Mini Project Report on
Text Recognition android app

Submitted in partial fulfilment of the required for the award of the degree of

Bachelors of Technology in Computer Science & Engineering

Submitted by:

Name: Shreyansh Shukla

University Roll No: 2119210

Submitted To:

Dr. Gunjan Chabra

(Class Coordinator Sec L)

Graphic Era Hill University

(Department of Computer Science & Engineering)

Dehradun Campus , Uttrakhand

ACKNOWLEDGEMENT

The success and final outcome of this min project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my mini project. All that I have done is only due tosuch supervision and assistance and I would not forget to thank them.

I respect and thank Dr. Gunjan Chabra , for providing me an opportunity to do the project work in "Text Recognition Android app " and giving us all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teachingstaffs of Dept of environment which helped us in successfully completing our mini project work.

Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Shreyansh Shukla

Date:

22/07/2023

Table of Content

Introduction

Optical Character Recognition (OCR) defines the process of mechanically or electronically converting scanned images of handwritten, typed, or printed text into machine-encoded text. Think of it as the process of turning analog data, digital.

Here the term used ‘optical character Recognition’(OCR) is also known as Text Recognition.

As OCR stands for optical character recognition, OCR technology deals with the problem of recognizing all kinds of different characters. Both handwritten and printed characters can be recognized and converted into a machine-readable, digital data format.

Think of any kind of serial number or code consisting of numbers and letters that you need digitized. By using OCR you can transform these codes into a digital output. The technology makes use of many different techniques. Put simply, the image taken is processed, the characters extracted, and are then recognized.

What OCR does not do is consider the actual nature of the object that you want to scan. It simply “takes a look” at the characters that you aim to transform into a digital format. For example, if you scan a word it will learn and recognize the letters, but not the meaning of the word.

Now OCR is used in creating handwritten text recognizer whose function is to recognize texts from any handwritten text on any surface either digital or in hard copy format.

All of this is possible due to Google’s API & Text Recognition ML kit.

How does Optical Character Recognition Work?

Let’s have a look at three basic steps of optical character recognition: image pre-processing; character recognition; and the post-processing of the output.

Step 1: Image Pre-Processing in OCR

OCR software often pre-processes images to improve the chances of successful recognition. The aim of image pre-processing is an improvement of the actual image data. In this way, unwanted distortions are suppressed and specific image features are enhanced. These two processes are important for the following steps.

Step 2: Character Recognition in OCR

Character Recognition of License Plates

For the actual character recognition, it is important to understand what “feature extraction” is. When the input data is too large to be processed, only a reduced set of features is selected. The features selected are expected to be the important ones while those that are suspected to be redundant are ignored. By using the reduced set of data instead of the initial large one, the performance is increased.

For the process of OCR, this is important as the algorithm has to detect specific portions or shapes of a digitized image or video stream.

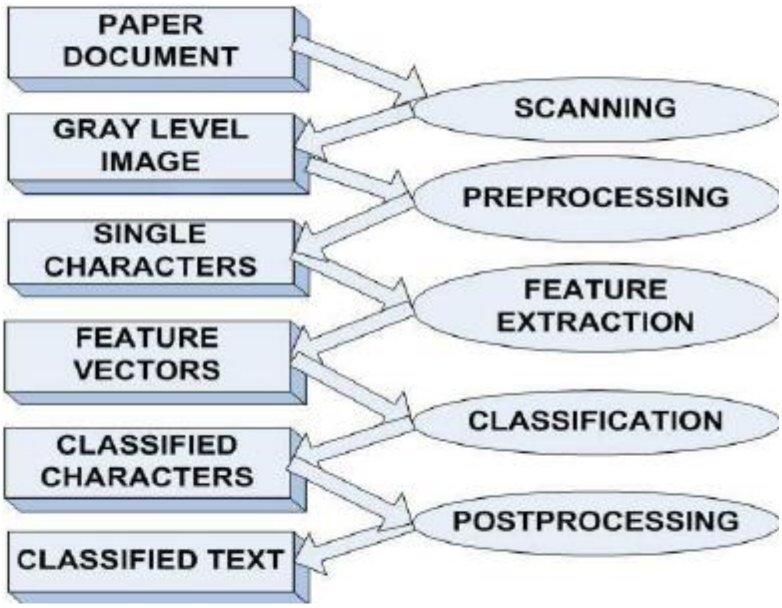
Step 3: Post-Processing in OCR

Post-processing is another error correction technique that ensures the high accuracy of OCR. The accuracy can be further improved if the output is restricted by a lexicon. That way, the algorithm can fall back to a list of words that are allowed to occur in the scanned document for example.

