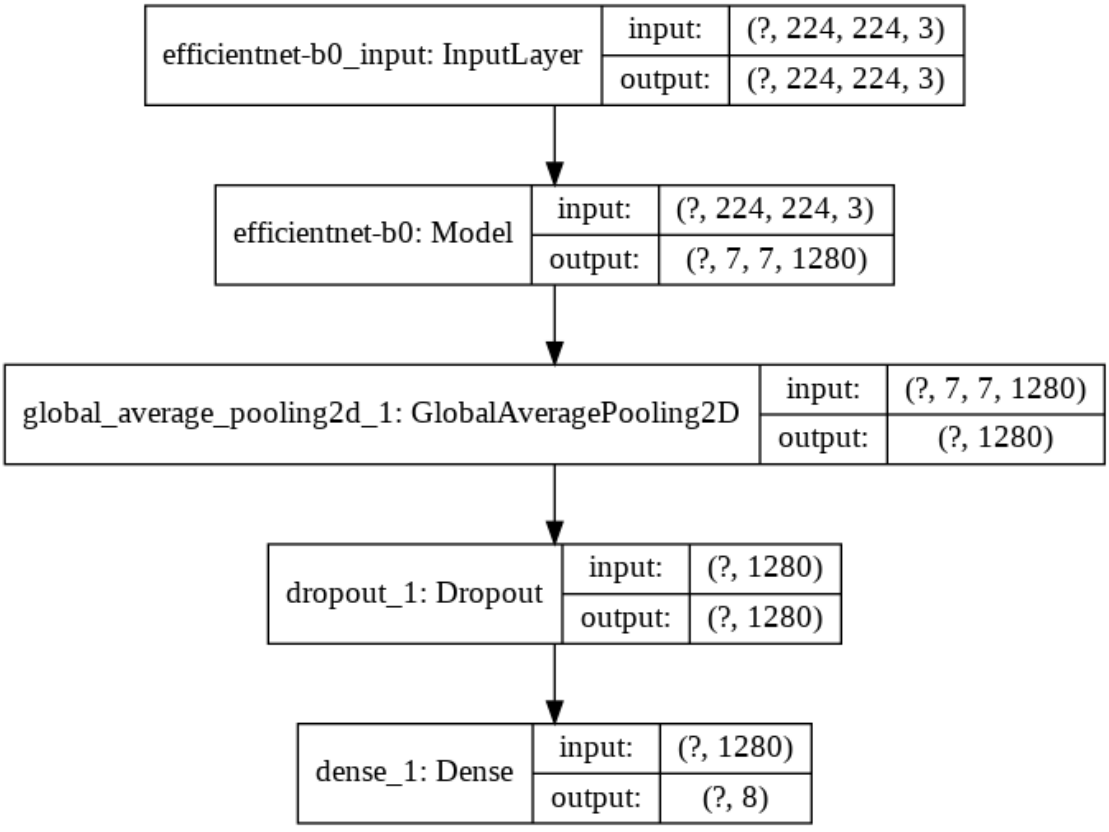


Figure 38: Neural Network Architecture

The results achieved so far are:

- We trained and tested our model on Google Colab.
- The training time for the most efficient model was around 5-6 hours for around 20 epochs.
- We tried retrieving the model over three optimizers, i.e. Adam, RMSProp and SGD (Stochastic Gradient Descent).
- Besides this, we did hyper parameter tuning by changing the learning rate and data augmentation.

Finally, we were successfully able to achieve a best validation accuracy of 99.81% and training accuracy of 99.50% using Adam Optimizer, sparse_categorical_crossentropy as the loss function and 0.0001 as the Learning Rate.

