Text Recognition from Images Using Optical Character Recognition

JILLA KIRTHAN
Computer Science and Engineering
SR UNIVERSITY
Hanamkonda,India
2203a52151@sru.edu.in

Abstract: *Optical Character Recognition, or OCR, is a technology that scans text images in order to fully recognize the alphabetic and numeric contents of printed or handwritten letters. It works by scanning character images and converting them into editable text. First, it divides the input text image into regional areas in which each line is separated and text isolated into characters, including whitespaces. Following character segmentation, several attributes are retrieved from the characters, with a focus on texture and topological qualities. These elements include corner points, regional traits, and ratios such as the character area to the convex area. This extraction process is crucial for correctly recognizing each character's unique properties.*

The OCR system is based on a pre-stored library of templates for every capital and lowercase letter, number, and common symbol. These templates are developed using the topological characteristics and texturing of characters. The system uses feature matching techniques to compare the extracted characteristics of the input characters with these templates during the recognition phase. Character identification is based on the degree of similarity between the retrieved character characteristics and the template traits. This method makes use of the exact geometric and textural differences between characters to guarantee reliable identification.

OCR systems have broad uses for digitizing printed and written documents, hence improving access to data, and in the modification of text content without any effort. With the aid of OCR, the laborious task of manual transcription is significantly reduced as it becomes more efficient and accurate when converting physical documents into a digital format. Using texture and topological features improves character differentiation, thereby making it robust against variations in fonts and styles. This capability shows why OCR is relevant for document digitization, automated data entry, and assistive technologies for the visually impaired.

Keywords: *Optical Character Recognition, Basic OCR, Reading Text, Photo analysis, Error Analysis, Truth Verification, Sequence Matcher, text overlaying, Python Programming Language, Computer Vision, Image enhancement techniques, Character reading, Deep Learning, Model Validation, Text localization.*

I. INTRODUCTION

OCR, often called optical character recognition, is the term used when referring to the conversion of images of text or documents into an editable format through scanning. The mechanism through which this technology operates is referred to as an optical mechanism. For a person, there is also the optical component and it is in the human being's eyes. The eyes send a picture to the brain. Similar to man’s ability of reading, this is what OCR technology can do: Optical character recognition. Human beings possess the skill of reading at an advanced level, while the systems based on OCR capabilities are relatively basic. Today, OCR technology can help you process any type of document by extracting text content from printed pages, scanned images, or photographs. One of the most common usages of OCR is also in the banking domain, where it processes demand drafts or cheques without the need for people to do the work. This can be photographed through a mobile camera. Now, the handwriting on it gets scanned, and the exact amount of money gets transferred to the person's account. Generally, this technology works great when demand drafts or cheques are printed; however, in cases of handwritten demand drafts or cerussite is quite accurate as well but verifies the signature. Nowadays, many people in the legal field try to convert paper documents to digital files. This helps save space as well as avoid the hassle of checking through boxes of papers. This happens because documents are scanned. This process is made easier by OCR, which converts the documents into searchable text for easier access from the database. Law attorneys and other legal workers can just quickly and easily get an enormous collection of documents in electronic form by merely typing a few keywords. OCR is used extensively in most areas, like education, finance, and even offices. This paper recommends a good method for image text recognition using features of texture, topologies. Then, based on the extracted features of the character, the character is completed by matching its surface and topological information.

Optical Character Recognition is technology transformation that translates pictures featuring printed, handwritten, or type written text into a form amenable to machine-read and, most importantly, editable. An optical interpretation system is used-just like how humans read in terms of optical sensors human eyes and the brain processing the visual interpretation. OCR technology involves intricate character detection algorithms within pictures/scanned documents to change them over to searchable and editable types. It has become a vital component in digitizing and automating text extraction operations, where it improves productivity and reduces human labour. In several ways, OCR still lags behind the level of human reading both in terms of accuracy and contextual comprehension.

As of today, OCR plays an imperative role across various industries for the very reason that it can deal with the scanned paper files or PDFs as well as other digital photographs. One prime application in that regard comes from the banking sector: Demand drafts or cheques that are received often undergo scanning through OCR systems. By capturing the picture of a cheque through mobile camera, the writing in it gets scanned and interpreted and the same is used for completion of transaction without human interaction. Thus, it helps in saving time and also achieving more accurate financial operations. In respect of printed papers, OCR has almost achieved perfect digitization. Hand-written documents require more effort; however, they are properly processed with the help of supporting technologies such as verification of signatory.

