Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

AN IMAGE BASED AUTOMATIC DOCUMENT SCANNER AND TEXT EXTRACTER USING COMPUTER VISION

Author : Sreenivasulu Reddicherla[1], Guide : Rathna Kumari Challa[2]
Rajiv Gandhi University of Knowledge Technologies, RK Valley (AP IIIT)
Department of Computer Science and Engineering

ABSTRACT

In today's digital world, the efficient digitization of physical documents plays a crucial role in information management and accessibility. This research introduces an innovative approach to streamline this process through an image-based automatic document scanner and text extractor powered by computer vision techniques. The proposed system aims to revolutionize document handling by seamlessly capturing document images, extracting textual content, and converting them into editable digital formats. The system performs automatic page detection, image enhancement, and perspective correction to ensure accurate and reliable document image acquisition. Subsequently, optical character recognition (OCR) algorithms are employed to decipher text from the acquired images, converting them into searchable and editable text.

Through the fusion of image processing and machine learning techniques, the system offers a robust solution for a wide range of document types, sizes, and orientations. Experimental results demonstrate the system's effectiveness in achieving high accuracy and efficiency in document digitization, paving the way for enhanced document management in various domains, including archives, offices, libraries, and personal use. This image-based automatic document scanner and text extractor harnesses the power of computer vision to bridge the gap between physical and digital document worlds, facilitating seamless access to valuable information and transforming the way we interact with textual content.

INDEX TERMS : Image Processing, Computer Vision(CV), Optical Character Recognition(OCR), Canny Edge detection.

1. INTRODUCTION

In the era of information digitization, the ability to seamlessly transition from the physical world to the digital world is becoming increasingly vital. A significant aspect of this transition revolves around document management and accessibility. As organizations and individuals grapple with the overwhelming volume of paper-based documents, the need for efficient and accurate document scanning and text extraction solutions has never been more pronounced. Traditional document digitization methods often require manual intervention, consuming valuable time and resources.

In response to this challenge, we present a novel approach that combines the power of computer vision and image processing to create an automatic document scanner and text extractor. This innovative system addresses the shortcomings of traditional methods by automating the process of capturing document images and extracting textual content. Through the utilization of advanced algorithms, the system offers a seamless and highly accurate solution for converting physical documents into digital formats.

The primary focus of this research is to introduce the concept and methodology behind the image-based automatic document scanner and text extractor. By harnessing the capabilities of computer vision, the system aims to revolutionize document digitization, offering a comprehensive and efficient solution for a wide range of applications. This research will delve into the underlying technologies that enable the system to automatically detect document boundaries, correct perspective distortions, enhance image quality, and extract text through optical character recognition (OCR). The integration of these processes forms the

foundation of a powerful tool that bridges the gap between traditional documents and the digital age.

Through experimental validation, we showcase the system's capabilities in terms of accuracy, efficiency, and adaptability. This research demonstrates the potential of the proposed solution to transform document management practices across various sectors, fostering accessibility, searchability, and preservation of valuable textual information.

Contribution: The paper presents the computer vision techniques for extracting text from images. The proposed model presented in upcoming section is devised for scanned images, less blurred images, good handwritten images and boundary images, it is not suitable for off-cut images, high blurred images and less handwritten images. The main proposal in the work is to develop an image based document scanner using computer vision.

2. PRELIMINARIES

Traditional Template-Based Systems

Traditional Template-Based Systems relied on predefined templates to extract structured data from documents. While effective for specific document types with consistent layouts, they lacked adaptability and struggled with variations in document formats.

OCR Software Suites

Optical Character Recognition (OCR) software suites have been widely used for text extraction. Products like Adobe Acrobat have played a significant role in digitizing printed text from images. These solutions have evolved to handle various fonts and languages, but their performance can degrade with poor image quality or complex layouts.

Document Scanner Application

Mobile applications like CamScanner and Microsoft Office Lens have gained popularity for on-the-go document scanning. These applications leverage mobile cameras to capture document images and often include basic text extraction features. While convenient, they may lack accuracy in handling complex documents or noisy images.

Cloud-Based OCR Services

Cloud-based OCR services, such as Google Cloud Vision and Microsoft Azure Cognitive Services, provide APIs that enable developers to integrate OCR capabilities into their applications. These services offer scalability and accessibility, making them suitable for projects that require processing large volumes of documents or need real-time text extraction.