The following table shows the results of our training phase:

Table 1: Training and validation accuracy over different hyperparameters

Base Model	Augmentation	Size	Train-Acc.	Validation-Acc.
InceptionV3	low	155.9 MB	90.05%	87.5%
	high	228.1 MB	69.79%	69.4%
EfficientNet	low	15.8 MB	99.58%	99.44%
	high	**46.7 MB**	**99.5%**	**99.81%**

The model training phase was also performed by using VGG16 and InceptionV3 as our base models, but they were rejected because of the following reasons:

- VGG16: This model is a very heavyweight model, which slowers down our predictions. Since we require our predictions to be as fast as possible, so we decided to discard using this as our base model.
- InceptionV3: This model stopped learning after reaching certain percentage accuracy (very low accuracy value). Since the achieved accuracy wasn't satisfactory for our application, we decided to discard the same.

We have used different optimizers, learning rates and augmentations:

1. Optimizers: Adam, RMSProp & SGD (Stochastic Gradient Descent).
2. Learning rates: varying from 0.0001 to 0.001
3. Data Augmentation: Flips, Zoom, Shifts, Shear, Rotations

Model Predictions:

Our predictions had all bases covered. The model was found to predict correctly on most of the edge test cases. Some of the cases are listed below:

1. Normal case of currency prediction:

```
File Name: images\IMG_20211121_174820.jpg
Prediction: 20
```

Figure 39: Prediction Case 1

2. Folded currency notes:

```
File Name: images\IMG_20211121_175500.jpg
Prediction: 50
```

Figure 40: Prediction Case 2

3. Cropped currency notes:

```
File Name: images\IMG_20211121_230112.jpg
Prediction: 500
```

Figure 41: Prediction Case 3

4. Flipped and Cropped currency notes:

```
File Name: images\IMG_20211121_230151.jpg
Prediction: 200
```

Figure 42: Prediction Case 4

5. Image without currency note:

```
File Name: images\IMG_20211121_230424.jpg
Prediction: Background
```

Figure 43: Prediction Case 5

Hence, our model performed well on all the different scenarios.

The model training and validation accuracy on the first 10 epochs is shown in the below graph:

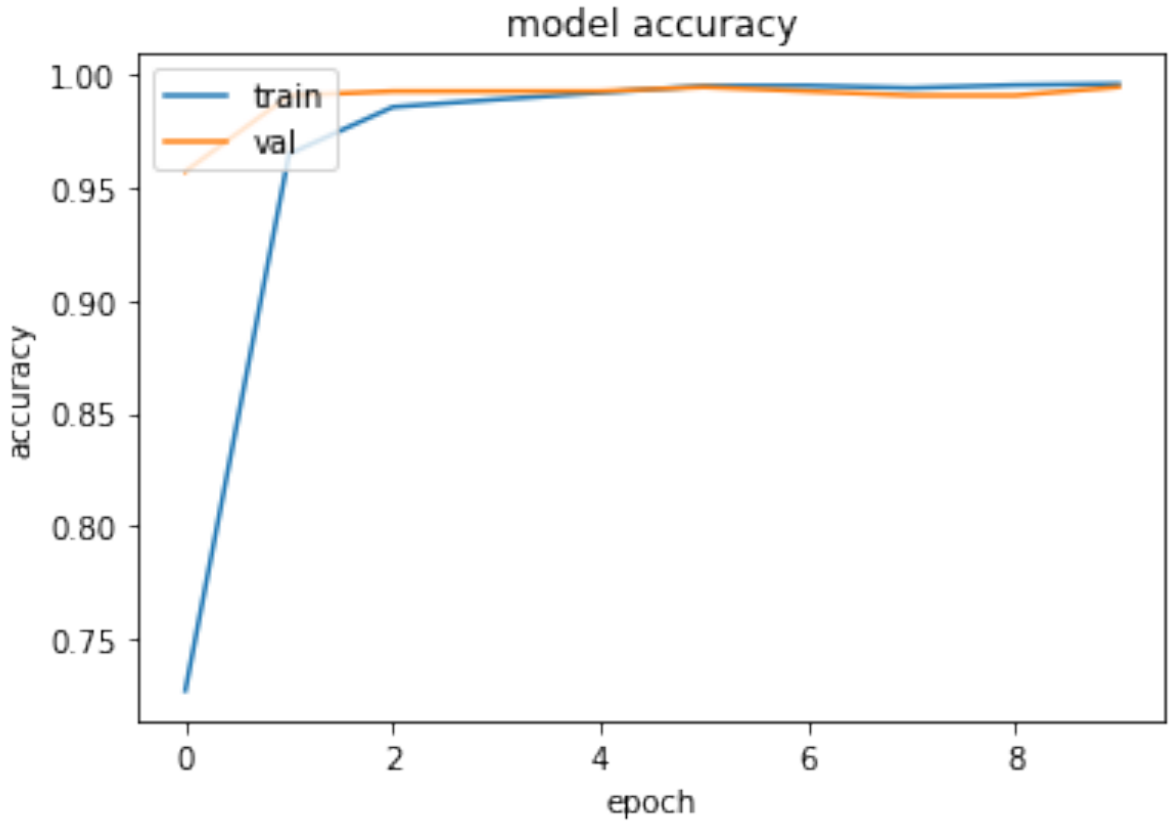


Figure 44: Accuracy Curve

5.1.2 OCR Text Detection

We trained two models (ResNet50 and VGG16) and evaluated their performance and loss. As we can see in the graphs below, the loss of VGG16 constantly decreases with every increasing epoch, which indicates that the model is learning well.

On the other hand, ResNet50 had a loss curve which keeps on fluctuating and doesn't decrease consistently. After a point, the loss stopped decreasing indicating that the model has stopped learning and any more training for further epochs won't help.

Hence, VGG16 was chosen as our final model for text detection in an image for both document as well as non-document.

