Document Text and Template Identification, Data Extraction Using Machine Learning And OCR On Android Environment

Md. Minhazul Islam Rimon - 20101078
Md. Fuad Islam - 20101060
Tasnim Mobarak - 20101296
Kaushik Roy - 20101185
Mysha Samiha Priota - 20301205

Supervisor:
Dr. Md. Khalilur Rahman

Date of Submission
18/09/2023

Department of Computer Science and Engineering

BRAC University

Student's Full Name and Signature:

Minhaz

MD MINHAZUL ISLAM RIMON

fuad

MD FUAD ISLAM

Roy

KAUSHIK ROY

Tasnim Mobarak

TASNIM MOBARAK

Priota

MYSHA SAMIHA PRIOTA

Supervisor:
(Member)

Md. Khalilur Rhaman

Dr. Md. Khalilur Rhaman
Associate Professor
Department of Computer Science and Engineering
Brac University

Contents

Document Text and Template Identification, Data Extraction Using Machine Learning And OCR On Android Environment

Abstract

At present, we are living in a technical world in which digital devices are available to everyone. Be it advanced computers and tablets or in general smartphones, we even surpassed the age restriction and now toddler and pre-school children can somewhat understand smartphones. Even though we have the opportunity to use these devices as a medium to fill official documents and use it to carry virtual identification documents, we are still stuck with obsolete paper based documents. However, when it comes to sending the documents for official reasons, we have to scan and send them. Then comes the most manual part of the process of identifying the data of the documents and putting it in the system. We wish to automate this part of the process for better efficiency and remove the scope of error. As most people have access to a smartphone, we want to apply a machine learning model in an android environment so that it can identify documents and its templates with different sizes with data scattered in unique fashion for each of them as the person is using it and suggest recommendations. After identifying the area of text, we wish to use OCR - Optical Character Recognition to identify the text and upload it to a suitable location. This way we can save time by shortening three stages of processing, data uploading and verifying into one step of taking a picture of the document.

Introduction

In our daily life, we go through and process an impressive amount of documents. Documents like birth certificates, income tax, different types of government forms, passports, exam papers etc. Almost in all cases, we are required to extract data from these documents and copy them to a different virtual storage. These processes are tedious and time consuming. And in more than enough cases, there are bound to be some mistakes made which makes them even more obscure. That's why we need a system that will automate these processes and make these tasks more efficient.

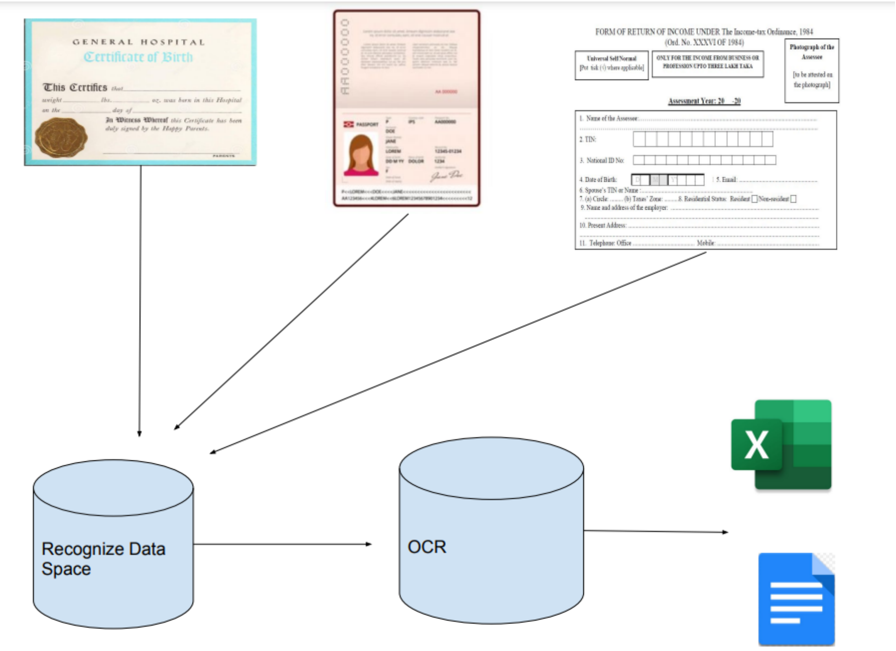
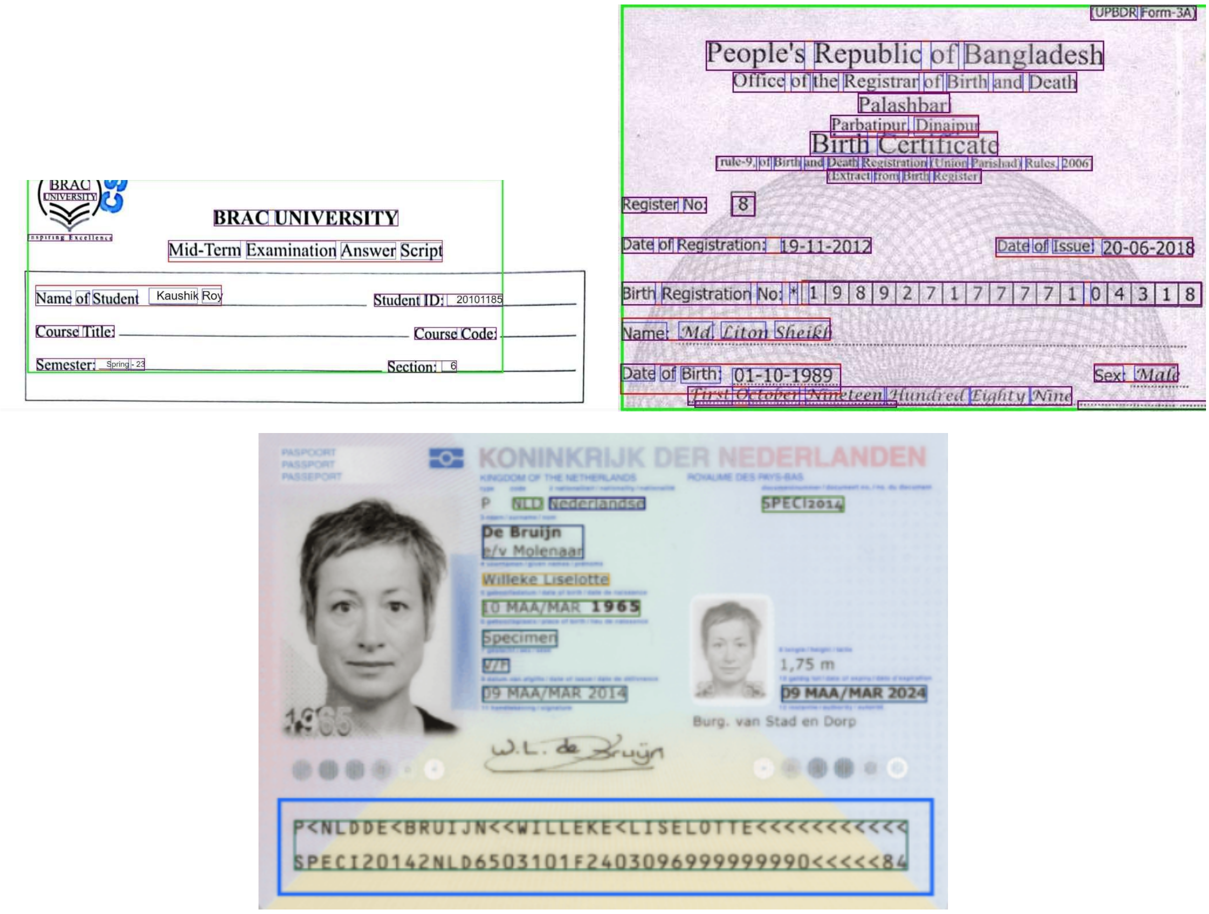


Figure 1: *General Process of the Application.*

The first obstacle that we need to overcome is identifying the document. Each type of document has their own characteristics and there are specific places from where we need to extract our data. We are approaching this problem with a two step process. Before extracting data from any type of document, we will ask the user to make a custom template where they will take a test picture and highlight the pieces from where data needs to be extracted. Behind these processes, we will have a Machine Learning model which will analyze the user behavior and learn the pattern of places where important data are stored. So, when the next time the user wants to create a template again, this model will give the user recommendation based on the previous cases. This ML process of test case and test data will occur simultaneously with the user using the system and it will be done offline. By being offline, we can ensure that a large number of users who do not have immediate access to the internet can use the service whenever necessary.