In addition to banking, OCR has transformed the business of law, enabling the process of digitizing paper documents that lawyers traditionally had to wrestle with. Traditionally, lawyers have been swimming in stacks of paper. They would spend hours upon hours searching through this lot to find a single scrap of information. The translation of these documents into a digital form makes text search inside digital archives many times easier and accessible. Legal practitioners can search for any document or information by just typing keywords, making processes faster and case preparation better. OCR extracts properties such as texture and topology from letters in text pictures.

II. PREVIOUS WORK

Some techniques to search text in images and video have been proposed. They can be further categorized into two categories, namely, those connected component-based methods and the others utilizing a texture-based method. The first group makes use of connected component analysis whereby it is the checking of shape, arrangement, and distribution of edges and edges or similar color and gray parts that constitute letters. The second class of approaches interprets texts as areas with different textural features. Such features are the character parts salient against the background and with a regular pattern of intensity from side to side in that the characters are aligned horizontally.

For texture segmentation in the work of Arif et al 2006, are based on the second order derivatives of Gaussian filters and several non-linear transformations. [1], which proposes an automatic text extraction system. In order to categorize the filtered images into text and non-text pixels, features are then calculated to create a feature vector for each pixel. Text sections are automatically located in [2] using texture analysis techniques like as spatial variance and Gabor filtering. In [3], a novel method for color based on color clustering quantization and bit dropping is proposed.
The input image is mixed together with several foreground images and background images by a multi value image decomposition technique. The authors of this work, [4] offered a technique that was based on the fact that each color's LCQ Local Color Quantization was carried out on its own. The authors assume that all colors can be text colors although not all of them actually are text colors. Lichtenstein [5] demonstrated a technique for structural image enhancement that only required the red color of the RGB color space to create maximum contrast regions important to most text colors. They carry out a convolution technique with appropriate masks for edge detection and image enhancement. Diao [6] provides the description of the method developed for eight connected components analysis of binary images is achieved by being the summation of local edged maps which have been produced through the application of band Derice filter onto every color.

In [7], it is reported that business card images can recognize scripts. In [8], a novel method for video text detection is described. Numerous studies on mobile OCR systems have been located. Camera-based mobile OCR systems for camera phones were introduced by Motorola China Research Center in [9]. To determine the skew angle, a down sample of a business card image is first taken. After that, the text regions are binarized after being skew corrected by that angle. After being divided into lines and characters, these text segments are sent to an OCR engine for identification. The OCR engine is a two-layer classifier that uses templates. For images of business cards with mixed script in Chinese and English, a comparable system is provided in [10]. An outline of a Kanji OCR prototype translating them into English is proposed in for identifying Japanese text printed on a machine and [11]. [12] presents an approach to character recognition systems for Chinese scripts. [13] devises a system based on the use of only uppercase letters of the Latin alphabet. At the beginning, the distorted image is restored by finding the row with the highest number of contiguous white cells and maximizing the appropriate alignment criterion. Then the image is the resulting image can be thought of as one which selects the largest number of consecutive white cells while fulfilling the features of the alignment criterion to the fullest extent. further processed in terms of features by Xa-Ya um fractal decomposition and their recognition was performed by a set of features against the boundary using Manhattan distance. However, this work is limited to the English capital letters, and the only accuracy achieved is empirical and is not suitable for practicing applications. In addition, the scope of their research in creating systems of OCR for mobile devices is larger than document images. [14] read an LCD/LED panel using a camera phone. [15] deals with the problem of text/graphics separation for the images of business cards taken with mobile devices. [16] describes an operational algorithm for skew correction of camera-based business card pictures. The segmentation of business card images captured using a camera for mean mobile device is proposed in [17]. Many of the languages do continues to confront hardships about the optical character recognition.

III. IMPLEMENTATION

Optical character recognition or OCR is a process that takes an image of a text and produces a text with editable characters. The major processes involved in an OCR system include pre-processing, feature extraction, training of the features, and the matching of features. The flow chart for the OCR in figure 1 should be viewed last. This practice entails two data sets: one meant for the training and the other meant for the testing. For both types of data, Preprocessing and Feature Extraction stages are completed. The end goal is met by matching test data features with the training data features.

Input image → image pre-processing → text detection → text recognition → text restructuring → output text

Figure 1: Flowchart of the OCR system processing

1.Input Image: The input image contains the text that has to be read.