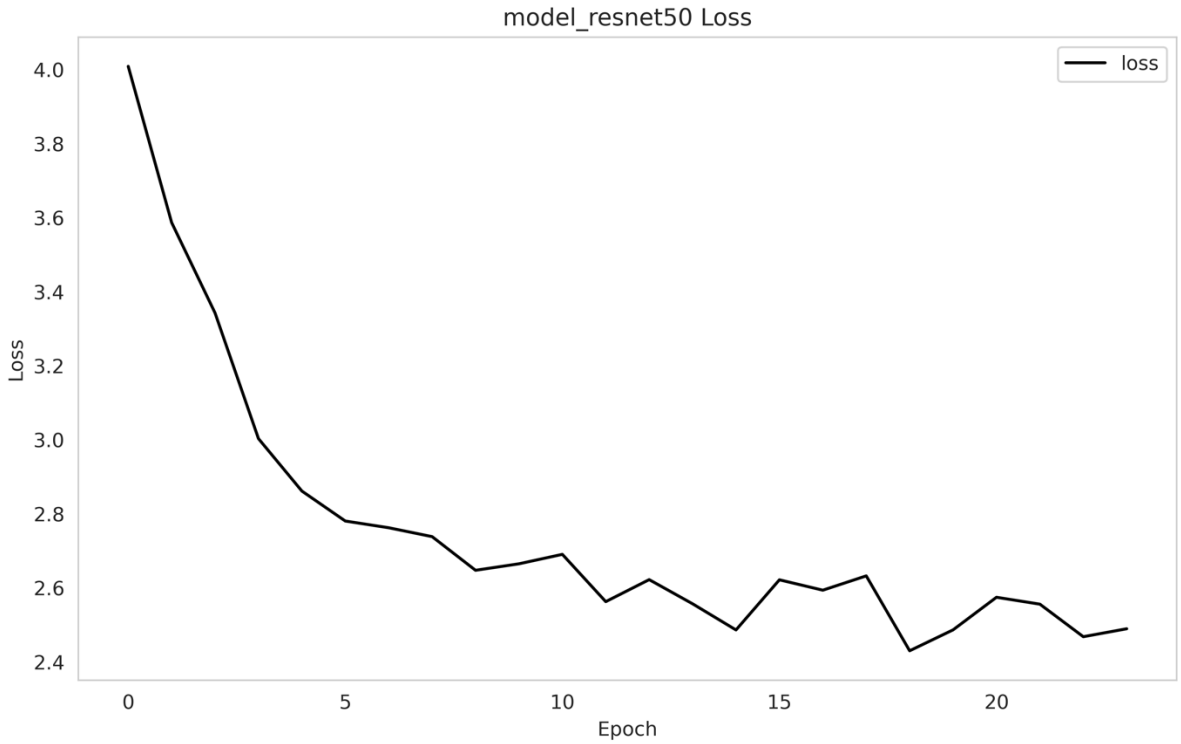


Figure 45: Loss Curve with ResNet50 as Base model

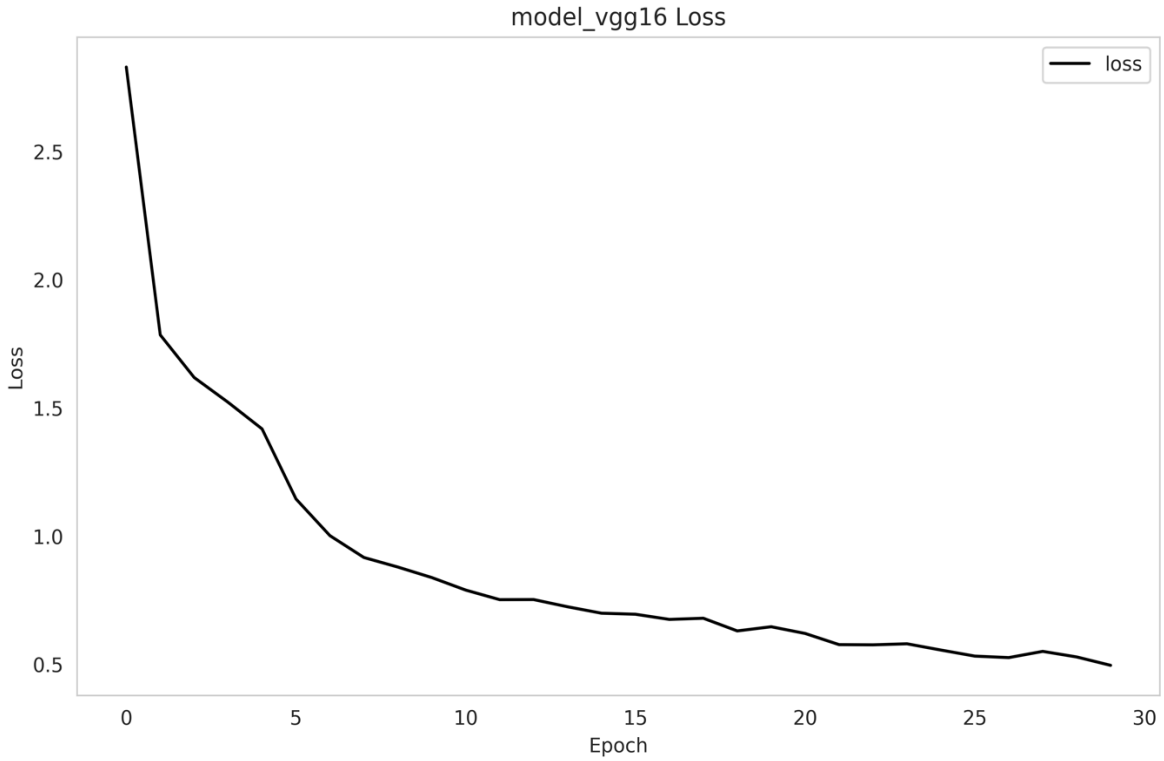
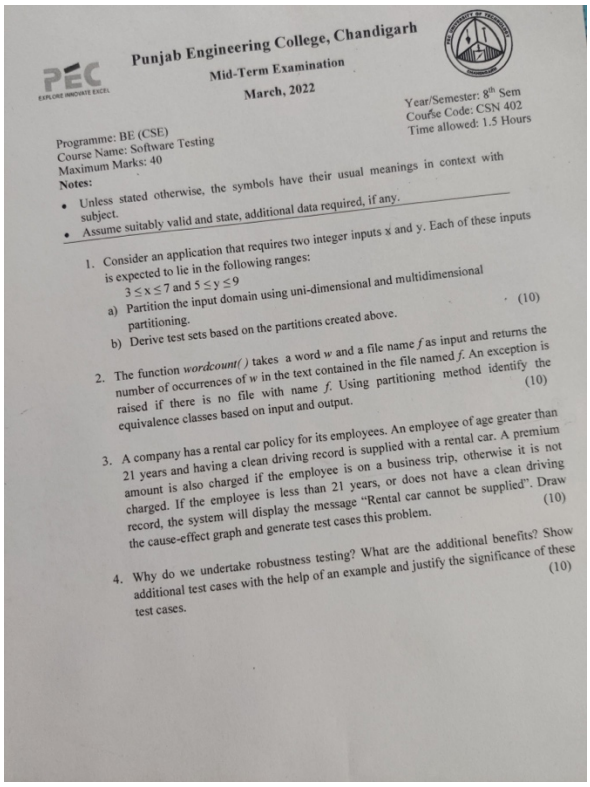