OCR is not only used to identify proper words but can also read numbers and codes. This is useful for identifying long strings of numbers and letters, such as serial numbers used in many industries.

To better deal with different types of input OCR, some providers started to develop specific OCR systems. These systems are able to deal with the special images, and to improve the recognition accuracy, even more, they combined various optimization techniques.

For example, they used business rules, standard expressions, or rich information contained in the color image. This strategy of merging various optimization techniques is called “application-oriented OCR” or “customized OCR”. It is used in applications such as business card OCR, invoice OCR, and ID card OCR.



Use Cases for OCR Technology

The possibilities for using optical character recognition software is widespread as OCR can be combined with a broad range of technologies. Here are a few examples of possible use cases including OCR software:

1. Identification Processes in OCR

Machine Readable Zone (MRZ) in a Passport

Passports and IDs have a machine-readable zone (MRZ) that can be scanned. OCR can speed up the process of identifying and registering people. This is useful for security forces at borders or other checkpoints. It can also be used for commercial purposes to increase customer engagement, such as the check-in process within hotels, or the registration process with banks and other businesses.

2. Marketing Campaigns with OCR

Leading brands are making use of OCR to run innovative and engaging campaigns to drive engagement with their customers. Think of all the voucher codes that customers can redeem by typing them in. Or numbers printed on the inside of a bottle cap that you need to collect.

