IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SURVEY PAPER ON TEXT FETCHER APPLICATION FROM IMAGE PROCESSING

Prof.Nidhi Sharma
Department of Computer

Himanshu Komalsingh Pardeshi
Department of Computer

Sonali Sathe
Department of Computer

Bharati Vidyapeeth college of engineering, Navi Mumbai, Maharashtra, India

Abstract:

Picture processing is one of the major domains that has seen a boom in recent years. In general, we need sentences and paragraphs in text format for documents. Often the data is in the form of an image, and we need to extract the sentences from the image data. This requires us to retype the entire data set based on what we see in the picture, which makes the job even more boring.

To solve this issue, we'll create a graphical user interface framework that will convert the written text in the image to a text format that can be read. For this, we will use the image processing principle and programming. To retrieve the photo, the application will be granted access to the system's data. The user only needs to open the application and select a picture from a folder containing text. The photo will be scanned and the text in the photo will be translated to actual text format before being opened in a local note editor when it is imported to the application.

Programming with appropriate packages and modules, GUI design, and an application building framework will all be part of this application. As a result, the application's main task would be to extract text from images.

LIST OF ABBREVIATIONS

	Abbreviation description
GUI :	Graphical User Interface
OS:	Operating System
OCR:	Optical Character Recognition
IDE:	Integrated Development Environment
VCS:	Version Control System
NUI:	Natural User Interface
API:	Application Program Interface
TUI:	Tangible User Interface
PIL:	Python Imaging Library
DFD:	Data Flow Diagram

Introduction:

Purpose

There is a high demand for applications that can recognise and recognise characters these days. The paper document and data must be saved in digital format. However, as paper papers are scanned, they become pictures that cannot be edited. So programme that can recognise characters and print them in textual form that can be edited if necessary. . To do so, the paper text must be transformed into images, which must then be processed, a process known as image processing. The character is identified after the picture has been processed. Optical Character Recognition (OCR), which relies on machine learning and image processing, can help with this.

Product Scope

What does OCR have in store for the future? OCR will become a valuable tool for potential data entry applications if enough entrepreneurial designers and research and development dollars are invested.

data entry systems in the future However, in a capital-short setting, the limited availability of funds can limit the technology's developmentThe OCR scheme, however, can provide a lot of benefits if given the proper impetus and encouragement. They are as follows: -

OCR, or optical character recognition, is one of the most appealing labor-saving technologies.

The machine recognises new font characters very quickly and easily.

We can more easily edit the information in the papers, and we can reuse the edited information as needed. Future research will focus on programme extensions other than editing and searching. We've presented a strategy for making effective use of OCR. This strategy focuses on creating apps that can be used in everyday life. This approach can be used to accurately translate a wide variety of languages.

REVIEW OF LITERATURE

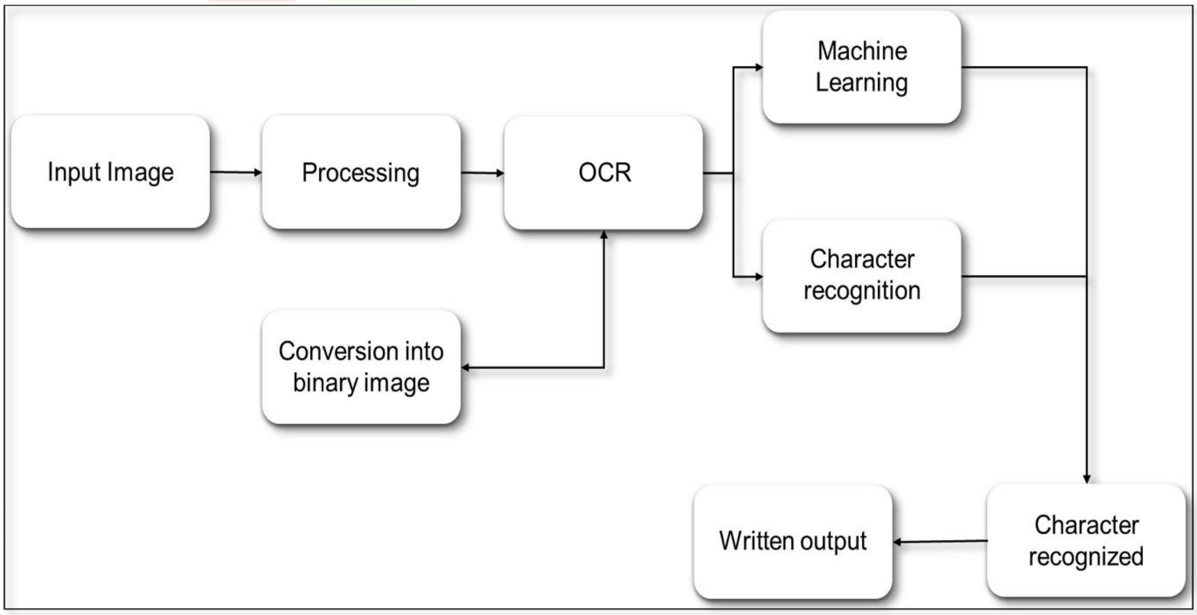
2.1 Related work

Various methods for detecting and localising text in photographs and videos have been suggested in the past. Color, strength, connected-components, edges, and other properties related to text in an image are all taken into account in these methods. These properties are used to differentiate text regions from the image's context and/or other areas The input image is first pre-processed to eliminate any noise if it exists in the colour clustering algorithm. The image is then separated into colour layers and a grey section. This method takes advantage of the fact that colour Characters in text are usually distinct from colour data in the background. These layers' linked component based heuristics are used to locate possible text regions. Also used is the aligning and merging analysis (AMA) process, which examines each row and column value separately. The algorithm is robust in locating mostly Chinese and English characters in photographs, according to the experiments; however, some false alarms occurred due to uneven lighting or reflection conditions in the test images. Color continuity is also used in the text detection algorithm. For text area extraction, it also employs multi-resolution wavelet transforms and blends low and high level image functions. Text is separated from its context using texture-based segmentation. A bottom-up "chip generation" method is also used, which takes advantage of the spatial cohesion property of text characters. For text area extraction, low and high level image features are used. Text is separated from its context using texture-based segmentation. In addition, the spatial cohesion property of text characters is used in a bottom-up "chip generation" process. Chips are groups of pixels in an image that represent possible text strokes and edges. Except for very small text characters that are not properly identified, the results show that the algorithm is robust in most situations. Misclassifications in texture segmentation often occur when the image has poor contrast. A scheme for text area localization based on concentration of attention has previously been suggested. Text regions in images are detected using intensity profiles and spatial variance. With the original image at various resolutions or scales, a Gaussian pyramid is formed. The text regions are observed in the pyramid's highest resolution image, then in each subsequent lower resolution image.

Author	Year	Approach	Features
Ohya et al.	1994	Adaptive thresholding and relaxation operations	Color, scene text (train, signboard, skew and curved), localization and recognition
Lee and Kankanhalli	1995	Coarse search using edge information, followed by connected component (CC) generation	Scene text (cargo container), localization and recognition
Smith and Kanade Zhong et al.	1995	3×3 filter seeking vertical edges	Caption text, localization
Yeo and Liu	1996	Localization based on large inter-frame difference in MPEG compressed image	Caption text, localization
Shim et al.	1998	Gray level difference between pairs of pixels	Caption text, localization
Sato et al.	1998	Smith and Kanade's localization method and recognition-based character extraction	Recognition
Antani et al.	1999	Multiple algorithms in functional parallelism	Scene text, recognition
Zhong et al	2000	Texture analysis in DCT compressed domain	Caption text, JPEG and I-frames of MPEG, localization
Chen et al.	2001	Text detection in edge-enhanced image	Caption text, localization and recognition