Figure 46: Loss Curve with VGG16 as Base model



Punjab Engineering College, Chandigarh

Mid-Term Examination

March, 2022

Programme: BE (CSE)
Course Name: Software Testing
Maximum Marks: 40
Year/Semester: 8th Sem
Course Code: CSN 402
Time allowed: 1.5 Hours

Notes:

- Unless stated otherwise, the symbols have their usual meanings in context with subject.
- Assume suitably valid and state, additional data required, if any.

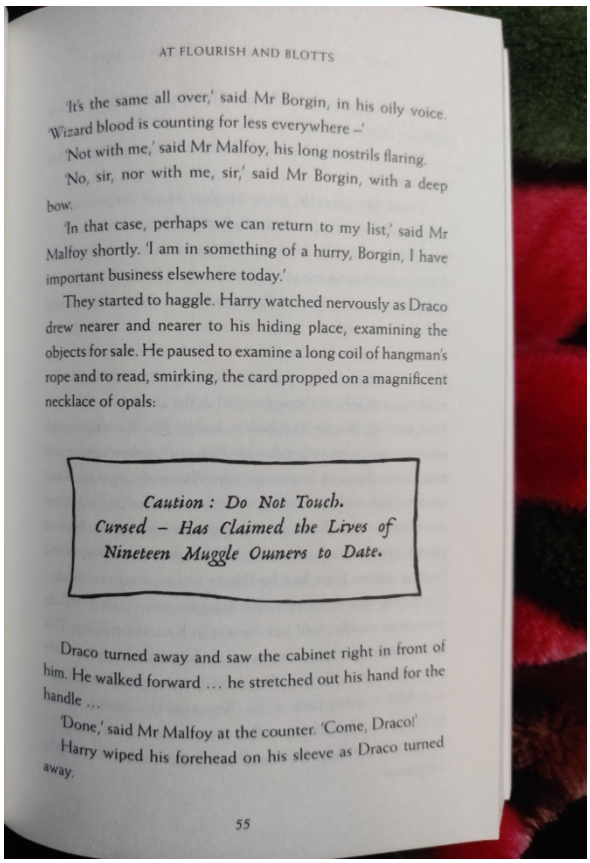
1. Consider an application that requires two integer inputs x and y. Each of these inputs is expected to lie in the following ranges:
   $3 \le x \le 7$ and $5 \le y \le 9$
   a) Partition the input domain using uni-dimensional and multidimensional partitioning.
   b) Derive test sets based on the partitions created above. (10)