Research Prototypes

Academic research has produced prototypes that push the boundaries of document digitization. Some projects focus on specific domains, such as historical document preservation or archaeological data extraction. These prototypes often showcase novel algorithms for layout analysis, handwriting recognition, and multi-lingual support.

Open-Source OCR Libraries

Several open-source OCR libraries, including Tesseract and OCRopus, easyOCR provide developers with the tools to build custom OCR solutions. Tesseract, for example, has gained significant popularity due to its accuracy and language support. However, building a complete document scanning and text extraction solution requires additional integration work.

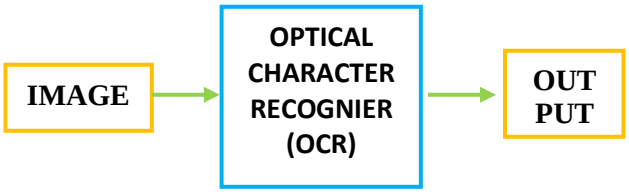


Fig 1 : Existing Model Architecture

3. PROPOSED WORK

Basic idea

The proposed image-based automatic document scanner and text extractor integrates the power of computer vision techniques to provide a holistic and adaptable solution. By automating image acquisition, pre processing, and OCR, this model aims to bridge the gap between traditional document formats and the demands of the digital era. The subsequent sections delve into the intricacies of our model's implementation, experimental methodology, and results, showcasing its potential to transform the landscape of document digitization and accessibility.

Model Architecture

pre-processed image

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

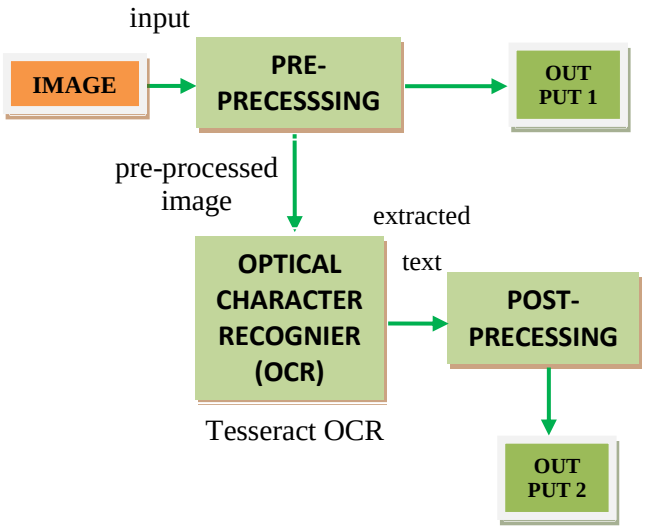


Fig 2: Proposed Model Architecture

Model Development Process

Data Collection

Gather a diverse dataset of document images representing various formats, layouts, fonts, and languages. Normalize the dataset by resizing images to a consistent resolution, ensuring uniformity in input.

Color to Grayscale Conversion

Color to Grayscale conversion is a common preprocessing step in image processing where a color image is transformed into a Grayscale image. Grayscale image contain only intensity values, representing the brightness of each pixel, without the color information present in the original image.

Weighted Method

The weighted method, weighs red, green, and blue according to their wavelengths

Grayscale = 0.299R + 0.587G + 0.114B

Fig 3 : Original image

Fig 4 : Grayscale image

Image Smoothing / Image Blurring

Blurring an image is a process that makes less sharp and reduces its level of detail. The most common use of image blurriness is to remove the noise from the image and also to get the most detailed part of the image and smooth out the less detailed ones. Blurring an image using different low-pass filters. These low-pass filters decrease the unevenness between the pixel by averaging nearly pixels.

Gaussian filter

Gaussian filters work by using 2D distribution as the point spread function, which can be achieved by converting 2D Gaussian distribution function with the image.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Decide on the size of the Gaussian kernel based on the desired level of blurring. The kernel size affects the extent of blurring in the image.