BRAC UNIVERSITY
Inspiring Excellence

BRAC UNIVERSITY
Mid-Term Examination Answer Script

Name of Student Kaushik Roy
Student ID: 20101185
Course Title:
Course Code:
Semester: Spring - 23
Section: 6

(UPBDR Form-3A)

People's Republic of Bangladesh
Office of the Registrar of Birth and Death
Palashbari
Parbatipur, Dinajpur
Birth Certificate
rule-9, of Birth and Death Registration (Union Parishad) Rules, 2006
(Extract from Birth Register)

Register No: 8
Date of Registration: 19-11-2012
Date of Issue: 20-06-2018
Birth Registration No: * 1 9 8 9 2 7 1 7 7 7 7 1 0 4 3 1 8
Name: Md Liton Sheikh
Date of Birth: 01-10-1989
First October Nineteen Hundred Eighty Nine
Sex: Male

PASPOORT
PASSPORT
PASSEPORT

KONINKRIJK DER NEDERLANDEN
KINGDOM OF THE NETHERLANDS
ROYAUME DES PAYS-BAS

P NLD Nederlandse
SPECI2014
De Bruijn
e/v Molenaar
Willeke Liselotte
10 MAA/MAR 1965
Specimen
V/F
1,75 m
09 MAA/MAR 2014
09 MAA/MAR 2024
Burg. van Stad en Dorp
1965
W.L. de Bruijn

```
P<NLDDE<BRUIJN<<WILLEKE<LISELOTTE<<<<<<<<<<<<
SPECI20142NLD6503101F2403096999999990<<<<<84
```

Figure 2: *Bounding-box on around the texts.*

For the data extraction part we will use OCR. OCR is currently one of the most researched topics as even if there are quite a few models on the market. None of them are able to achieve 100% accuracy. There are Google's cloud vision API, IBM datacap, Adobe readers etc. These software scans the written or printed paper characters, processes them and converts them to machine readable text. The available models that we discussed are available only on computer devices or doesnt work offline. For the convenience of the users and lift the restriction, we want to preprocess them locally in the android devices. For this we need to scan the documents first, then preprocess it to fit for extracting, then extract and classify it into machine readable texts. Then we show the output as classified text of the document.

Figure 3: *OCR (Optical Character Recognition)*

Finally uploading the extracted data into a suitable file. For this simple step we can add an algorithm to upload data to a local file or user connected file on google drive for user convenience. In addition to these, there are times when we have the option to check the validity of the data from an already existing database. For example: correct fathers name from old passport or birth certificate, correct tax id from old tax reports etc. In these cases, we can add a dynamic program which will crosscheck the extracted data from the existing database. When partial encounters of discrepancies in unique keys, it can alert users for checkup.

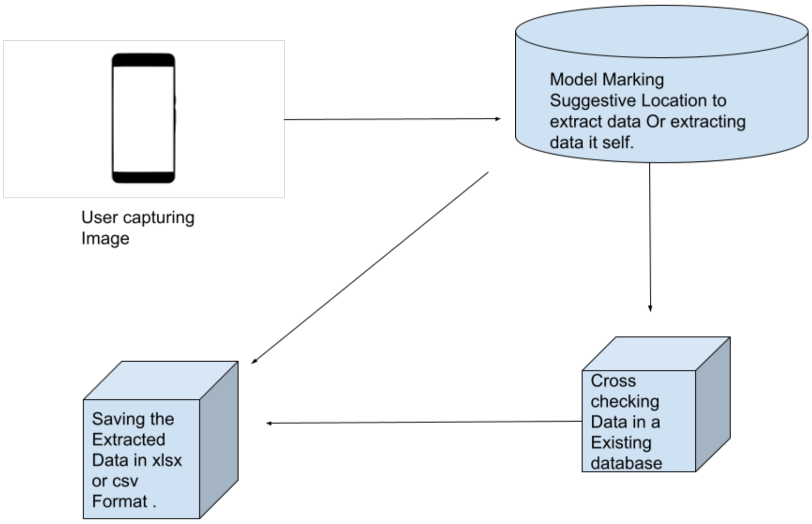

User capturing Image
Model Marking Suggestive Location to extract data Or extracting data it self.
Cross checking Data in a Existing database
Saving the Extracted Data in xlsx or csv Format .

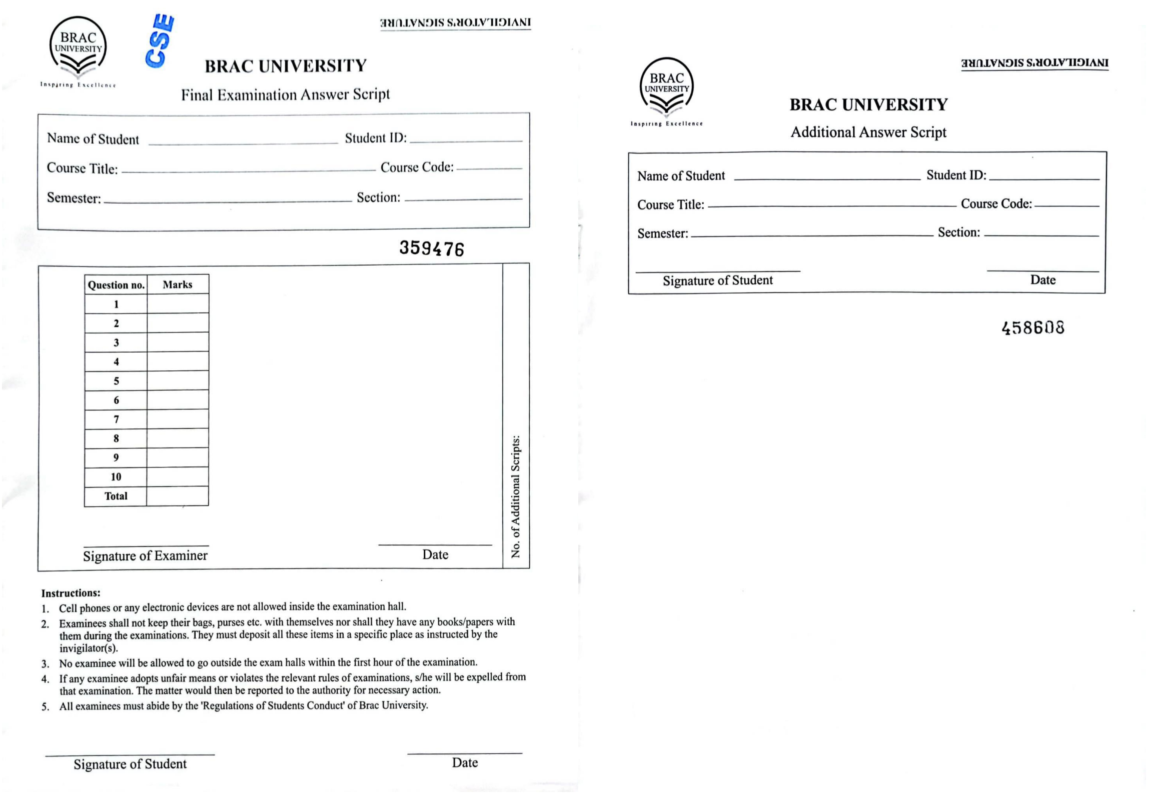
Figure 4: *Steps of working process of the Application.*

This three phase model aims to increase the efficiency of extracting data and improve the accuracy of existing OCR models. In addition to this, this will give users an easy to use and access environment of the day to day task. These report aims to discuss about following topics:

1. A detailed description of the current situation of the OCR models
2. Advantages and disadvantages of shifting to mobile device for these task instead of specific gadgets
3. Restrictions faced by past models and how we aim to overcome them
4. Suggestion of implementing Machine Learning algorithms in any android environment and effects

I. Research Problem

Nowadays everything is digitized and any information can be easily accessed if it is in document form. Every day we come across different types of documents which we often take pictures to preserve. Such documents include driving license, passport, national identity card, exam papers etc.The texts in these are written using a wide array of letters, numbers, and symbols. But if we could collect the information from such different pictures and keep it documented in one place, it would be convenient to arrange it as well as it can be found quickly when needed. Our goal is to to detect these distinct information from a picture and input them straight into an excel sheet.