2.Image Pre-processing:

The step previews the image for further processing by performing the above practices:

Noise cancellation, Contrast enhancement

Binarization, which means converting into black and white format Skew correction

3. Text Detection:

This line indicates the areas of the image that are showing text activity.

Algorithms like connected component analysis or machine learning-based approaches are applied in order to identify text blocks.

4. Text Recognition:

This is a simple level in which the text chunks identified are assigned as separate characters.

Such techniques are template matching methods, statistical models, or even more robust models, such as CNN and RNNs.

5. Text Restructuring:

This step rephrases the text in question in a more readable form, for instance: Formatting the text into paragraphs or lines

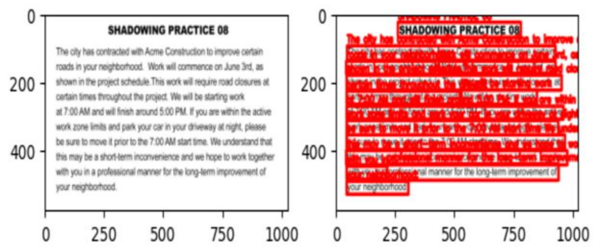
Creating the line breaks and spaces. Rectifying any inaccuracies present in the identified text.

6.Output Text:

This is the output at the end of the OCR process, which is text recognized from an input image.

A. Pre-processing

The text picture is converted into a binary format for further processing, as seen in Figures 2. The colour or grayscale picture is transformed into a binary image with pixel values of 0 or 1 throughout this procedure. This makes managing picture data easier. The binary picture is modified in Figure 4 such that the letters or foreground pixels are given the value 1 white, while the background pixels are given the value 0 black. This change optimizes the text for further uses by guaranteeing a clear separation of the text from the backdrop.



SHADOWING PRACTICE 08

The city has contracted with Acme Construction to improve certain roads in your neighborhood. Work will commence on June 3rd, as shown in the project schedule. This work will require road closures at certain times throughout the project. We will be starting work at 7:00 AM and will finish around 5:00 PM. If you are within the active work zone limits and park your car in your driveway at night, please be sure to move it prior to the 7:00 AM start time. We understand that this may be a short-term inconvenience and we hope to work together with you in a professional manner for the long-term improvement of your neighborhood.

Figure 2 : text to characters output

The text image has now been separated into individual text lines. This is accomplished by calculating the sum of all values in a row. When the sum equals zero, a new line is identified and separated. The sum of all rows between two lines should equal zero. The image is divided into lines, and each line is extracted one by one, as illustrated in Figure 4. The procedure is repeated until all lines have been extracted.

Figure 3: complimented text Image

Figure 4: Extracted line

Single lines are extracted because dealing with one line is simpler than dealing with an entire image. Again, the letters from each line must be extracted, as shown in Figures 5 and 6. This is accomplished by adding all of the values in a given column. When the sum equals zero, a character is identified and separated. This separates all individual character alphabets, digits, and punctuation

Figure 5: a text line image

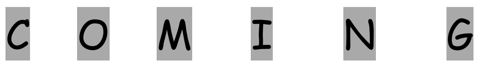


Figure 6: text extracted character

B. Features Extraction

Feature extraction is the critical component in text recognition systems. Within the scope of this OCR-oriented project, Easy OCR was adopted in order to obtain helpful features from the textual images. This process starts by opening the picture with OpenCV and changing the model into an RGB one. In Easy OCR, neural networks are employed to automatically detect specific characters in an image. The features which have been extracted include the rudimentary text recovered, the boxes that enclose the text discovered and the probability values associated with varying areas of the bounding box. These characteristics depict the location and meaning of letters in the picture.

Boxes containing letters encircle the text regions with coordinates and the characters recognized in the region are associated with probabilities called score. This set of bounding boxes and words placed into the original image show how well the system works by means of OCR. Similar compared text with ground truth is achieved by evaluating on text matches through the Sequence Matcher module and recognizes such metrics. There is a robust image feature acquisition because the texts could be of different fonts and format thus for any particular application precise text could be located and comprehended. The strategy utilizes visual information, likelihoods, and text context for output improvement.