2. The function *wordcount( )* takes a word *w* and a file name *f* as input and returns the number of occurrences of *w* in the text contained in the file named *f*. An exception is raised if there is no file with name *f*. Using partitioning method identify the equivalence classes based on input and output. (10)

3. A company has a rental car policy for its employees. An employee of age greater than 21 years and having a clean driving record is supplied with a rental car. A premium amount is also charged if the employee is on a business trip, otherwise it is not charged. If the employee is less than 21 years, or does not have a clean driving record, the system will display the message "Rental car cannot be supplied". Draw the cause-effect graph and generate test cases this problem. (10)

4. Why do we undertake robustness testing? What are the additional benefits? Show additional test cases with the help of an example and justify the significance of these test cases. (10)

```
Punjab Engineering College, Chandigarh
EXPLORE INNOVATE EXCEL
Mid-Term Examination
March, 2022
Programme: BE (CSE)
Year/Semester: 8" Sem
Course Name: Software Testing
Course Code: CSN 402
Maximum Marks: 40
Time allowed: 1.5 Hours
Notes:
Unless stated otherwise, the symbols have their usual meanings in context with
subject.
Assume suitably valid and state, additional data required, if any.
Consider an application that requires two integer inputs x and y. Each of these inputs
is expected to lie in the following ranges:
3<x<7and5<y<9
Partition the input domain using uni-dimensional and multidimensional
partitioning.
b) Derive test sets based on the partitions created above
(10)
2. The function wordcount() takes a word w and a file name fas input and returns the
number of occurrences of w in the text contained in the file named f An
exception is
raised if there is no file with name f
Using partitioning method identify the
equivalence classes based on input and output.
(10)
3.
A company has a rental car policy for its employees. An employee of age greater than
21 years and having a clean driving record is
'supplied with a rental car. A premium
amount is also charged if the employee is on
business trip, otherwise it is not
charged. If the employee is less than 21 years, or does not have a clean driving
record, the system will display the message "Rental car cannot be
supplied". Draw
the cause-effect graph and generate test cases this problem
(10)
4
Why do we undertake robustness testing? What are the additional benefits? Show
additional test cases with the help of an example and justify the
significance of these
test cases.
(10)
```



AT FLOURISH AND BLOTTS

'It's the same all over,' said Mr Borgin, in his oily voice. 'Wizard blood is counting for less everywhere –'

'Not with me,' said Mr Malfoy, his long nostrils flaring.

'No, sir, nor with me, sir,' said Mr Borgin, with a deep bow.

'In that case, perhaps we can return to my list,' said Mr Malfoy shortly. 'I am in something of a hurry, Borgin, I have important business elsewhere today.'

They started to haggle. Harry watched nervously as Draco drew nearer and nearer to his hiding place, examining the objects for sale. He paused to examine a long coil of hangman's rope and to read, smirking, the card propped on a magnificent necklace of opals:

Caution: Do Not Touch.
Cursed – Has Claimed the Lives of
Nineteen Muggle Owners to Date.

Draco turned away and saw the cabinet right in front of him. He walked forward … he stretched out his hand for the handle …

'Done,' said Mr Malfoy at the counter. 'Come, Draco!'

Harry wiped his forehead on his sleeve as Draco turned away.