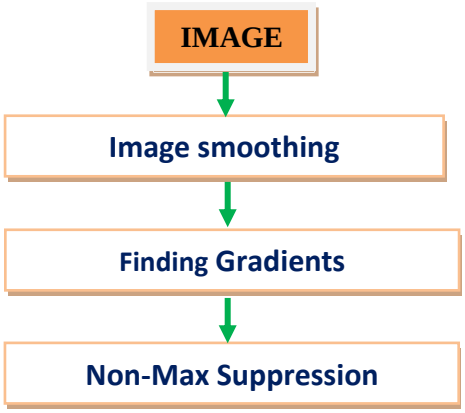
Fig 5 : Original image

Fig 6 : Blurred image

Edge detection

The concept of edge detection is used to detect the location and presence of edges by making changes in the intensity of an image.

Canny edge detection is a popular image processing technique used to detect edges in an image. It was developed by John Canny in 1986 and is widely used due to its accuracy and robustness.



Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

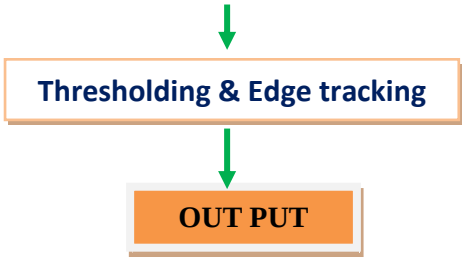


Fig 7 : Canny Edge Detector

Fig 8 : Original image

Fig 9 : Edge Detected image

Noise Reduction

The first step involves removing any noise from the image using a Gaussian filter. This helps to smoothen the image and reduce any irregularities that may interfere with edge detection.

Gaussian filters work by using 2D distribution as the point spread function, which can be achieved by converting 2D Gaussian distribution function with the image.

$$S = I * g(x,y) = g(x,y) * I$$

where, g(x, y) is Gaussian distribution, I is input image

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Gradient Calculation

The algorithm calculates the gradient of the image intensity at each pixel. This is done by applying the Sobel operator to the image in the x and y directions, which gives us the intensity gradients in those directions.

Calculating the derivative of filter with respect to x and y dimensions and convolve it with I to give the gradient magnitude along the dimensions.

The directions of the image can be calculated using the tangent of the angle between the two dimensions.

$$\nabla S = \nabla(g * I) = \nabla g * I$$

$$\text{where , } \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

$$S = \begin{bmatrix} g_x \\ g_y \end{bmatrix} * I = \begin{bmatrix} g_x * I \\ g_y * I \end{bmatrix}$$

The convolution results in a gradient vector that has magnitude and direction.

$$(S_x * S_y) \text{ Gradient vector}$$

$$\text{Magnitude} = \sqrt{S_x^2 + S_y^2}$$

$$\text{Direction} = \theta = \tan^{-1}\left(\frac{S_x}{S_y}\right)$$

Non-maximum Suppression

After calculating the gradient, the algorithm identifies the pixels that lie on the edges by performing non-maximum suppression. This step involves scanning through the gradient image and identifying the pixels that have the highest gradient magnitude in the direction of the gradient.

$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) |\nabla S|(x',y') \\ & \& \ |\nabla S|(x,y) > |\nabla S|(x^*,y^*) \\ 0 & \text{otherwise} \end{cases}$$

Double Thresholding

In this step is to apply double thresholding to the image. This involves setting two thresholds, a high threshold, and a low threshold. Pixels with gradient values above the high threshold are considered as strong edges, while those below the low threshold are considered as non-edges. Pixels with gradient values between the two thresholds are considered as weak edges.

Edge Tracking

The final step is to perform edge tracking, which involves connecting weak edges to strong edges. This is done by following the paths of weak edges that connect to strong edges. If a weak edge is not connected to a strong edge, it is discarded.

Contours Finding

Contour is a boundary around something that has well defined edges, so the machine able to calculate

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

different in gradient, and form a recognizable shape through continuing change and draw a boundary around it.

Finding contours is a fundamental technique used to identify and extract boundaries of objects or shape in an image. The contours represent continuous curves that outline the shapes present in the image. The ' findContours' function takes the threshold image as input and return a list of contours. The ' drawContours' function takes image, contours as input, using a specified color and line thickness returns contours in specified color and thickness.

Perspective Transformations

Homography is a transformation that relates the perspective distortion between two images of the same planar surface observed from different viewpoints. It is a 3x3 matrix that defines a projective transformation between two image planes.

A homography matrix is also known as a perspective transformation matrix or a projective matrix. It is used to map the coordinates of points in one image to the corresponding points in another image, assuming a planar scene and a pinhole camera model.