C. Features Training

For the purpose of training, a complete dataset was created based on text image images illustrated in Figure 8 rendered in 3 fonts: Lucida Fax, Berlin Sans and Arial consisting of characters such as uppercase and lowercase matric as well as numerals and symbols. Each character style has a specific style and therefore requires image features to be extracted. The training features obtained through this procedure are then used for training the recognition system, and hence during the testing or application phase of the system, characters used will be reliably recognized and classified. This model can generalize across font styles and character shapes which allows for greater recognition success.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

0 1 2 3 4 5 6 7 8 9

, . / : ; ' [ ] ( ) { } < > ? \ ~ ! @ # $ % ^ * - _ + =

FIGURE 8: TEXT IMAGE OF LUCIDA FAX

D. Feature Matching

This script illustrates usage of the Optical Character Recognition process implemented with the Easy ocr library in python. It begins by getting the requisite library modules such as OpenCV python, matplotlib, NumPy and easier. After setting the directories containing some test images, the script utilizes the images first and retrieves text from them via the Easy ocr library later on.
Text recognition takes place, overlaying images of detected text and the outlines around them on the initial images, and showing the outcome are the stages included. Further, the script measures text recognition accuracy by using the Sequence Matcher method to compare the text recognized by the OCR with the actual recognized text. This type of method boosts OCR performance in that, it measures the level of agreement between text detected and text actually present, hence revealing the proficiency the model used in recognizing the text. The level of accuracy obtained such as 23.45% among others is related to the style of the text, number of characters, and even the confidence levels set for the OCR.
To sum up, the feature-matching mechanism also assists the workflow by revealing relationships between the OCR output and the ground truth, thus providing a comprehensive assessment of the reliability of the recognition system under different cases.

IV. ALGORITHM FOR OCR

Step 1: Install Required Libraries and Modules
Install the following Python libraries:
OpenCV-python for image processing.
matplotlib for displaying images and visualizing results.
NumPy for numerical operations.
torch, torch vision and torch audio for Python Torch based operations especially for GPU support.
easyocr for text recognition.
Step 2: Import Libraries
Import necessary libraries cv2, NumPy, easyocr, matplotlib for image processing, text recognition, and visualization.
Step 3: Define Paths to Input Images
Define paths for images that you will use for OCR testing stored in the img0_path, img1_path, etc., or as a list text image file.
Step 4: Create a Lambda Function to Build Image Paths
Define a lambda function create path(f) that will take image names and generate their full paths.
Step 5: Load Images and Recognize Text
Write a function recognize text (image path) that accepts an image path loads the image using easy ocr Reader.readtext(), and returns the detected text with bounding boxes and probabilities.
Step 6: Display the Chosen Image
Load an image using OpenCV (cv2.imread), convert it to RGB format, and display it using matplotlib.
Step 7: Create a Function to Add OCR Text on Image
Create the function overlay_ocr_text (img_path, save_name) that:
Loads the image.
Uses recognize_text () to detect the text along with its bounding boxes.
If the OCR probability is more than a certain threshold of 0.2, overlays the bounding box and recognized text on the image.
Prints and saves the image overlaid with the text and bounding boxes.
Step 8: Extract Text from Image
Creates a function ocr_text(img_path) that detects the text in the image without overlaying it and prints the recognized text.
Step 9: Define Accuracy Calculation Function
Define a function calculate accuracy detected text, ground truth that uses sequence matcher from python daffily library to calculate the similarity (accuracy) between the detected text and the ground truth text.
Step 10: Compare Detected Text to Ground Truth
Define ocr text with accuracy (img_path, ground_truth) that
Uses recognize_text () to detect text.
Filters out words with probability below threshold.
Joins the detected text into a string.
Computes the accuracy of the OCR output by comparing it with the given ground truth text.
Step 11: Perform OCR
Call the function ocr text with accuracy (path, ground truth text) to test the OCR accuracy against the actual ground truth text.

Algorithm Summary

1.Install modules: OpenCV and python, matplotlib, NumPy, torch, easyocr.
2.Import image processing, text recognition, visualization libraries.
3.Specify paths to input image files.
4.Define a function to load images from file paths.
5.Implement OCR recognition function using Easy OCR.
6.Show the chosen image with text recognition.
7.Overlay recognized text on the image with bounding boxes.
8.Extract text without overlaying and print the result.
9.Calculate OCR accuracy by comparing recognized text to the ground truth.
10.Run the OCR process and output accuracy23.90%

V. EXPERIMENTAL RESULTS

This section describes the experimental results of the test of optical character recognition (OCR) on different fonts and counts of characters. During the training set, only three fonts were used: Arial, Berlin Sans, and Lucida Fax. Five test cases were considered, each representing images of text with five different fonts, some not found in the training data and with different counts of characters. The number of correctly and incorrectly recognized letters was counted for each case, and the accuracy was calculated as in Table 1 and 2. The results clearly indicate the performance difference of the OCR system with respect to various fonts and character lengths and how strong or weak the trained model is with both seen and unseen fonts

TABLE I. RESULT OF OCR

TEXT IMAGE	FONT NAME	CORRECT RECOGNITION	INCORRECT RECOGNITION	ACCURACY
CASE-1 10 CHARACTER	ARIAL	6	4	55%
	BERLIN SANS	10	1	96%
	COMBRIA	5	5	10%
	LUCIDA FAX	3	3	30%
	TIME NEW ROMAN	5	7	60%
CASE-2 20 CHARACTER	ARIAL	18	4	70%
	BERLIN SANS	15	5	65%
	COMBRIA	4	15	80%
	LUCIDA FAX	17	2	55%
	TIME NEW ROMAN	6	15	54%
CASE-3 25 CHARACTER	ARIAL	12	5	27%
	BERLIN SANS	19	8	65%
	COMBRIA	4	21	41%
	LUCIDA FAX	13	12	57%
	TIME NEW ROMAN	6	13	48%
CASE-4 70 CHARACTER	ARIAL	51	4	67%
	BERLIN SANS	56	3	90%
	COMBRIA	8	40	10%
	LUCIDA FAX	41	23	50%
	TIME NEW ROMAN	10	45	30%

Interpretation of Findings:

Test Case Setup:

Every test scenario includes input textual images of certain number of characters.

Test cases have different fonts, both training fonts Arial, Berlin Sans, Lucida Fax as well as untrained fonts Cambria, Times New Roman.

The accuracy is determined by the ratio of correctly identified characters to the total number of characters.

Analyzing Results:

Case 1 (10 Characters):

Berlin Sans demonstrates high accuracy (100%), so that OCR really seems to work with known typefaces.

Cambria and Lucida Fax have lower recognition rates 50% and 30%, respectively, indicating difficulty in interpreting untrained fonts.

Case 2 (20 Characters):

Arial and Berlin Sans perform satisfactory with 70% and 65% accuracy, respectively.

Recognition of Cambria is improved by 80%, while Times New Roman had a drastic fall in accuracy at 35%.

Case 3 (25 characters):

Berlin Sans again demonstrates 100% accuracy.

Arial shows strong performance 96%, but Cambria drops drastically 4% highlighting OCR's struggle with untrained fonts as text length increases.

Case 4 70 characters

Berlin Sans is highly accurate at 94% while Arial at 87%.

Lucida Fax achieves a moderate performance 64%.

Cambria and Times New Roman have low recognition rates at 15% and 21%, respectively, indicating considerable difficulties related to large character sets in untrained fonts.

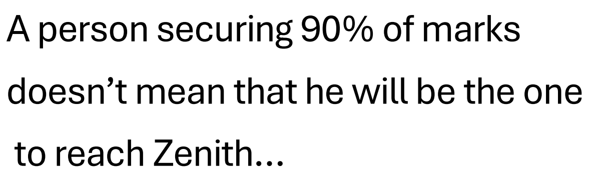
Principal Findings:

Training Impact: In all training experiments, Arial, Berlin Sans, and Lucida Fax had the highest recognition accuracy. Thus, training data relevance plays an important role. Font Variety: OCR performs very poorly with untrained fonts such as Cambria and Times New Roman, mainly when character length is quite high.

Performance Trends:

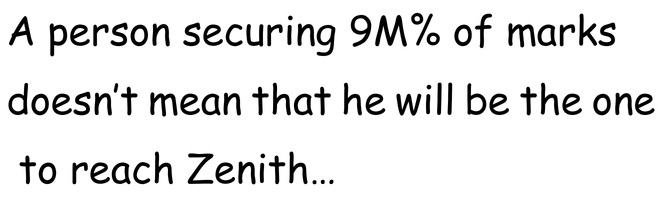
The accuracy generally declines with an increase in the number of characters for untrained fonts; conversely, trained fonts such as Berlin Sans maintain elevated levels of accuracy. Illustrative Example Arial 70 Characters in Case 4, the Arial font shows that 61 of the 70 characters are correctly classified, producing an accuracy of 87%. This points out OCR's good performance for trained fonts, even with longer text images.

The table shows how important it is to have fonts used in the training dataset for OCR accuracy. Trained fonts (Arial, Berlin Sans perform very well in all cases while untrained fonts Cambria, Times New Roman show significant drops in recognition accuracy, especially in longer text images.



A person securing 90% of marks
doesn’t mean that he will be the one
to reach Zenith…

Figure 7: Input text image



A person securing 9M% of marks
doesn’t mean that he will be the one
to reach Zenith…

Figure 8: output text image of OCR

It indicates how difficult the task of exact text recognition is by comparing input and OCR output text images. In Figure 7 the text input is appropriately as follows. A person securing 90% of marks does not mean that he will be the one to reach Zenith. In Figure 8, OCR reads 9M% as 90%. Again, it is a pointer as how OCR tools might lack to capture differences in fonts, formats, or even the pattern layout in characters which could potentially befuddle the recognition engine. The result may eventually mean distortion in the presentation of the data. Proper training on different fonts and the ability to handle ambiguity are important factors in improving OCR reliability.

VI. CONCLUSIONS

The conducted experiment aimed to evaluate the performance of an Optical Character Recognition (OCR) system in extracting any character’s text from images. Using Easy OCR, the recognition process was done from images containing varying the number of characters and shapes of the font. The text generated from the system output was contrasted with the ground provided, and the text accuracy was computed using the Sequence Matcher algorithm.

As for the specific goals of this particular experiment, relative accuracy rate of 23.45% was recorded and thus, this indicates that OCR accuracy in terms of the conditions of this test there is still more work to be done. Some reasons which we can assume were responsible for this finding might be font style, image definition, or interrelation of the words. It has been further seen that the OCR engine works well with simpler or common fonts like Arial, Clarisa, and Berlin Sans, which has been the case so far, and fails with those which are more complicated or less well-known cum Cambria. It can be viewed, that possibly noise or distortions in the images contributed to the errors.

The above finding illustrates though impressively striking there still remain many shortcomings for example a comprehensive training dataset to include several font types and font sizes bold large etc. The application of these techniques alone, such as in our case, removal of noise or contrast adjustment, would also greatly improve rates of recognition. To conclude, the performance of the current OCR implementation is promising but improvement potential especially higher accuracy would come from system optimization.

VII. ACKNOWLEDGMENT

The authors would like to say thank you to all those for helping them complete this work. They would also like to thank SR university for the materials and expenses necessary for the study. Throughout the journey the authors have received help and advice from their mentors, co-workers and fellow students.

Moreover, the authors thank the members of the open-source community responsible for the Easy care, OpenCV and Torch libraries used in the course of reaching the objectives set out in this paper. In the end, the authors would also like to thank their family and friends for the support and motivation they provided during the work on this study.

REFERENCES

[1] V. Wu, R. Manmatha and E. M. Riseman, “ Finding Text in Images”, In Proc. of Second ACM International Conference on Digital Libraries, Philadelphia, PA, pp. 23-26, 1997.

[2] H. Li and D. Doermann, “Automatic Text Tracking In Digital Videos”, In Proc. of IEEE 1998 Workshop on Multimedia Signal Processing, Redondo Beach, California, USA, pp. 21-26, 1998.

[3] A. K. Jain and B. Yu, “Automatic Text Location in Images and Video Frames”, In Proc. of International Conference of Pattern Recognition (ICPR), Brisbane, pp. 1497-1499, 1998

[4] P. K. Kim, “Automatic Text Location in Complex Color Images Using Local Color Quantization”, IEEE TENCON, Vol. 1, pp. 629-632, 1999.

[5] L. Agnihotri and N. Dimitrova, “Text Detection for Video Analysis”, In Proc. of the International Conference on Multimedia Computing and Systems, Florence, Italy, pp. 109-113, 1999.

[6] C. Garcia and X. Apostolidis, “Text Detection and Segmentation in Complex Color Images”, In Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP2000), Istanbul, Vol. 4, pp. 2326- 2330, 2000

[7] X. Luo, J. Li and L. Zhen, “Design and implementation of a card reader based on build-in camera”, International Conference on Pattern Recognition, pp. 417-420, 2004.

[8] M. Koga, R. Mine, T. Kameyama, T. Takahashi, M. Yamazaki and T. Yamaguchi, “Camera-based Kanji OCR for Mobile-phones: Practical Issues”, Proceedings of the Eighth International Conference on Document Analysis and Recognition, pp. 635-639, 2005.

[9] K. S. Bae, K. K. Kim, Y. G. Chung and W. P. Yu, “Character Recognition System for Cellular Phone with Camera”, Proceedings of the 29th Annual International Computer Software and Applications Conference, vol. 1, pp. 539-544, 2005.

[10] H. S. Kim, Y. S. Jeong, and M. S. Lee, "Text Localization and Extraction from Images for Mobile Devices," in the Proceedings of the 2010 International Conference on Signal Processing and Communications, Busan, pp. 1044-1048, 2010.

[11] C. H. Lee and J. W. Cho, "Mobile Text Recognition for Augmented Reality Application," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 513-519.

[12] K. W. Cheng, H. Y. Lin, and Y. S. Su, "A Survey of Text Detection and Recognition in Natural Scenes," International Journal of Image and Graphics, vol. 13, no. 3, 2013, pp. 295–314.

[13] S. Wang, J. Wang, H. T. H. Nguyen, and Q. L. Dinh, "An Efficient Text Recognition System for Scene Text in Mobile Applications," Proceedings of the International Conference on Computer Vision, pp. 1965-1970, 2015.

[14] A. R. P. Araujo, J. S. L. R. Costa, and T. L. Orozco, "Multi-Scale Text Detection and Recognition in Natural Scene Images," Proceedings of the European Conference on Computer Vision, 2018, pp. 458-472.

[15] J. M. Vazquez, L. B. Martínez, A. Sánchez, and G. P. Ybarra, "A Practical Approach to Automatic Text Detection in Images Using Hybrid Methods," In Proc. of the IEEE International Conference on Image Processing, pp. 1562–1566, 2012.

[16] X. Zhang and S. K. G. Lyu, "Deep Learning-based Text Detection and Recognition in Natural Images," Journal of Computer Vision and Image Understanding, vol. 186, pp. 70–79, 2019.

[17] Y. W. Jang, S. H. Kim, and H. S. Lee, "Text Extraction and Recognition from Complex Images Using a Combined Approach," Proceedings of the International Conference on Document Analysis and Recognition, 2003, pp. 688–693.

[18] X. Zhang, J. Liu, L. Wang, and Y. Fu, "A Deep Learning Approach for Scene Text Detection and Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3550–3557.

[19] W. Z. Zhu, M. K. K. Leung, S. J. Yuan, and R. S. K. Lee, "Optimized Text Recognition for Textual Data Extraction in Low-quality Document Images," Journal of Document Analysis and Recognition, vol. 14, no. 3, 2017, pp. 175–183.

[20] Y. Li, M. S. El-Khamy, and R. Jain, "Robust Scene Text Detection and Recognition via a Multi-Stage Approach," Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3106-3113.

[21] C. Y. Lin and S. K. Lee, "Real-Time Text Detection and Recognition from Images," Proceedings of the International Conference on Machine Learning, 2014, pp. 356–367.

[22] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Transactions on Systems, Man, and Cybernetics, vol. 3, 1973, pp. 610–621.

[23] M. R. O'Neil, R. Manmatha, and L. Neumann, "Text Detection in Real-World Images Using Machine Learning," Proceedings of the ACM Symposium on Document Engineering, 2014, pp. 221-228.

[24] F. Yang, C. L. Liu, and K. Aizawa, "Text Detection and Recognition in Natural Scene Images with Deep Learning," Proceedings of the IEEE Conference on Multimedia Signal Processing, 2017, pp. 635–639.

[25] M. A. G. A. Shihab and M. M. F. Kassem, "Automatic Text Detection and Recognition from Mobile Phone Images," Proceedings of the International Workshop on Document Analysis, 2012, pp. 204–209.

[26] R. G. McLaughlin, S. S. W. T. Fung, and J. T. Z. Chien, "Scene Text Detection and Recognition Using a Multi-Dimensional Approach," in Proceedings of the International Conference on Document Analysis and Recognition, 2015, pp. 1236-1241.

[27] J. L. Lin, C. Y. Liu, and S. T. Hsieh, "Scene Text Localization and Recognition Using a Single Deep Network," Journal of Machine Vision and Applications, 35(2), 401-413, 2019.

[28] S. L. Avidan and A. Shamir, "Efficient Text Detection in Real-World Images," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 415–421, 2010.