55

```
AT FLOURISH AND BLOTTS
'It's the same all over,' said Mr
Borgin, in his oily voice
Wizard blood is counting for less everywhere
'Not with me,' said Mr Malfoy, his long nostrils flaring.
'No, sir, nor with me, sir, said Mr Borgin, with a deep
bow,
'In that case, perhaps we can return to my list/ said Mr
Malfoy shortly. 'I am in something of a hurry, Borgin, I have
important business elsewhere today.
They started to haggle. Harry watched nervously as Draco
drew nearer and nearer to his
hiding place, examining the
objects for sale. He paused to examine a long coil of hangman's
rope and to read, smirking, the card propped ona
magnificent
necklace of opals:
Caution
Do Not Touch.
Cursed
Has Claimed the Lives of
Nineteen Muggle Owners to Date.
Draco turned away and saw the cabinet right in front of
him. He walked forward
he stretched out his hand for the
handle
'Done,' said Mr Malfoy at the counter. 'Come, Draco!
Harry wiped his forehead on his sleeve as Draco turned
away.
55
```

Figure 47: Original Document Images along with the Prediction

Punjab Engineering College Chandigarh Deemed To Be University

20 Dal,Corn,Rice Kurkure Chili Chatka 10% extra

Figure 48: Original Non-Document Images along with the Prediction

The image used for prediction and their corresponding predictions are given above.

5.2 Discussion

Through this project, we help a blind or visually impaired person by helping them in their everyday activities such as recognizing currency notes, live location sharing, and text to speech from a video feed. Also, this project has the potential to get implemented in different parts of the world helping users with their day-to-day chores.

The currency recognizer will really add a lot of value to the daily life of a blind or a visually impaired person as it involves financial transactions and helps them easily recognize currency notes, because recognizing it just through size and texture of notes is extremely difficult, as they are almost the same.

Also, the text to speech feature can be widely used for reading out text from a video input, which can be of a book, newspaper or text written on a product. Also reading out non documents would enable the visually impaired person in their day to day activities for example buying groceries, navigation with the help of sign boards etc

Also, with the help of live location, the loved ones of visually impaired person can track the location and effectively and reach out in case he/she needs any help.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Drishti is a mobile application made using React Native, Expo, JavaScript, and Flask while incorporating Deep learning techniques.

- A model trained to predict currency notes in a variety of situations, considering that the users of the app will be blind and visually impaired people.
- Model and individual pipeline for optical character recognition for documents and non-documents.
- Mobile App features:
    - Currency Note Recognition
    - OCR for documents and non-documents
    - Live location sharing to the loved ones
- Output: Voice output for guiding the users.
- Easy to use mobile app for the users.

We have made a Application, which helps a visually impaired or a blind person in their everyday life tasks such as Recognizing Currency Notes, reading text from documents such as books, newspapers, novels etc and also from other non-document objects such as grocery items, household items, packaged food, etc.

6.2 Future Scope

CURRENT CHALLENGES:

- In documents some portions of image are unwanted and disturb the meaning of the entire document. For example remains of other page while clicking a picture of the book. In such case our model is unable to distinguish the required portion from the irrelevant portions.
- Some documents have multiple column format for eg in newspaper. In such case the model will read lines row wise and wouldn't be able to distinguish columns
- In a non-document, if the text is too fancy, then there are slight degradations in the performance. Also, in certain special cases, sorting can pose a problem since the text can be in any orientation at any place in the image.

FURTHER SCOPE:

- Local Navigation Aide: We aim to guide the visually impaired people and inform them about the obstacles around them.

Figure 30: local navigation aide

- Text to speech from a video feed: We plan to convert a video feed containing text to speech. Basically, the user will have the text content underneath the camera, and then using a short video clip, we will process the text in it, and then give the processed text as an audio output for the user, which currently works for input images.

- Smart Wearable Glasses: An additional feature which we plan to implement is smart glasses, which will incorporate all the above-mentioned features in wearable glasses. The glasses will have a camera attached to it, which will directly take in input feed. The output for the same can be given using the glasses itself.