BRAC UNIVERSITY

Final Examination Answer Script

Name of Student ____________ Student ID: ____________

Course Title: ____________ Course Code: ____________

Semester: ____________ Section: ____________

359476

Question no.	Marks
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total	

Signature of Examiner Date

No. of Additional Scripts:

Instructions:

1. Cell phones or any electronic devices are not allowed inside the examination hall.
2. Examinees shall not keep their bags, purses etc. with themselves nor shall they have any books/papers with them during the examinations. They must deposit all these items in a specific place as instructed by the invigilator(s).
3. No examinee will be allowed to go outside the exam halls within the first hour of the examination.
4. If any examinee adopts unfair means or violates the relevant rules of examinations, s/he will be expelled from that examination. The matter would then be reported to the authority for necessary action.
5. All examinees must abide by the 'Regulations of Students Conduct' of Brac University.

Signature of Student Date

BRAC UNIVERSITY

Additional Answer Script

Name of Student ____________ Student ID: ____________

Course Title: ____________ Course Code: ____________

Semester: ____________ Section: ____________

Signature of Student Date

458608

Figure 5: *Front Page of Exam Paper which will be used as template.*

In every educational setting, evaluating exam papers is a crucial task. Keeping track of the number the student received during their exam at a later date is likewise a time-consuming chore. The process of recording exam results is getting more and more time-consuming due to the rate at which the enrollment in educational institutions, particularly universities, has increased in recent years. Various types of sections can be found on an exam paper. Some of which include the student's name, certain details about his class, or some contain information about the grades he earned. All of these distinct pieces of data must be included in one document in order to maintain a record of a student's exam results. Our goal is to simplify this task through the findings of our research on the subject. When reviewing exam papers, the most frequent issue teachers encounter is having to spend a lot of time filling out an excel sheet with all of the student information. The final outcome is delayed significantly as a result of the lengthy process of uploading each student's grades into an excel sheet. We intend to automate this manual approach and reduce computational overhead while maintaining excellent performance and accuracy in document detection and data extraction.

Using machine learning and OCR to address the aforementioned issues might be a practical approach. Our goal is to include these methods into user-friendly software in the Android environment. But implementing such an approach involves several challenges. Because not all students have identical handwriting, this activity presents a number of difficulties.The way that each student writes letters and numbers may be different, thus there will be unique writing styles. The system would then need to be trained on various handwriting styles in order to recognize the same letter or number. Our objective is to reliably distinguish letters and numbers from various styles of handwriting.

II. Research Objective

This research aims to create an android application which will enable the users to scan and extract data from any kind of document with different kinds of templates ex: passport, certificate, government form etc. It will try to utilize the Android environment to create a ML and OCR model which will process the data accordingly. So the objectives with this model are:

1. Enable the user to extract data from scanned documents and save them to users convenience
2. Give user recommendation to the places from which data can be extracted
3. Upload the extracted data to convenient online or offline database or excel file
4. Crosscheck data from existing database (when available) to ensure data validity and accuracy
5. Enable processes in the local environment to ensure reliability when the internet is not available.

Literature Review

Shubham et al. [1] used CNN (convolutional neural network) model for recognizing handwritten texts. They used the Eminst dataset for both training and testing purposes. In their approach, they first pre-processed the data via normalization, rotating and reversing, input image filtering. With data processed and fit for training, they used the SEQUENTIAL model which consisted of 8 linear stacks of layers. Lastly, they used various optimizers available for keras to train their model. After experimenting with all the models,they found out Adamax, a first order gradient-based optimization method provided the most accurate results.Also they used 2 convolutional layers for maximizing accuracy. With Adamax, the trained model had an accuracy of 87,1 percent. Though other optimizers like Adadelta, Adam, SGD provided close enough accuracy, Adamax reduced the training time significantly. With better data pre-processing methodology the accuracy, efficiency of this model can be improved and for our work using a 2 convolutional layers model with Adamax optimizer provided by keras is a viable option.

Shruti et al. [2] used semantic segmentation and other pre-processing methodology to recognize mixed data (handwritten, printed texts) faster. In their work, they prioritize pre-processing data over improving conventional OCR models. Their approach focused on using any OCR engine readily available and getting optimal results by inputting fit data for training which they would get from their elaborate pre-processing techniques. First the input image will be pre-processed where binarization, noise reduction, slant removal, text alignment issues were performed. The resultant image was then processed using Liner removal, Gray Scale Conversion, Gaussian Blurring, Thresholding, 3 Channel Re-Conversion. Afterward, using U-net image segmentation was performed and on the output using OCR engines digital text was generated from handwritten/printed text. In their image processing module, there was a label isolation model that allowed them to segregate handwritten texts from printed text. This module can be used in our research for using segregated printed text for recognizing and categorizing different labels. Also, their pre-processing approach is also very viable for our approach as we want to recognize texts into different segments based on labels. Their work focused on forms which are an example of mixed data while our work focuses on exam transcripts information page where student information and grading is provided in mixed data as well.

Zhengchao et al.[3] combined CNN and RNN to recognize scene text. CNN was used to extract features from an input image. Different descriptors like VGG16, VGG19, ResNet34, ResNet50 were used for feature extraction. Now extracted data was sequential feature map and CNN has a disadvantage in such cases. This is where RNN which can extract sequential objects of arbitrary lengths comes in. After experimenting with different neural networks they managed to combine advantages of both CNN and RNN and claimed that a deeper CNN with deep descriptor will be more effective in predicting scene texts.

Yi Jiang et al.[4] proposed a text recognition algorithm to solve text segmentation difficulties, dictionary dependence using Attention mechanism and connection time classification.Here, low resolution images are upgraded using ESPCN into high resolution images. Afterward via multi-scale CNN, features are extracted. Later using Attention - CTC said encoded feature sequences are decoded. Traditional RCNN has great success in recognizing text but the training takes a significant amount of time and the calculations are large. Via the algorithm identifying the target is simpler and takes less time and calculation. In their work, they came to the conclusion that training models based on Attention CTC avoids the gradient disappearance problem of RNN. Proposed algorithm here can be advantageous if it can be implemented on android devices as effectively reducing time needed to operate would lessen the burden on the hardware.

Again, Himank et al. [5] proposed faster and efficient pre-processing methodology. Here geometric rectification, pre- processing, image detection, text extractions are key elements that are used for recognizing text data. Findings of their experiments suggest that though their approach reduces the time complexity and simplifies the issue at hand for untrained text fonts the approach seems to be lacking. However, with trained text fonts the accuracy rate of successfully digitizing texts is around 80 percent. We can improve the OCR model to make it adaptable to untrained fonts and use the proposed methodology in our endeavor.

Xia et al. (2022) proposed a Transformer-based encoder-decoder structure with a two-stage attention mechanism for scene text recognition. The overall location of the text in the image is captured at the encoder using a first-stage attention module merging spatial attention and channel attention, and the position of each letter in the text image is precisely determined at the decoder using a second-stage attention module. According to their research findings this two-stage attention technique can more precisely locate the position of the text and increase recognition precision. For the encoder, they also created a multi-branch feature fusion module that can combine features from several receptive fields to produce more robust features. This study suggests that this framework can speed up training and is more conducive to learning than the RNN-based STR technique. However, it also adds that because the input is an image, the convolutional layers are used in place of the linear ones in the encoder of this framework, while the same structure as the original Transformer encoder is maintained.

In their work,Panchal et al. (2022), investigated how Android's image recognition technology can extract text and features from images. They studied acceptable algorithms and their accuracy, which is a vital component of each paper, by using a variety of research papers from other academics. Every research paper was examined in the context of Android technology. Utilizing the earlier recommended methodologies and algorithms, feature and text extraction work has reached 85% to 88% accuracy as of 2012. Since Android has been widely used since day one, using an android application makes this analysis more authentic. The paper concludes by suggesting that even though there are many methods and APIs accessible, there isn't one single, universally applicable solution because it relies on the application's requirements.

Anarghya et al. (2020), worked on stroke detection and Hog transformation method. Their strategy presents a method towards character identification and recognition that combines the advantages of feature extraction methods associated with components distribution. The algorithm focuses on the clarity of text background segmentation. After reviewing some research papers and algorithms, they proposed integration of the Hog transformation and stroke detection method to achieve higher accuracy in localization and recognition in OCR technology. This approach highlights its efficiency in dealing with different text patterns, light and shadow conditions, and linguistic issues. Standard datasets have been demonstrated and improvement on previous OCR technology has played a significant role in providing a solution for real-time scene text recognition. This paper used MODI which can create difficulty in getting contents from books.

Jyothi et al. (2020), used OCR model to detect text and separate them into different graphical portions. Their proposed OCR model can classify different identity documents such as, passport, license into different categories. First of all, their model categorized the photos and derived information from the text extraction module and then the identification details were stored in the database. A new neural network based OCR engine LSTM(Long Short-Term Memory) is used to detect character patterns in this research. The research is mainly focused on the main project's purpose, technologies employed, used databases and the functional procedures. But they couldn’t define a best solution for making the algorithm more efficient.