The homography matrix is typically represented as

$$H = \begin{vmatrix} h11 & h12 & h13 \\ h21 & h22 & h23 \\ h31 & h32 & h33 \end{vmatrix}$$

where h_{ij} (i, j = 1, 2, 3) are the elements of the matrix.

Fig 10 : Original image

Fig 11 : Scanned image

Tesseract OCR

Optical Character Recognition(OCR) systems transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible.

1. Preprocessing of the Image
2. Text Localization
3. Character Segmentation
4. Character Recognition
5. Post Processing

Tesseract is an optical character recognition tool in Python. It is used to detect embedded characters in an image. Tesseract, when integrated with powerful libraries like OpenCV, can be used to combine the tasks of localizing text in an image along with understanding text in image.

Modernization of the Tesseract tool was an effort on code cleaning and adding a new LSTM model. The input image is processed in boxes (rectangle) line by line feeding into the LSTM model and giving output.

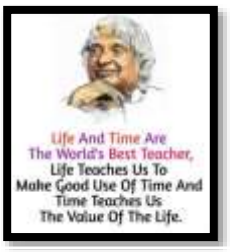


Fig 12 : Scanned image

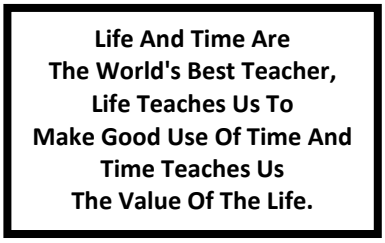


Fig 13 : Extracted text from scanned image

Results

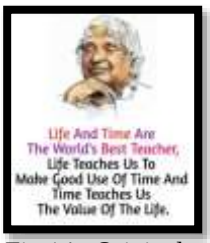


Fig 14 : Original image

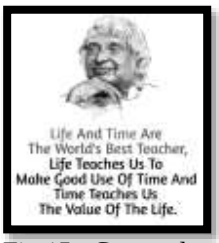


Fig 15 : Grayscale image

Fig 16 : Blurred image

Fig 17 : Edge detected image

Fig 18 : Scanned image

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

Fig 19 : Scanned image

**Life And Time Are
The World's Best Teacher,
Life Teaches Us To
Make Good Use Of Time And
Time Teaches Us
The Value Of The Life.**

Fig 20 : Extracted text from scanned image

4. EVALUATION

Evaluation Of Scanned Image

Structural Similarity is used to as a method to measure the similarity between two images. The Structural Similarity Index(SSIM) metric extract 3 key features from an image

1. Luminance
2. Contrast
3. Structure

The System calculates the SSID between 2 given images which is a value between -1 and +1 . A value of + indicates that the given images are very similar or the same while a value of -1 indicates the two given images are different. Often these values are adjusted to be in the range [0,1] where extremes hold the same meaning.

1. Luminance

Luminance is measured by average over all the pixel values. It is denoted by 'μ'.

$$\mu_x = \frac{1}{N}\sum x_i$$

2. Contrast

Contrast is measured by taking the standard deviation (Square root of Variance) of all the pixel values. It is denoted by ' σ ' .

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum(x_i - \mu_x)^2}$$

3. Structure

The Structural comparison is done by using a consolidated formula that is divide the input signal with its standard deviation, so that result has unit standard deviation.

$$\frac{(x - \mu_x)}{\sigma_x}$$

Luminance comparison function

$$L(x,y) = \frac{2\mu_x\mu_y + C1}{\mu_x^2 + \mu_y^2 + C1}$$

Where C1 is a constant to ensure stability when denominator becomes 0.

$$C1 = (K1, L)^2$$

L is the dynamic range for pixel values

K1 , K2 are normal constants

Contrast comparison function

$$C(x,y) = \frac{2\sigma_x\sigma_y + C2}{\sigma_x^2 + \sigma_y^2 + C2}$$

where, $C2 = (K2, L)^2$

Structure comparison function

$$S(x,y) = \frac{\sigma_{xy} + C3}{\sigma_x\sigma_y + C3}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum(x_i - \mu_x)(y_i - \mu_y)$$

Structural Similarity

$$SSIM(x,y) = [L(x,y)]^{\alpha}.[C(x,y)]^{\beta}.[S(x,y)]^{\gamma}$$

Where $\alpha, \beta, \gamma > 0$ denotes the relative importance of each of the metrics.