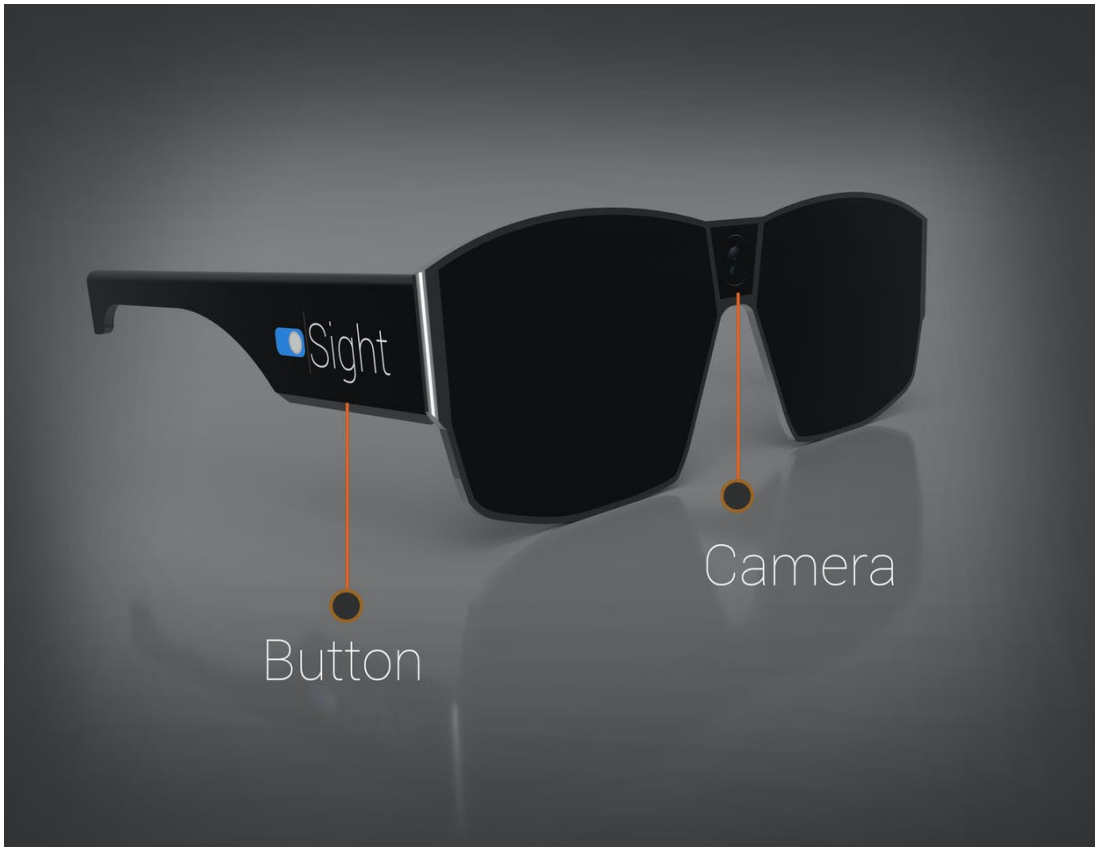


Figure 49: Smart wearable glasses

CHAPTER 7

REFERENCES

[1] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[2] Weiss, K., Khoshgoftaar, T.M. & Wang, D. A survey of transfer learning. J Big Data 3, 9 (2016). https://doi.org/10.1186/s40537-016-0043-6

[3] K. Gautam, "Indian Currency Detection using Image Recognition Technique," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020, pp. 1-5, doi: 10.1109/ICCSEA49143.2020.9132955.

[4] N. Patange, V. Dhutre, V. Wani and S. Oak, "Comprehensive Analysis of Indian Currency Recognition System and Location Tracking for Visually Impaired," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498430.\

[5] G. Singh Sodhi and J. Singh Sodhi, "A Robust Invariant Image-Based Paper-Currency Recognition Based on F-kNN," 2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE), 2021, pp. 1-6, doi: 10.1109/ITSS-IoE53029.2021.9615287.

[6] Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., & Lee, H. (2019). What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. arXiv. https://doi.org/10.48550/ARXIV.1904.01906

[7] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazan, and ´ L. de las Heras. ICDAR 2013 robust reading competition. In ICDAR, 2013. 5