EXPERIMENTAL METHODOLOGY/DESIGN

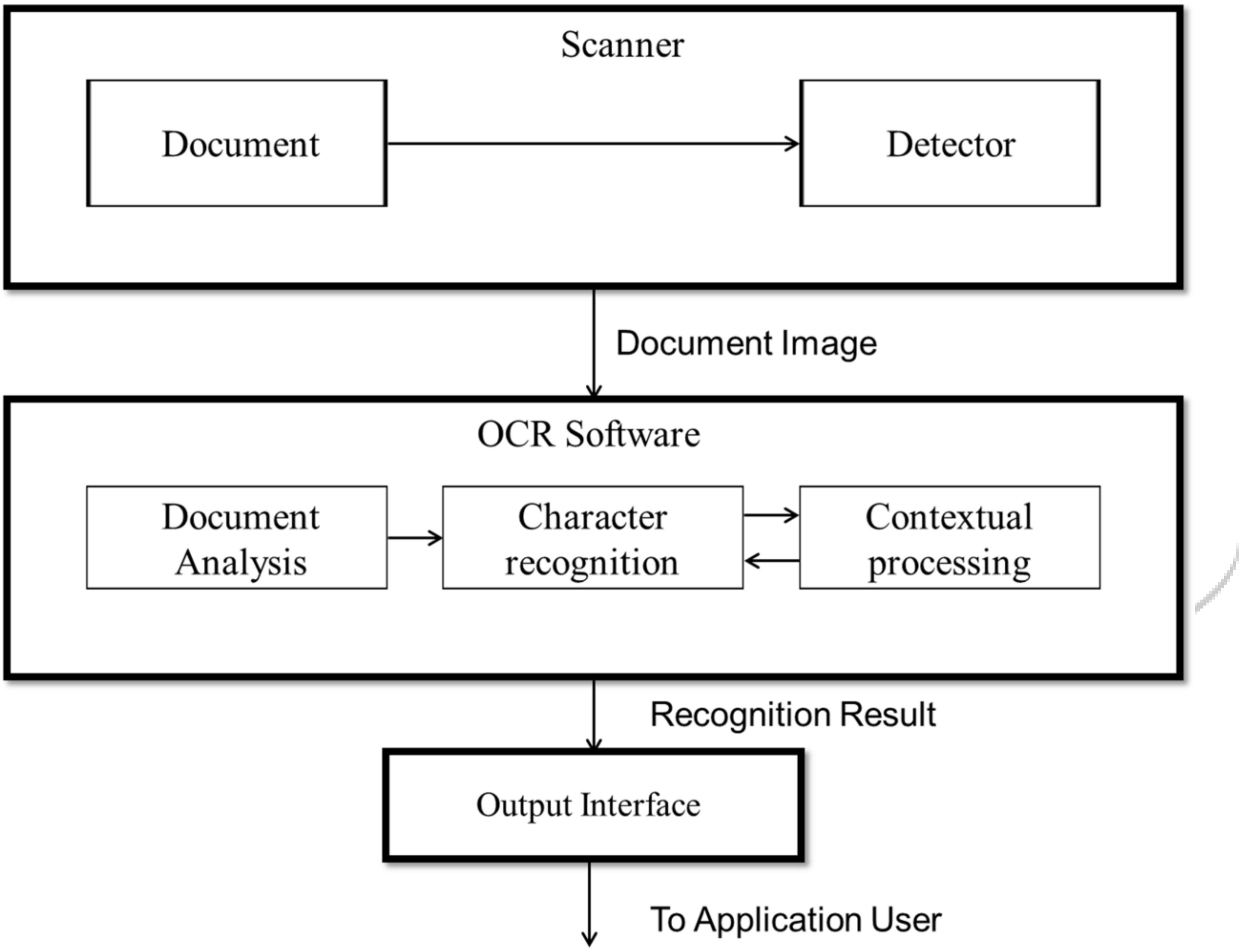
3.1 Block Diagram



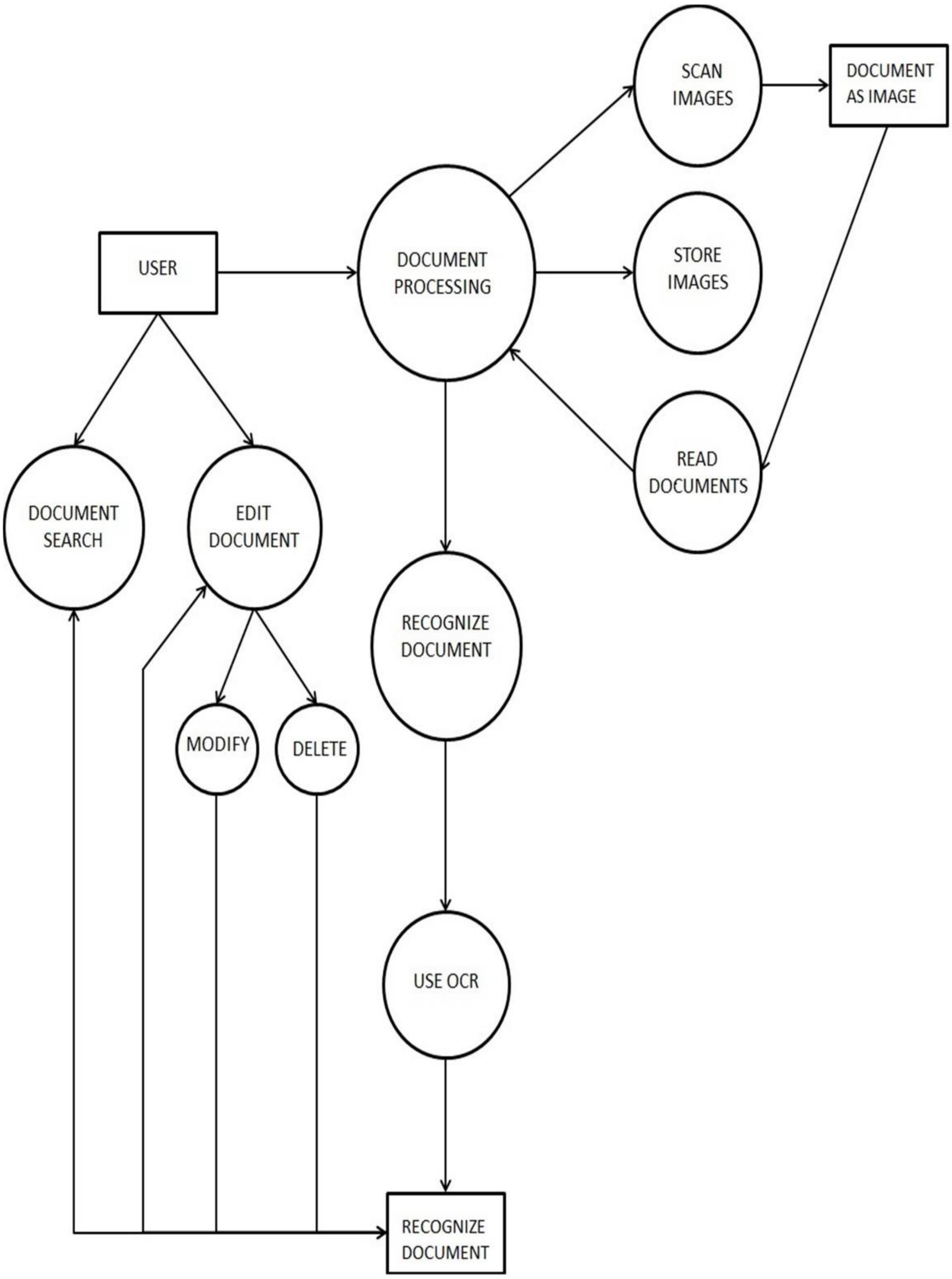
There is a written text in an image or document that is chosen. Via the gallery, the picture is submitted to the application. The image is scanned and analysed for further processing after it is uploaded into the application.

The colour difference between the background and the foreground is tested when the picture is processed. And there's Optical Character Recognition to considerThe foreground and background are separated in OCR by converting the input image to a binary image. And a pattern is recognised in This binary image is built on the pattern and orientation of the image's zeros and ones. After all of the characters have been extracted from the image, they are compared to the alphabet input characters that are already present in the OCR programme for character comparison and recognitionWhen a character's pattern is paired with a character in an OCR application, the pattern is recognised as a character. Character identification is completed and then translated into the required text format. After all of the characters have been translated to text format, the current text is saved in.txt format. as a whole so that now it can be used by the user as per their convenience.

Proposed Algorithm



Data flow:


SCAN IMAGES
DOCUMENT AS IMAGE
USER
DOCUMENT PROCESSING
STORE IMAGES
READ DOCUMENTS
DOCUMENT SEARCH
EDIT DOCUMENT
RECOGNIZE DOCUMENT
MODIFY
DELETE
USE OCR
RECOGNIZE DOCUMENT

CONCLUSION

We've presented a strategy for making effective use of OCR. This strategy focuses on creating apps that can be used in everyday life. This approach can be used to accurately translate a wide variety of languages.

REFERENCES

[1]. J. Ohya, A. Shio, and S. Akamatsu, Recognizing Characters in Scene Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (2) (1994) 214-224.

[2]. C.M. Lee, and A. Kankanhalli, Automatic Extraction of Characters in Complex Images, International Journal of Pattern Recognition Artificial Intelligence, 9 (1) (1995) 67-82.

[3]. M.A. Smith and T. Kanade, Video Skimming for Quick Browsing Based on Audio and Image Characterization, Technical Report CMU-CS-95-186, Carnegie Mellon University, July 1995.

[4]. Yu Zhong, Kalle Karu, and Anil K. Jain, Locating Text In Complex Color Images, Pattern Recognition, 28 (10) (1995) 1523-1535.

[5]. B.L. Yeo, and B. Liu, Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video, Proc. of SPIE, 1996, pp.142-149.

[6]. T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, Video OCR for Digital News Archive, Proc. of IEEE Workshop on Content based Access of Image and Video Databases, 1998, pp. 52-60.

[7]. S. Antani, U. Gargi, D. Crandall, T. Gandhi, and R. Kasturi, Extraction of Text in Video, Technical Report of Department of Computer Science and Engineering, Penn. State University, CSE-99-016, August 30, 1999.

[8]. Yu Zhong, Hongjiang Zhang, and Anil K. Jain, Automatic Caption Localization in Compressed Video, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, (4) (2000) 385-392.

[9]. D. Chen, K. Shearer, and H. Bourlard, Text Enhancement with Asymmetric Filter for Video OCR, Proc. of International Conference on Image Analysis and Processing, 2001, pp. 192-197.

[10] https://pypi.org/project/Python-tesseract/

[11] https://github.com/madmaze/Python-tesseract