Implementation

To implement the model for real world scenarios and to maintain accuracy, the steps will be:

1. Template gather: As there are many different types of templates, in order to generalize the pattern of spaces where data are available. We need to gather different types of document templates such as: Birth Certificate, passport, tax document, financial document, exam papers etc.
2. Text data gather: As we want to process our OCR model locally, we want to initialize it with a bunch of hand written and typed characters in different fonts and shapes.

3. Template pre-processing: Before using it into a Machine learning model, we want to process it to cut out unnecessary areas such as logos, images etc.

4. Template processing for model: Now we want to process these templates to find the places for each unique case where the data are situated. Then the model will suggest the user's finding. Based on user response it will correct itself and apply that decision for future reference.

5. OCR pre-processing: Same as template, we want to only use the characters from our data set, extra spaces around and in between should be cut out to improve the accuracy.

6. OCR processing: With the character dataset, we will process them to identify the pattern for the characters and cross reference the output with the existing database (if available). Then show the output to the user and upload to the database.

7. Testing and Evaluation: In order to measure the performance of the model we will use different real world documents for testing. Effectiveness will be measured with appropriate metrics to ensure user satisfaction in real world scenarios.

8. Deployment: With appropriate model design and evaluation with real world tests, it would approximately take 7 months to make it to a feasible phase. Then we can deploy it for a real world environment.

9. Maintenance: The human handwriting is different from one another and it continues to differ as time progresses. To ensure the performance of the system and handle future gimmicks, a scheduled maintenance will be done to make appropriate tweaks. Additionally as user preference is saved in devices, we can gather them as a qualitative research data set to thoroughly understand user preference of text and data space and come up with appropriate solutions for the future.

Description of the Model

In our research, we want to utilize machine learning to identify the text boundaries and then another machine learning process to identify the text from that boundaries and utilize it. Machine learning is a subset of Artificial Intelligence, which allows a machine to recognize patterns from a given set of examples so that it can make decisions based on what is learned in the test data. It allows a machine to learn by itself and make improvements for better results.

To identify the text border and utilize it, we are using the EAST model. EAST is a deep learning model used for identifying text areas in images with various backgrounds. It extracts the input image using a variation of a convolutional neural network(CNN). These CNNs are effective at extracting different types of features such as bold/hierarchical and abstract hints from the image. With these layers of data extracted from the input, it then incorporates a multiscale feature fusion which ensures to detect text of various sizes and orientations in the image. With the data now processed, it predicts and scores different parts of the images as potential places for the text boundary and scores it down. Then based on the score, it generates and extracts the text with the boundary.

With the text and boundary in place, now we transfer this data to a RCNN model. The RCNN model has the CNN part which extracts the text from the image and ‘R’ stands for region. It detects the region where the text is detected. Which is similar to EAST. We are training the model with the MNIST dataset which contains 60000 training and 10000 testing images for digits. With the pattern learned from it, RCNN resizes and crops the input image in various ways so that it can extract the feature vectors. Then it classifies and determines the presence of a text is available or not which is usually done by a support vector machine. Like EAST, it also does bounding box regression. But to improve the efficiency we have decided to add both in our model.

After training the model, we compiled it. At this point the model we received was not adaptable in the android environment. So, we converted our model using tensorflow lite into a tflite model which is usable in android.

EAST

EAST stands for Efficient and Accurate Scene Text Detection, As per Zhou et al.[10], hence the name of the detector. It is a deep learning model, used to extract text from a scene or image. EAST is mainly renowned for its speed and effectiveness, contributing to the field of text identification, although it is equally recognized for its significantly simplified and efficient pipeline. We can use the EAST model to detect text boundaries in our form-like answer script front page to identify texts which can then be used to process the image better for text recognition models. The creators of the model claim that its pipeline can predict words and lines of text on 720p images in a variety of orientations while operating at 13 frames per second. The ability to bypass computationally expensive sub-algorithms, such as candidate aggregation and word partitioning, which other text detectors typically use, is the model's most notable advantage.

As per Long et al [11], the EAST algorithm is a modification of the conventional anchor-based default box prediction method. In the common SSD network, the default boxes of various receptive fields are recognized based on a range of feature maps of varying sizes.Generally The U-Net [13] structure, or gradual upsampling, is used in this model to merge all feature maps. The resulting feature map has c-channels and is 1/4 the size of the original input image. Each pixel on the final feature map is used in a regression to determine the bounding box of the underlying text line, which can be rectangular or quadrilateral. This process is done assuming that each pixel only corresponds to one text line. As the final result, the algorithm predicts the existence of text and geometries. EAST is the first project to experience a noticeable speedup. However, it makes a number of changes to the prior framework. EAST's base-network is PVANet [14]. In the ImageNet competition, it hits a fair compromise between effectiveness and accuracy. Additionally, it streamlines the entire process into a non-maximum suppression phase and a prediction network. The prediction network transforms a picture into a feature map. It is a U-shaped [13] convolutional network. Each point in this diagram is a feature vector that identifies the expected text occurrence. When using non-maximum suppression, feature vectors from the same text instance are combined. It surpasses competitors on most datasets and reaches cutting-edge speed with an FPS of 16.8.

As per Zhou et al.[10], To detect text in a natural landscape, EAST uses a single neural network and outputs the results as a text with a quadrilateral shape. It combines a convolutional network and non-max suppression (NMS). Convolutional networks are used to identify words in images, and NMS is used to combine all of the identified text or boxes into a single, substantial text or box. The neural network has been trained to predict the text and its geometry in the landscape. The method was developed for text detection and it produces detailed per-pixel text prediction. The following diagram illustrates the general pipeline of this method. First, pixellevel text score maps and geometry maps are created for an image using a Fully Convolutional Network (FCN). In this manner, a generic Dense box is produced. There are two geometric forms available for text regions. One is a quad box (QUAD) and the other is a rotated box (RBOX). After the loss function for both the maps has been constructed, the threshold is applied. If the score is higher than the predefined forecast, that region is passed on to the non-max suppression (NMS) method. The final product is what comes after NMS.

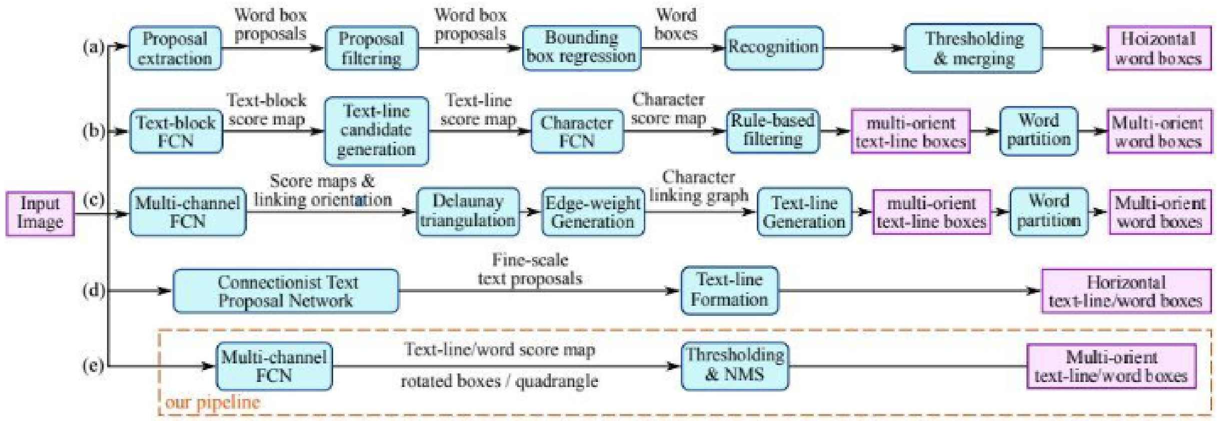


Figure 6: *Comparison of scene text detection algorithms from different recent studies. (a) Pipeline for horizontal word detection and recognition proposed by Jaderberg et al. [12]; (b) Pipeline for multi-orient text detection proposed by Zhang et al. [15]; (c) Pipeline for multi-orient text detection proposed by Yao et al. [16]; (d) Method of Horizontal text detection using CTPN, proposed by Tian et al. [17]; (e) Pipeline proposed by Zhou et al. [10] (EAST Model) which, according to the authors, is simpler and has two phases, eliminating the majority of intermediate procedures of the earlier systems.*

The fundamental element of this suggested method is the neural network model. It has been trained to accurately foresee the presence of text as well as their size or other geometric details from images. Thus the model provides dense per-pixel predictions of words or text lines. This is what makes the model unique and that is, it gets rid of intermediary procedures that were present in the previous models such as text region construction, candidate proposal as well as word partition. When creating a neural network, there are a lot of considerations to make. The size of the images and text in the natural setting varies widely. Thus it gets difficult to generalize the text's geometry. Which is why, in these feature maps Hypernet and UShape is utilized and a network with the ability to use various levels of features and with the least amount of computational expense is created. Below is a diagram outlining this method.Thus, the only post-processing procedures performed On anticipated geometric shapes are thresholding and NMS.