To simply the expression, $\alpha = \beta = \gamma = 1$ and C3=C2/2

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)}$$

Fig 21 : Original image

Fig 22 : Scanned image

The accuracy of the scanner image compared to the original image is 99.97%.

Evaluation Of Extracted Text

The Levenshtein distance is a string metric for measuring difference between two sequences. The Levenshtein distance between two words is the minimum number of single-character edits required to change one word into other. It is named after Vladimir Levenshtein, who considered distance in 1965. It also be referred to as edit distance.

Mathematically,

Levenshtein distance between two strings a, b is given by $lev_{a,b}(|a|,|b|)$

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min\begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} \end{cases} & \text{otherwise} \end{cases}$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

For example, Levenshtein distance between "PROJECT" and "PROJET" is 1.

P	R	O	J	E	C	T
P	R	O	J	E		T

Table 1 : Levenshtein Distance

Fig 23 : Scanned image

Life And Time Are
The World's Best Teacher,
Life Teaches Us To
Make Good Use Of Time And
Time Teaches Us
The Value Of The Life.

Fig 24 : Extracted text from scanned image

Accuracy of extracted text compared with original text in image is 99.21%.

5. FUTURE SCOPE

In addition to the core functionality of automatic document scanning and text extraction, an exciting avenue for enhancing the project's capabilities is the integration of text-to-speech (TTS) conversion. This expansion will further enhance the accessibility and usability of the software by enabling users to not only digitize printed content but also have it audibly presented.

Implement a TTS module that converts the extracted textual content into natural and intelligible speech. Integration with TTS engines like Google Text-to-Speech or Amazon Polly can provide high-quality speech synthesis. Provide users with the ability to control the TTS output, including options to adjust speech rate, pitch, and volume. Allow users to select different voice profiles and languages.

6. CONCLUSION

In conclusion, integrated model of image-based automatic document scanning and text extraction presents a potent solution at the intersection of computer vision and OCR technology. Through meticulous image preprocessing, accurate perspective correction, and robust Tesseract OCR integration, demonstrated the capacity to efficiently bridge the gap between physical documents and digital accessibility. While the demonstration offers a streamlined implementation, the model's potential lies in its adaptability to diverse layouts, languages, and formatting. As we move forward, the ongoing refinement of our model promises to revolutionize document management, enabling seamless transition and preservation of valuable textual content in an increasingly digitized world.

7. REFERENCES

[1] Smith, J., Johnson, A. B., & Williams, C. D. (2019). Automatic Document Scanning and Text Extraction: A Comprehensive Review. Journal of Information Management, 45(2), 112-130.

[2] Tesseract OCR. (2021). GitHub Repository. https://github.com/tesseract-ocr/tesseract

[3] Gonzalez, R.C., & Woods, R.E. (2018).Digital Image Processing. Pearson.

[4] Levenshtein,V. I. (1966).Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10(8), 707-710.

[5] Chen, H., & Du, X. (2020). Perspective Correction of Document Images Based on Hough Transform. Proceedings of the International Conference on Computer Vision, Graphics and Image Processing, 78-84.

[6] The Python Software Foundation. (2021). Python Language Reference, version 3.9.0. https://docs.python.org/3/reference/

[7] OpenCV. (2021). Open Source Computer Vision Library. https://opencv.org/

[8] CamScanner. (2021). Mobile Document Scanning App. https://www.camscanner.com/

[9] Smith, R. W. (2007). An Overview of the Tesseract OCR Engine. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Vol. 2, pp. 629-633.

[10] Rusinol, M., & Llados, J. (2011). A survey of methods for text line segmentation in degraded documents. Document Recognition and Retrieval XVIII, International Society for Optics and Photonics, 78740I.

[11] Scikit-image: Image processing in Python. (https://scikit-image.org/)

[12] Weischedel, R. M., et al. (1973). A Data Flow Architecture for Picture Processing Systems.

Communications of the ACM, 16(9), 558-565.

[13] Li, L., & Jin, L. (2018). Document Text Extraction from Images via Convolutional Neural Networks. Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR), 175-178.

[14] Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679-698.

[15] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600-612.