Take a look at how PepsiCo uses OCR for

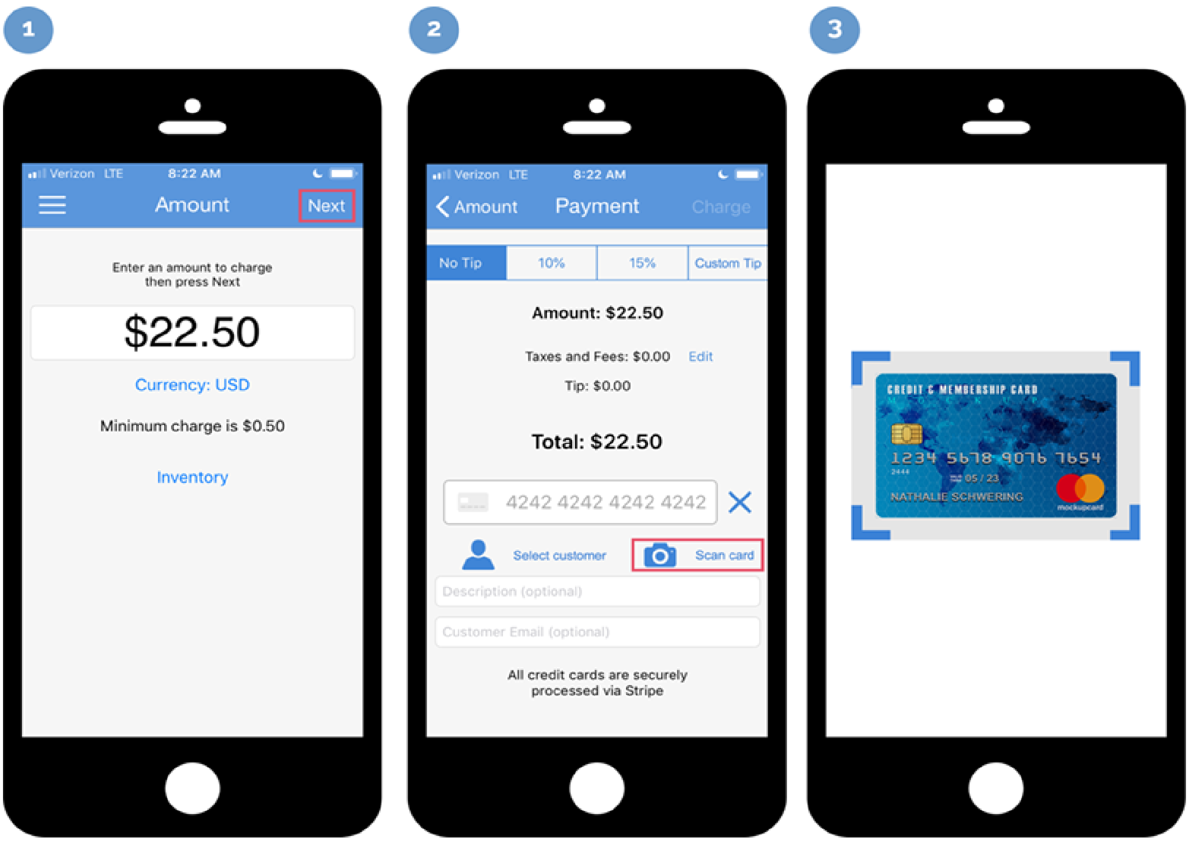
3. Scanning Documents

Anyone can convert large set of documentation into a digital format without much efforts. This is one of the most benefited use of this technology for common people.

4. OCR in Payment Processes

IBAN Scanning with OCR

The International Bank Account Number (IBAN) serves to identify bank accounts across borders. The IBAN may come in different lengths and can consist of numbers as well as letters. In case someone has banking details in handwritten format he can easily scan that and use payment process efficiently.



Limitation of the Project.

THERE ARE SEVERAL LIMITATIONS OF TEXT RECOGNITION APPLICATION:

1.**IMAGE QUALITY**: THE QUALITY OF THE IMAGE CAN GREATLY IMPACT THE ACCURACY OF TEXT RECOGNITION. POOR LIGHTING, BLURRINESS, OR PERSPECTIVE DISTORTION CAN MAKE IT DIFFICULT FOR THE ALGORITHM TO ACCURATELY RECOGNIZE THE TEXT.

2.**FONT VARIABILITY**: TEXT RECOGNITION ALGORITHMS MAY STRUGGLE TO ACCURATELY RECOGNIZE TEXT WRITTEN IN DIFFERENT FONTS STYLES, SIZES, AND ORIENTATIONS. THIS CAN MAKE IT DIFFICULT TO RECOGNIZE TEXT IN DIFFERENT LANGUAGES AND SCRIPTS.

3.**COMPLEX BACKGROUNDS**: COMPLEX BACKGROUNDS, SUCH AS PATTERNS OR IMAGES, CAN MAKE IT DIFFICULT FOR THE ALGORITHM TO ACCURATELY DIFFERENTIATE BETWEEN THE TEXT AND THE BACKGROUND.

4.**HANDWRITTEN TEXT**: RECOGNIZING HANDWRITTEN TEXT IS A MORE CHALLENGING TASK COMPARED TO RECOGNIZING PRINTED TEXT. HANDWRITING CAN VARY GREATLY FROM PERSON TO PERSON AND CAN BE DIFFICULT FOR THE ALGORITHM TO ACCURATELY RECOGNIZE.

5.**LANGUAGE AND SCRIPTS**: TEXT RECOGNITION ALGORITHMS MAY NOT BE ABLE TO ACCURATELY RECOGNIZE TEXT IN DIFFERENT LANGUAGES AND SCRIPTS, PARTICULARLY THOSE THAT USE NON-LATIN CHARACTERS.

6.**REAL-TIME PROCESSING**: TEXT RECOGNITION ALGORITHMS CAN BE COMPUTATIONALLY INTENSIVE, MAKING IT CHALLENGING TO PROCESS LARGE AMOUNTS OF TEXT IN REAL-TIME.

Font family	Font group	Font style	Point size	Word (original size)	Word (close-up size)	Recognized text
Georgia	Serif	Plain	9 pts.	band	band	Inn::!
Georgia	Serif	Plain	8 pts.	fright	fright	*tri/ht*
Verdana	Sans serif	Italics	9 pts.	*band*	*band*	-
Verdana	Sans serif	Italics	8 pts.	*pack*	*pack*	-
Times New Roman	Serif	Plain	9 pts.	band	band	band

The Limitation mentioned above are briefly explained by the above image. Now if we take a closer look to the image we will notice that many things effects for an OCR to scan properly and mention the result accordingly. We can also observe that even in case of same font family but different font group and point size the result is different as well as we can also observe that OCR is not recognizing the texts which are in italic font size. The are some brief observations of the experiment conducted on OCR.