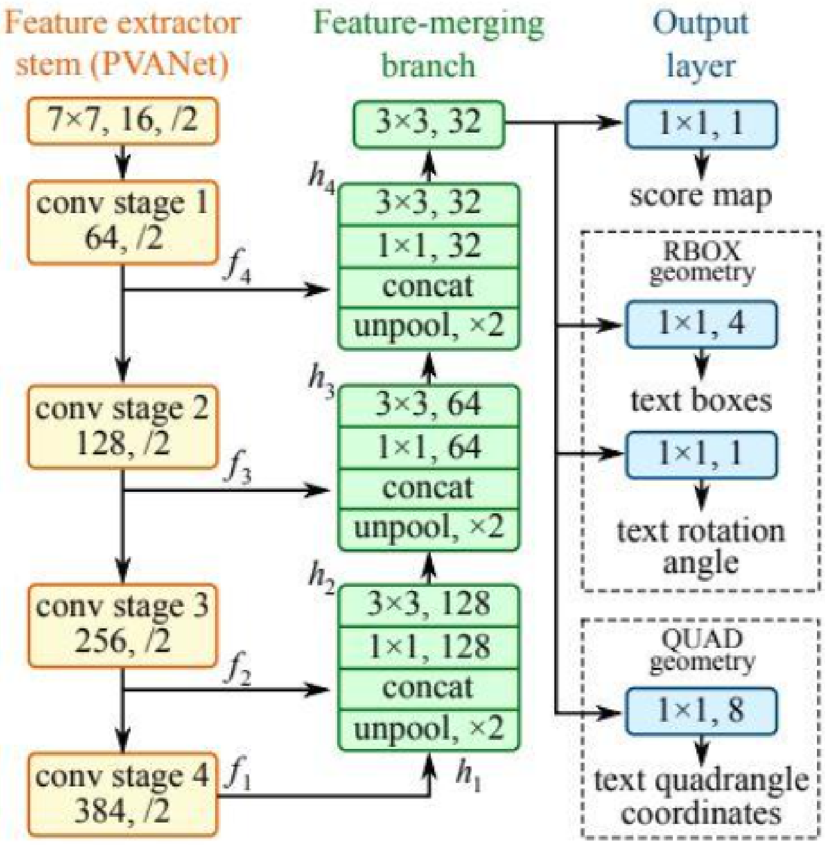
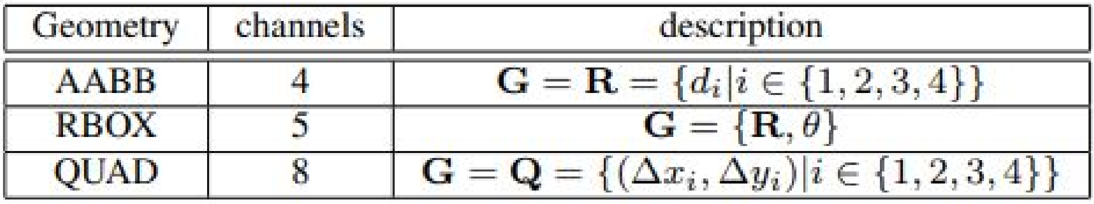


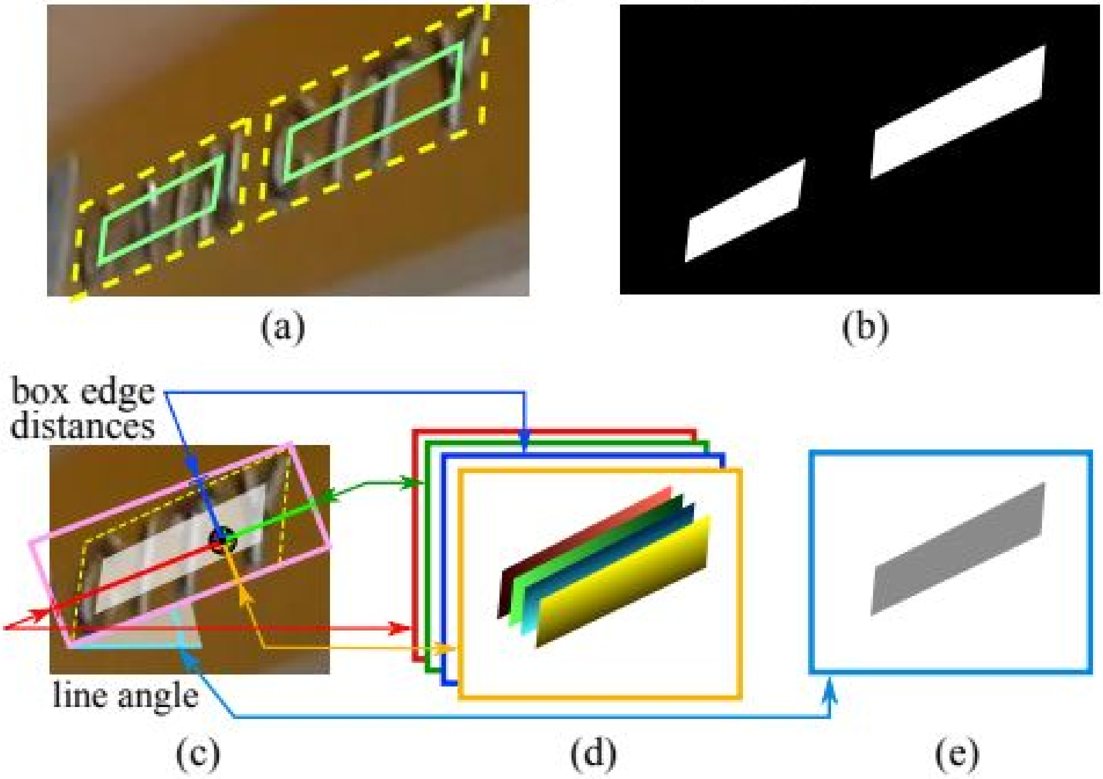
Figure 7: *Structure of EAST text detection FCN.*

The model can be broken down into three parts and they are- Feature Extracting Stem, which is a layer trained on an image net dataset with interleaved convolution and pooling layers. Feature Merging Branch which combines the featured output from the other layers. Here, U-Shape is utilized since manual merging is expensive. Finally, The final feature map is created and sent to the output layer. Finally comes the output layer. Score and a geometry map are the two components of this layer. The geometry map may be in RBOX or QUAD and the map is made up of the box's coordinates.



Geometry	channels	description
AABB	4	$\mathbf{G} = \mathbf{R} = \{d_i \mid i \in \{1,2,3,4\}\}$
RBOX	5	$\mathbf{G} = \{\mathbf{R}, \theta\}$
QUAD	8	$\mathbf{G} = \mathbf{Q} = \{(\Delta x_i, \Delta y_i) \mid i \in \{1,2,3,4\}\}$

Table 1. Output geometry design



(a) (b)

(c) (d) (e)

Figure 8: *Label generation process. (a) Text quadrangle (yellow dashed) and the shrunk quadrangle (green solid); (b) Text score map; (c) RBOX geometry map generation; (d) 4 channels of distances of each pixel to rectangle boundaries; (e) Rotation angle.*

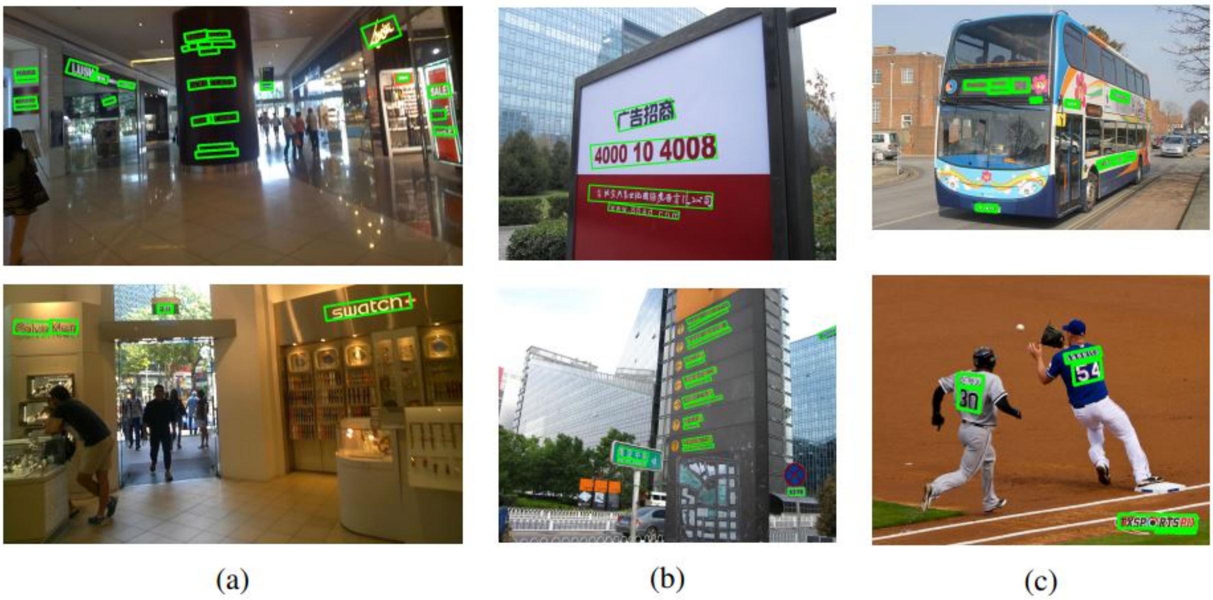


Figure 9: *Qualitative results of the proposed algorithm. (a) ICDAR 2015. (b) MSRA-TD500. (c) COCO-Text.*

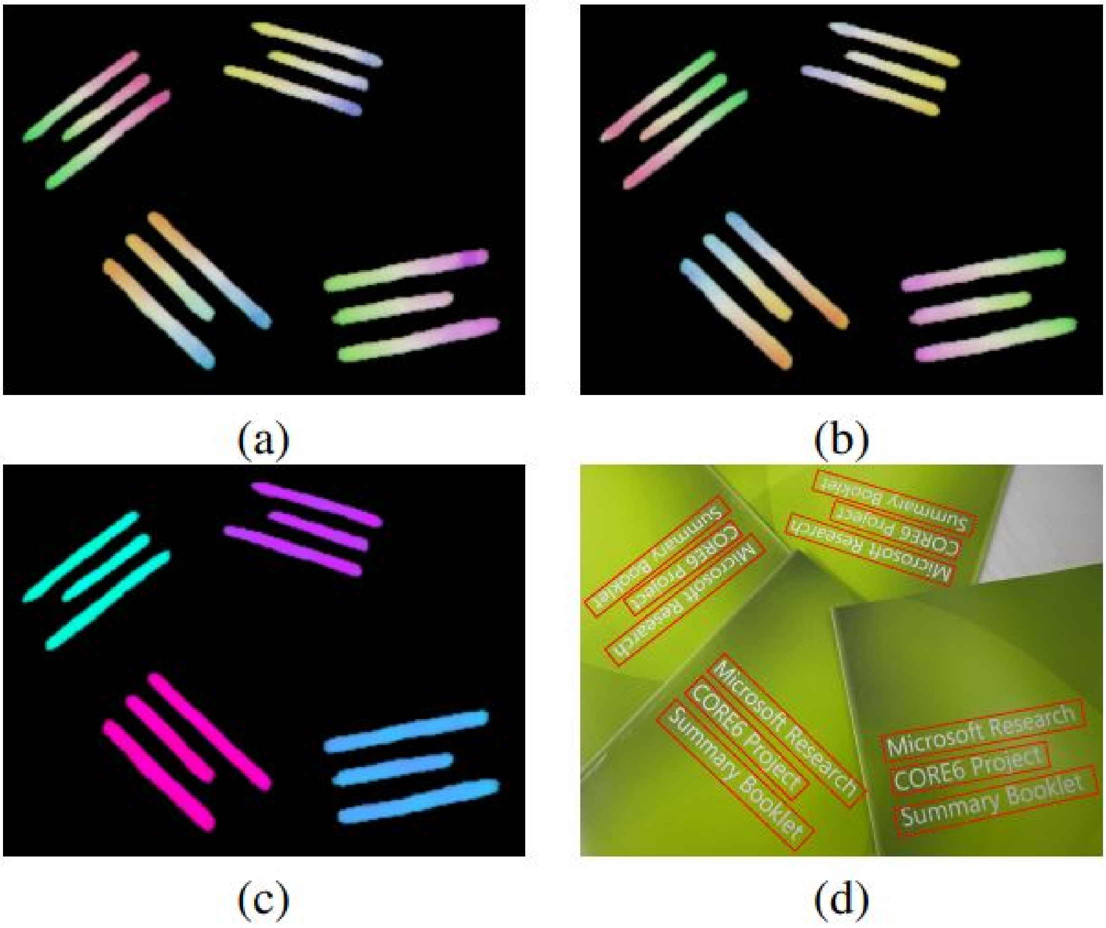


Figure 10: *Intermediate results of the proposed algorithm.*

RCNN

The "Region-Based Convolutional Neural Network," or RCNN, is an early yet prominent method of identifying objects in computer vision. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik first proposed it in their 2014 paper, "Rich feature hierarchies for accurate object detection and semantic segmentation" [18] . First, it selects the possible number of bounding box regions through selective search. Then, for classification, it pulls CNN features from each region. The fundamental goal of RCNN is to take the input image and generate bounding boxes, each of which contains objects as well as their classification. The RCNN model is made up of several basic components:

Selective Search: In RCNN, the Selective Search algorithm is applied to produce region suggestions inside an image. It divides the image into multiple smaller sections, or superpixels, with the goal of locating likely object placements. The method groups those sub-regions into bigger regions, which are suggested as suitable bounding boxes, by combining different similarity criteria, including color, texture, and shape. These suggested regions were not selected at random. These suggestions for regions are potential bounding boxes with a high probability of containing items [19].

Feature Extraction: A fixed-sized image segment is taken out and scaled to a predetermined size for every area proposed. After obtaining the region proposals, RCNN extracts specific characteristics from each region to get the necessary data. The pre-trained CNN model employed in this stage normally demands that each area be expanded to a specified size. AlexNet or VGGNet, which are renowned for their potent feature extraction abilities, are frequently used for the pre-trained CNN architecture [23]. Each region receives the CNN treatment, and the feature vectors produced by this process store the specific characteristics of the content contained in each region, making them ideal for later object recognition.

Classification: A machine learning classifier then receives the features that were retrieved from each region suggestion. The goal of this classifier is to assess whether a given area contains an object. If yes, it determines which category it belongs to. RCNN often uses a multi-class classifier, which represents various object categories [21]. The classifier is developed on a labeled dataset to discover which categories of objects correspond to the extracted features.

Bounding Box Regression: In addition to object classification, RCNN is also capable of simultaneous bounding box regression. The goal of bounding box regression is to improve the locations and dimensions of the bounding boxes generated by the original region recommendations. The accuracy of object localization is increased by RCNN by modifying the coordinates of these boxes. The correction of the bounding box sizes and placements relative to the initial suggestions can be achieved by learning the offsets required (21).

Non-Maximum Suppression (NMS): There could be many bounding boxes that relate to the same object or significantly overlap after object categorization and bounding box regression. RCNN uses the Non-Maximum Suppression (NMS) post-processing stage to provide a complete set of non-overlapping, high-confidence bounding boxes. NMS filters out redundant and overlapping bounding boxes, keeping just the most certain ones. It does this by deleting objects that have a significant amount of overlap with the bounding box that has the greatest confidence score for each detected object [21].

As RCNN depends on the selective search algorithm, it is computationally expensive and slow, which is one of its drawbacks. By integrating the area suggestion generation step into the deep learning pipeline, Faster RCNN improved the speed and accuracy of object detection [23]. The area Proposal Network (RPN) is a neural network feature that can produce area suggestions directly from the common convolutional map features [20]. It was added to Faster R-CNN to significantly enhance the regional proposal generation process. As a result, the entire object detection pipeline became completely trainable and a lot quicker.