Now in case of handwritten text recognition I personally have face many issues except these above mentioned ones. The OCR system created by me with the help of GOOGLE's API only support recognition in case of capital letters and ideal condition of the image, all this can be improved if given some time.

Code for Capture Image button

```
package com.example.textdetectot;


import androidx.appcompat.app.AppCompatActivity;


import android.content.Intent;
import android.widget.Button;
import android.view.View;
import android.os.Bundle;


public class MainActivity extends AppCompatActivity {


    private Button captureBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        captureBtn = findViewById(R.id.idBtncapture);
        captureBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,ScannerActivity.class);
                startActivity(i);


            }
        });
    }
}
```

Code for capturing image and detecting text from image

```
package com.example.textdetectot;


import static android.Manifest.permission.CAMERA;


import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.annotation.NonNull;


import android.content.Intent;
import android.graphics.Point;
import android.graphics.Rect;
import android.os.Bundle;
import android.provider.MediaStore;
import android.content.pm.PackageManager;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.graphics.Bitmap;
import android.widget.TextView;
import android.widget.ImageView;


import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.mlkit.vision.common.InputImage;
```

```
import com.google.android.gms.tasks.Task;
import com.google.mlkit.vision.text.Text;
import com.google.mlkit.vision.text.TextRecognizer;
import com.google.mlkit.vision.text.TextRecognition;
import com.google.mlkit.vision.text.latin.TextRecognizerOptions;

public class ScannerActivity extends AppCompatActivity {

    private ImageView captureIV;
    private Button snapBtn,detBtn;
    private TextView resTV;
    private Bitmap imgBitmap;
    static final int REQ_IMG_CAPTURE = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scanner);
        captureIV=findViewById(R.id.idIVCaptureImage);
        resTV=findViewById(R.id.idTVDetectedText);
        snapBtn=findViewById(R.id.idBtnSnap);
        detBtn=findViewById(R.id.idBtnDet);

        detBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                detText();
            }
        });
private void reqPerm(){
```

```
        int PERM_CODE =200;
        ActivityCompat.requestPermissions(this,new String[]{CAMERA},PERM_CODE);
    }

    private boolean checkperm(){
        int camoerm= ContextCompat.checkSelfPermission(getApplicationContext(),CAMERA);
        return camoerm == PackageManager.PERMISSION_GRANTED;
    }

        snapBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(checkperm()){
                    captureImg();
                }
                else{
                    reqPerm();
                }
            }
        });

    }

      private void captureImg(){
        Intent takePic = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if(takePic.resolveActivity(getPackageManager())!=null){
            startActivityForResult(takePic,REQ_IMG_CAPTURE);
        }
    }
```

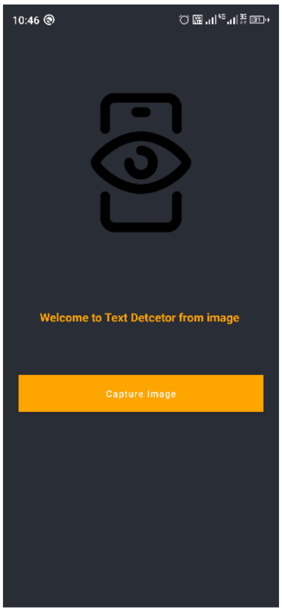
```
 @Override
   public void onRequestPermissionsResult(int requestCode, String[] permissions,@NonNull
int[] grantResults) {
       super.onRequestPermissionsResult(requestCode, permissions, grantResults);
       if(grantResults.length>0){
           boolean camPermission = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
           if(camPermission){
               Toast.makeText(this, "Permission Granted",Toast.LENGTH_SHORT).show();
               captureImg();
           }
           else{
               Toast.makeText(this, "Permission Denied", Toast.LENGTH_SHORT).show();
           }
       }
   }
 @Override
   protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
       super.onActivityResult(requestCode, resultCode, data);
       if(requestCode == REQ_IMG_CAPTURE && resultCode == RESULT_OK){
           Bundle extras = data.getExtras();
           imgBitmap=(Bitmap) extras.get("data");
           captureIV.setImageBitmap(imgBitmap);
       }
   }
   private void detText(){
         InputImage image = InputImage.fromBitmap(imgBitmap,0);
       TextRecognizer recognizer =
TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);
       Task<Text> res = recognizer.process(image).addOnSuccessListener(new
OnSuccessListener<Text>() {
```

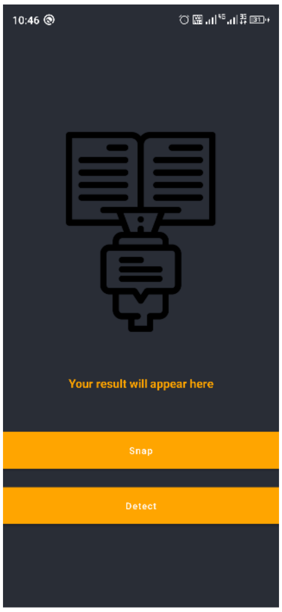
```
        @Override
        public void onSuccess(Text text) {
            StringBuilder res = new StringBuilder();
            for(Text.TextBlock block: text.getTextBlocks()){
                String blockTest = block.getText();
                Point[] blockCorPoint = block.getCornerPoints();
                Rect blockFrme = block.getBoundingBox();
                for(Text.Line l: block.getLines()){
                    String lText = l.getText();
                    Point[] lcornerPoint = l.getCornerPoints();
                    Rect lRect =  l.getBoundingBox();
                    for(Text.Element elm: l.getElements()){
                        String elmText = elm.getText();
                        res.append(elmText);
                    }
                    resTV.setText(blockTest);
                }
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(ScannerActivity.this, "Fail to detect text from
image"+e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
  }
}
```

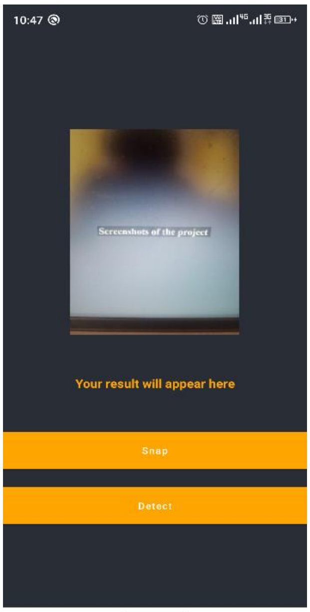
Screenshots of the project

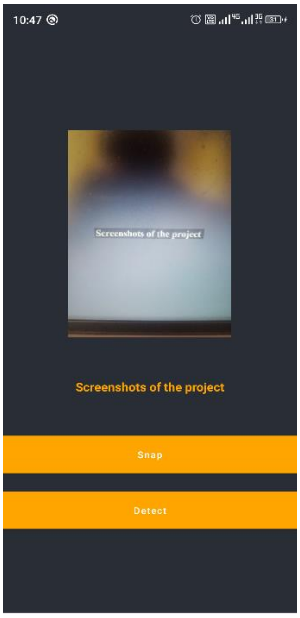
Application's Icon:

Application's Interface:





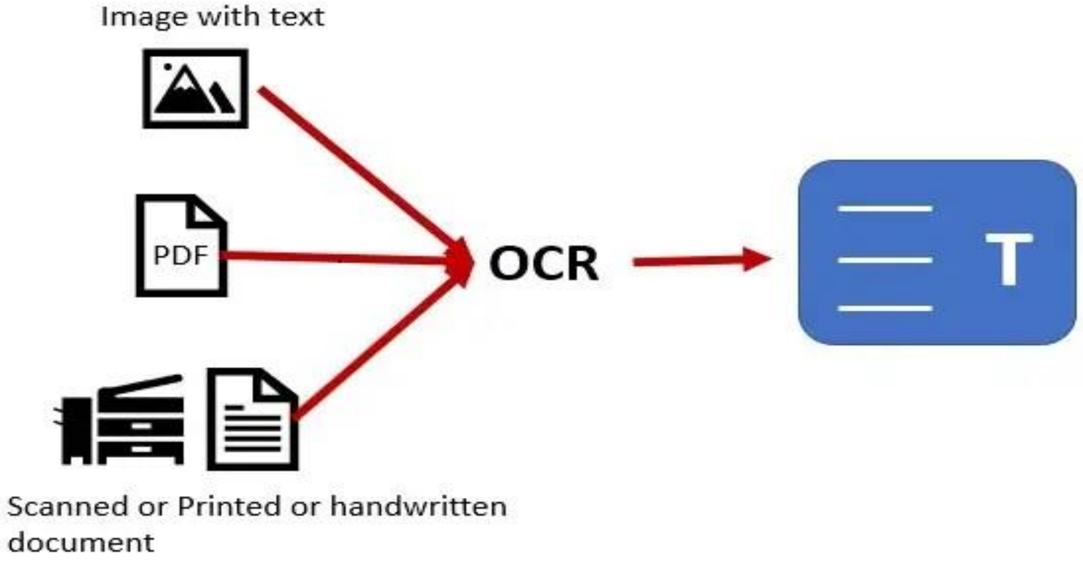




Conclusion

In conclusion, text recognition is a crucial technology for automating and streamlining various processes in industries such as finance, healthcare, and logistics. With the development of artificial intelligence and machine learning, the accuracy and efficiency of text recognition has improved significantly, making it a valuable tool for extracting information from images here which is extracting text from handwritten text. The technology has various applications, including payment processing, data entry, and record keeping, and it is likely to play an even larger role in automation and digital transformation in the future.

In conclusion, text recognition is an important technology that has numerous applications in various industries, particularly in the payment process. It allows for the automatic extraction of text information from images, which can save time and improve accuracy in tasks such as payment verification, data entry, and record keeping. There are various methods of text recognition, including Optical Character Recognition (OCR), Handwritten Text Recognition (HTR), and Intelligent Character Recognition (ICR). The accuracy of text recognition depends on several factors, including image quality, type of text, language, and font. With the ongoing developments in artificial intelligence and machine learning, the accuracy and efficiency of text recognition is expected to continue to improve in the future.



Reference

1. www.google.com
2. www.youtube.com
3. www.geeksforgeeks.org
4. OCR with https://anyline.com
5. "Optical Character Recognition (OCR) - A Survey" by A. Rosca, S. F. Chang, and C. C. Loy. Published in arXiv preprint arXiv:1707.06651, 2017.
6. "Handwritten Text Recognition (HTR) systems" by N. T. Vu and D. T. Nguyen. Published in Journal of Ambient Intelligence and Humanized Computing, Vol. 12, No. 4, pp. 4291-4309, 2021.
7. "Intelligent Character Recognition (ICR): A Comprehensive Review" by A. K. Jain and R. K. Choudhary. Published in Pattern Recognition, Vol. 28, No. 7, pp. 977-997, 1995.
8. "Deep Learning for Text Recognition in Images" by X. Bai, Y. Zhou, and J. Yang. Published in IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 6, pp. 1230-1244, 2017.
9. https://chat.openai.com

10. Nguyen, L., Chen, Y., & Hua, X. (2017). Handwritten text recognition (HTR) systems: A comprehensive survey. International Journal on Document Analysis and Recognition, 20(2), 91-107.

11. Wang, Y., & Chng, E. (2012). An overview of OCR technologies. International Journal of Computer Science and Network Security, 12(1), 1-7.

12. Smith, J. R. (1995). A review of optical character recognition (OCR) techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, 25(2), 356-370.
13. Almazan, J., Louloudis, G., & Karkanis, S. A. (2014). Handwritten text recognition: A review. arXiv preprint arXiv:1406.5298.
14. Buonora, G. (2018). Text recognition technologies and their applications. Proceedings of the 2018 International Conference on Computer Vision Theory and Applications (VISAPP), 129-136.
15. Jain, A. K., Ross, A., Nandakumar, K., & Niyogi, P. (2016). Introduction to biometric recognition. Springer.