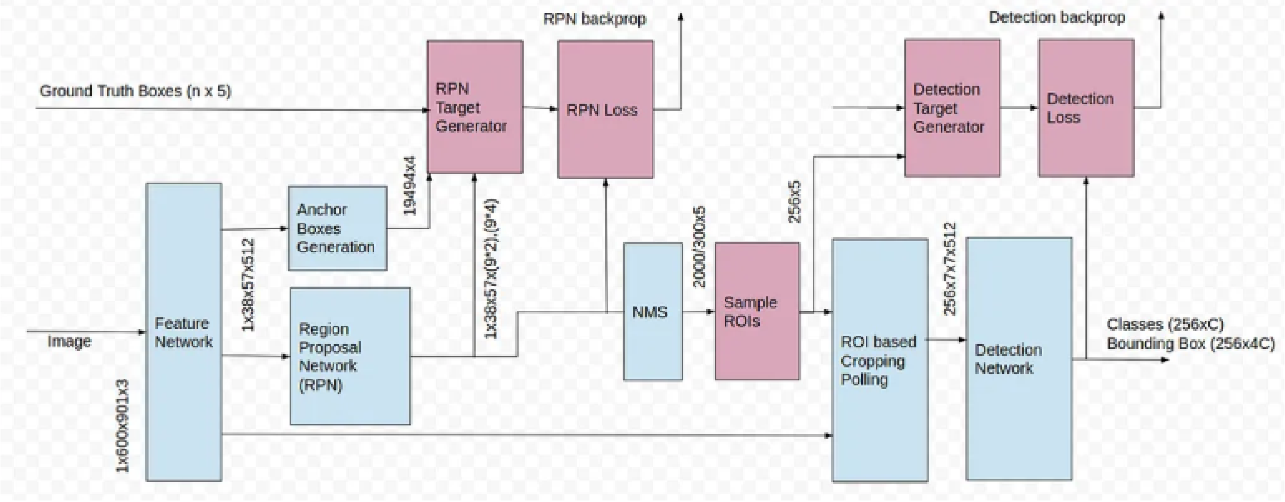


Figure 11: *Diagram of how Faster-RCNN works*

It is made up of five key elements: a deep fully convolutional network, a region proposal network, ROI pooling and fully connected networks, a bounding box regressor, and a classifier [23]. A large number of item candidates have been suggested using deep fully convolutional and region proposal networks, and the candidate regions (proposals) are normalized using the ROI Pooling layer (21). The fully linked layers then collect useful characteristics for classification and regression.

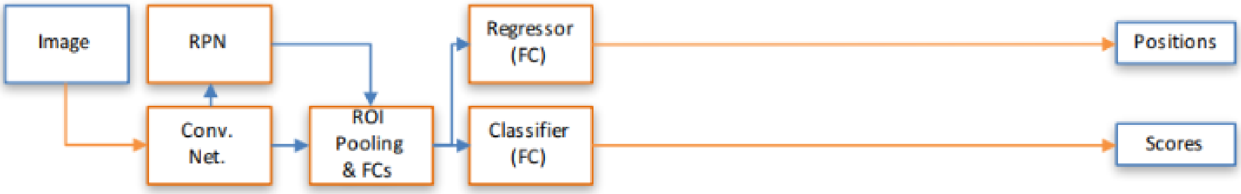


Figure 12: *Faster RCNN*

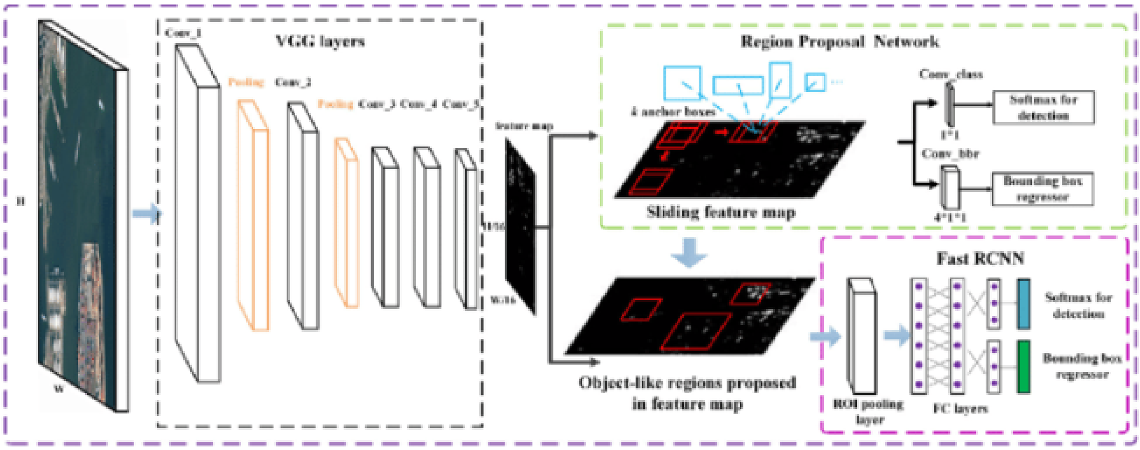


Figure 13: *The architecture of Faster R-CNN.*

Dataset

For the dataset to train our model, we have used the MNIST dataset. MNIST stands for "Modified National Institute of Standards and Technology." It was created to serve as a benchmark dataset for developing and testing machine learning algorithms, particularly those related to image classification and handwritten digit recognition. It contains a total of 70,000 images from which 60,000 are for training and 10,000 for testing. Giving us enough data to work with. Here, each image is 28x28 pixels in size and the images are grayscale so that we can utilize a single channel which indeed decreases our option to identify different color data. But with this step, it greatly improves the detection of every colored data as we preprocess it to our desired channel. The data are balanced and handwritten to remove any kind of bias in terms of occurrence of a specific digit and to include different keystrokes and quirks that humans have when writing. The various cursive, thickness, size and style all are present in order to include all the possibilities that may occur when inspecting a text written by humans. With it being free and widely accessible, it greatly helped us in terms of gathering data with variation with great ease.

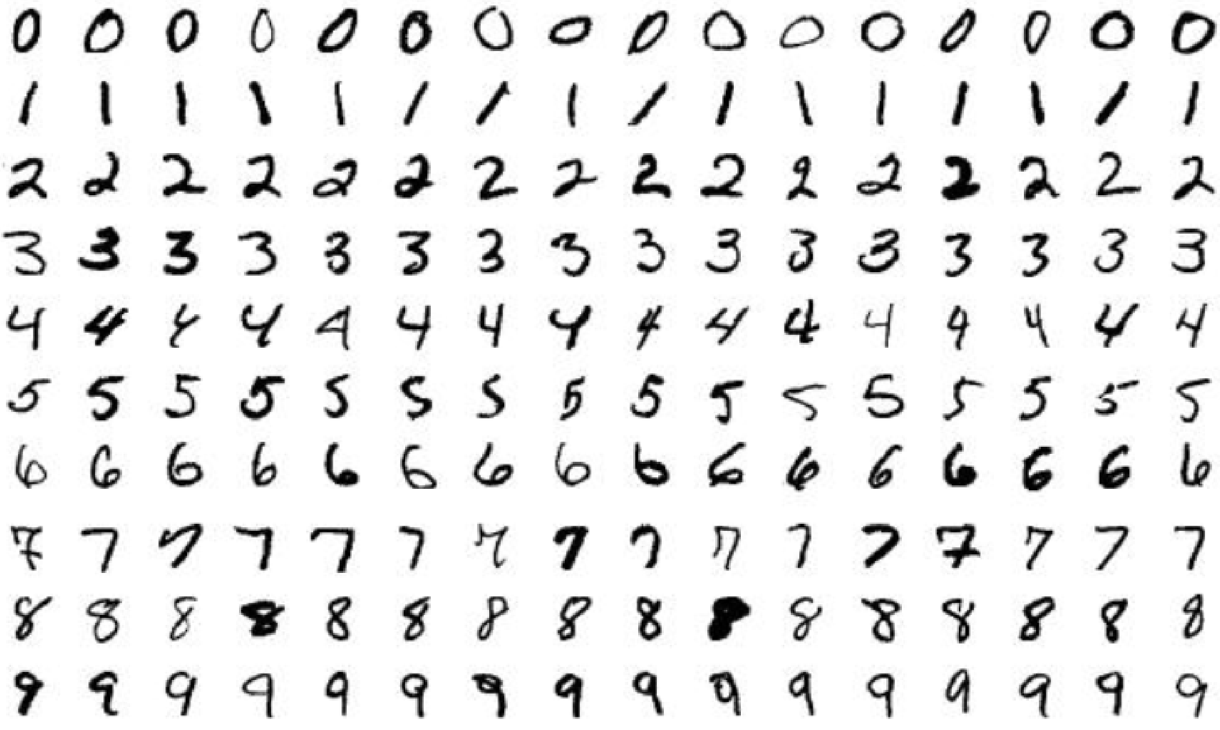


Figure 14: *MNIST dataset representation*

Data pre-process

We have to first make the image size constant. So, we have resized the image to 28*28. Then, we have to ensure the color channels consistency. We will be using grayscale image for training hence we have transformed all our training data image to grayscale version. Lastly, we have set the datatype for all the images to "Float32". When we are testing data or inputting data into the model, we will have to follow the same preprocessing procedure to prepare the image as well.

Preliminary analysis

We used RCNN to recognize the text area and digitize the text. However, though the region proposal has been effective in object detention, it was a different story when we used the same principle to detect text boundaries. Thus we shifted to using the EAST model for detecting text boundaries and CNN to recognize the text. Currently, our model is able to recognize a single individual digit at a time. However, in real world scenarios there will be inputs of variable length and the model has to have the ability to string them together. As we previously mentioned, EAST can be used for detecting text boundaries and for this reason to solve our issue, we can modify the EAST model to detect not the entire string of text but individual letters. In addition to that, we can use another approach where we use LSTM to process variable length strings.

Another issue we faced during our model creation is that while the accuracy of prediction for the data that are similar to the dataset used for training is considerably high, in real world implementation the accuracy took a huge backlash. This is mainly due to the quality of image being lower compared to the training dataset. There is more noise in the image, and pictures tend to get blurred as they are zoomed in on the digits. We are thinking of implementing better preprocessing methods that will improve the input data quality, ultimately increasing accuracy of the models prediction. Lastly, there is the part of implementing our trained model in an android environment. We were able to successfully convert the model into a version that is supported in the android. However, there were some issues regarding the model being successfully adapted into the system. We are planning to utilize other inference for the model in the future.

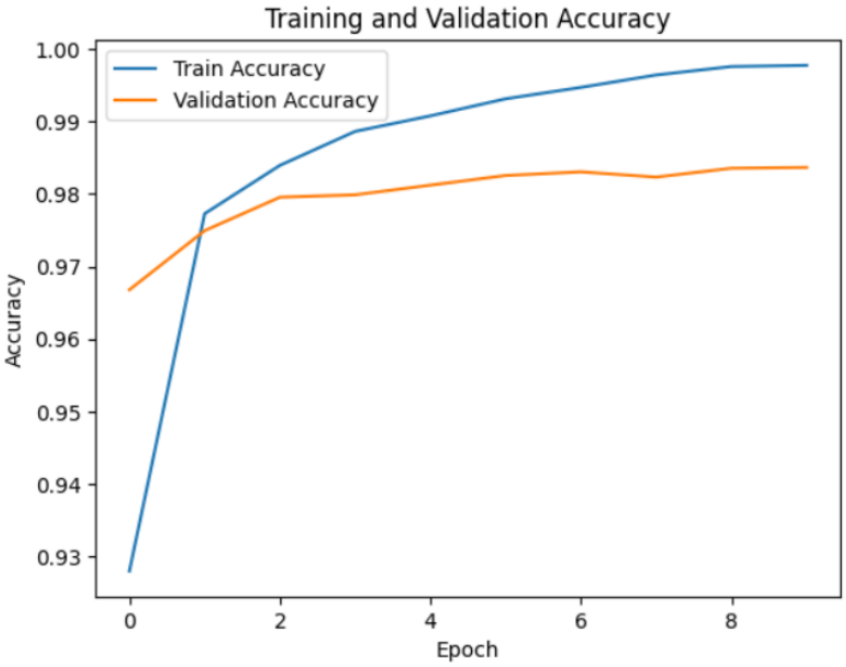


Figure 15: *Model Accuracy Result*

Conclusion

To conclude, integration of ML and OCR in an android environment has great potential as it directly enhances user experience by removing the human labor and makes it more time efficient. In addition, the cross checking feature will rapidly ensure data validity and speed up the process of error handling even more. Then by enabling the processing to occur in the local environment, people can now store data with ease when visiting rural areas with no reception. Furthermore, the model uses user reference to calibrate itself for better results. So, the model can ensure each user gets their own preference in extracting data. And by the fact that everyone in the modern world currently has access to a smartphone, we can obsolete the idea of a different device needed for this type of work in terms of scanning and extracting. So, this research will directly impact the common people in empowering them and help them integrate with the processes of the modern world.

References

[1] Mor, S. S., Solank, S., Gupta, S., Dhingra, S., Jain, M., Saxena, R. (2019, February). HANDWRITTEN TEXT RECOGNITION: with Deep Learning and Android. International Journal of Engineering and Advanced Technology (IJEAT), 8(3S), 7. https://www.ijeat.org/wp-content/uploads/papers/v8i3S/C11730283S19.pdf

[2] Patil, S., Vijayakumar, V., Mahadevkar, S., Athawade, R., Maheshwari, L., Kumbhare, S., Garg, Y., Dharrao, D., Kamat, P., Kotecha, K. (2022). Enhancing Optical Character Recognition on Images with Mixed Text Using Semantic Segmentation. Journal of Sensor and Actuator Networks, 11(4), 63. https://doi.org/10.3390/jsan11040063

[3] Lei, Z., Zhao, S., Song, H., Shen, J. (2018). Scene text recognition using residual convolutional recurrent neural network. Journal of Machine Vision and Applications, 29(5), 861–871. https://doi.org/10.1007/s00138-018-0942-y

[4] Jiang, Y., Jiang, Z., He, L., Chen, S. (2022). Text recognition in natural scenes based on deep learning. Multimedia Tools and Applications, 81(8), 10545–10559. https://doi.org/10.1007/s11042-022-12024-w

[5] Dave, H. (2020). OCR Text Detector and Audio Convertor. https://www.academia.edu/43229530/OCR_Text_Detector_and_Audio_Convertor

[6] Xia, S., Kou, J., Liu, N., Yin, T. (2022). Scene text recognition based on two-stage attention and multi-branch feature fusion module. Applied Intelligence. https://doi.org/10.1007/s10489-022-04241-5

[7] Panchal, B. Y., Chauhan, G., Panchal, S. R., Chaudhari, U. M. (2022). An investigation on feature and text extraction from images using image recognition in Android. Materials Today: Proceedings, 51, 798–802. https://doi.org/10.1016/j.matpr.2021.06.237

[8] U, A., S, S. U., Rodrigues, A. (2020, May). Text Localization and Recognition. International Journal for Research in Applied Science Engineering Technology (IJRASET), 8(V May 2020). www.ijraset.com

[9] E, J., Tejaswini, K., Chintalapati, L., Shafiulla, M. M. (2020, May). Text Extraction from Images Using OCR. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 8(V). www.ijraset.com

[10] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, “EAST: An efficient and accurate scene text detector,” in Proc. CVPR, 2017, pp. 2642–2651.

[11] L. Shangbang, H. Xin, Y. Cong, ”Scene Text Detection and Recognition: The Deep Learning Era” , 2018/11/10

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading Text in the Wild with Convolutional Neural Networks. International Journal of Computer Vision, 116(1):1– 20, jan 2016

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer International Publishing, 2015.

[14] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. PVANET: deep but lightweight neural networks for real-time object detection. arXiv:1608.08021, 2016.

[15] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In Proc. of CVPR, 2015.

[16] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002, 2016.

[17] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In the European Conference on Computer Vision, pages 56–72. Springer, 2016.

[18] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2013, November). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation | Request PDF. ResearchGate. Retrieved September 16, 2023, from https://www.researchgate.net/publication/258374356_Rich_Feature_Hierarchies_for_Accurate_Object_Detection_and_Semantic_Segmentation

[19] Zhong, Z., Sun, L., Huo, Q. An anchor-free region proposal network for Faster R-CNN-based text detection approaches, IJDAR 22, 315–327 (2019). https://doi.org/10.1007/s10032-019-00335-y

[20] Mahmood, T., Arsalan, M., Owais, M., Lee, M. B., Park, K. R. (2020, March 10). Artificial Intelligence-Based Mitosis Detection in Breast Cancer Histopathology Images Using Faster R-CNN and Deep CNNs Journal of Clinical Medicine https://www.mdpi.com/2077-0383/9/3/749

[21] Sri, M. S., Naik, B. R., Sanker, K. J. (2021, February) Object Detection Based on Faster R-CNN. International Journal of Engineering and Advanced Technology (IJEAT), 10(3). 2249-8958

[22] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, 2016: Region-Based Convolutional Networks for Accurate Object Detection and Segmentation, IEEETransactions Transactions on Pattern Analysis And Machine Intelligence, 38 (1), 142–58

[23] Roh, M.-C., Corp, K., Lee, J. (2017, May 8). Refining Faster-RCNN for Accurate Object Detection. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA).