© September 2025 | IJIRT | Volume 12 Issue 4 | ISSN: 2349-6002

Automated Detection and Classification of Text and Non-Text Elements in Images using Python and OpenCV

PRIYANKA G.V [1], Mr. SANTHOSH S. G[2]

[1] *PG Student, Dept of MCA, Jawaharlal Nehru new College of Engineering, Shimoga, Karnataka, India.*

[2] *Associate Professor, Dept of MCA, JNNCE, Shimoga, and Research Scholar, Dept of ICIS, Srinivas University, Mangalore, India. ORCIDID:0009-0004-1587-4656*

***Abstract*—This study introduced a lightweight and modular system for the automated detection and classification of textual and non-textual elements contained within digital images which synthesized deep learning-based OCR and classical computer vision processes. The outlined process combined EasyOCR for the detection and recognition of printed and handwritten text with OpenCV- based contour analysis to detect non-text elements including logos, diagrams, or other shapes. Implemented as a web application using Python and Flask, the system accepts user uploaded images, detects all textual and non-textual elements, outputs annotated digital images that denotes textual regions in green and non-text regions in blue. The device further measures the ratio of the text area relative to the image area, a useful metric for understanding how an image's content is displayed in regard to layout. The ratio of the text area is especially important for applications in document analysis, media summarization, and digital content moderation processes. The open-source tooling of the system promotes accessibility as well as modularity within the context of larger interpretations of the system. Overall, the project contributes a useful, interpretable hybrid classification of the text and non-text nature of content in an image which also raises opportunities for improving many of the items outlined in the future work section such as: real-time processing; multilingual amounts of content output; and structured data output regarding the detected textual and non-textual elements that can be integrated into pipeline systems.**

***Index Terms*—EasyOCR, Image Classification, Non-Text Segmentation, OpenCV, Python, Text Detection, Web Application.**

I. INTRODUCTION

In the contemporary digital era, there is a growing demand for smart systems that can understand and classify the content of images. Whether for document digitization, moderation of the content, advertising measures technologies for education, the fundamental challenges are distinguishing between text and non-text in a variety of image types given as the input. Most OCR tools and technologies can extract printed or handwritten text, which is part of the problem. Unfortunately, these technologies typically ignore non-text elements such as logos, charts, diagrams and icons - these 'non-text' elements are even more important when you are trying to understand the context within a visual document. Even commercial APIs for OCR, while powerful, have their own concerns. API delivery often makes significant assumptions about usage (e.g. internet, cost and customizability), and are typically not implementable for low/real-time or off-line deployment.

The study presents a web-based application in Python that integrates EasyOCR for high-quality text recognition and OpenCV for identifying non-text areas through contour analysis. As a whole, the application enables a straightforward user experience with image uploading through the viewer to receive annotated outputs in real-time; text areas highlighted in bright green and non-text areas highlighted in bright blue over an overlay displaying the percentage of the image area that contained text. This quantitative measure can be beneficial for layout analysis, content filtering, or media summary etcetera. The application is produced using only open-source tools, it is built with Flask, so that aims to be — accessible, modular, which, if required can rapidly scale the application. The new system takes advantage of deep learning and classical image processing, to realize a practical, generally effective and rapid option for hybrid image content classification and can also be taken further to real-time video processing or develop a multilanguage OCR or even build upon a generic object detection application framework.

II. LITERATURE SURVEY

Awasthi, et al., (2019) [1] The study targeted advertisement text overlay using both detected logoi. The authors focused on separating the visible content that is not text, from the textual content, using shape detection in contour form, and text recognition methods.

Chaitra Y. L. et al., (2023) [2] The authors used Faster-RCNN for text detection and EasyOCR for recognition and report an improvement F-score of 2.1 % for the ICDAR13 datasets and an improvement of 1.4 % for the ICDAR15 datasets; there was an improvement of 1.8 % in recognition accuracy.

Chen et al., (2021) [3] This study examined hybrid document analysis systems that integrate semantic text region detection (with EAST and CRAFT) with visual element segmentation. Based on their findings, the authors stated that the deep-learning-based text detectors are able to segment the text regions in a successful way, while contour-based methods are apt at handling the non-text content of the same document.

Hassan et al., (2021) [4] Assessed the outline contour-based non-text segmentation and OCR-based text detection described above in graphical illustrations such as research or scientific diagrams. The authors found that contour and thresholding methods worked well at identifying logos, icons, and charts, while adequately preserving text regions that were detected using an OCR process.

Kumar et al., (2020) [5] developed a hybrid system that integrated EAST to detect text and OpenCV thresholding to identify non-text objects. The authors successfully separated diagrams and charts from text-like objects in their scanned reports. They demonstrated the complementary nature of deep learning and classical image processing.

Li et al., (2019) [6] devised an end-to-end strategy to simultaneously detect text and visual elements in documents through learning combined with handcrafted features. They gave significant weight to layout-awareness in risks like automatic resume parsing or elements to build a legal document, finding that hybrid systems could improve detection of text and non-text.

Patel et al., (2019) [7] Explored hybrid document layout analysis to detect charts, logos, and images while separating text. They implemented edge-detection pipelines, creating non-text segmentations using contour descriptors of the non-text areas. Their findings, even with overlays inducing noise on images or scanned document would be robust.

Reddy et al., (2019) [8] This research focused on detection of visual objects of interest (e.g., charts, visual images) and their isolation in a variety of legal, financial, and other documents. They used a shape-based descriptor as well as a contour-based descriptor to isolate the non-text objects and proved the salient and structural information was more useful for the isolation than anything else.

Santhosh S. G. et al., (2024) [9], proposed a real-time text extraction and classification system for bilingual road signboards using OCR engines. The study highlights the current difficulties of recognizing multilingual characters and subsequently resolves this by utilizing the parallel processing of scripts to increase speed and accuracy when directly applied to live situations. The model showcased strong results in dynamic road conditions and varying degrees of illumination.

Santhosh S. G. et al. (2025) [10] reported improving machine learning methods to classify text in a bilingual document in real-time. The researchers focused on improving the preprocessing and feature extraction procedures to improve accuracy in classifying bilingual document types. Their model was able to examine mixed language data sets. While there was variability in how the text was structured, the classifier was able to classify these document types with confidence in a reliable manner.

Singh and Awasthi, (2018) [11] This study used shape-matching algorithms to recognise brand logos in advertisements. In this way, contour descriptors and edge-detection pipelines distinguished brands logos, which were non-textual content from the textual overlays demonstrating the functional characteristics of OpenCV based methods used to identify an image's visual elements in complex photographs.

III. MATH

The presented system calculates the percentage of text content and non-text content of a document image based on the bounding box information obtained for the text using OCR and on contours identified using contour detection. The equations which are used are shown below:

Text Area % = $\frac{\sum_{i=1}^{n}(Wi \times \mathrm{hi})}{W \times H} \times 100$

Non -Text Area % = 100 – Text Area %

where:

- W_i, h_i= width and height of the i^{th} detected text bounding box
- n = total number of text regions detected
- W×H = total image area in pixels

This model will correspond directly to the implementation, allowing measurement of how much of a document is taken up by the text and how much is non-text.

IV. METHODOLOGY

The system presented provides an automatic mechanism to separate text and non-text regions from digital document images. The process is divided into five steps that start with user input and document image upload, and end with back-end processing for the text and non-text detection and delivery of visual annotated results and statistical results.

4.1 User Interface

We developed a lightweight web-based interface with HTML, CSS, and JavaScript. The interface permits users to upload document images through the browser directly, without extended setup or configurations. When users select an image file from their device, it is sent to the server as a POST request that initiates the processing pipeline. This interface is designed to be cross-platform and user-friendly.

4.2 File Handling and Image Acquisition

Each uploaded image to the server is given a unique identifier and is stored temporarily by the server for processing. OpenCV is used to read the document images and convert them into numerical arrays, allowing processes to be carried out on the pixel levels that are needed for the detection of text and non-text regions. By using this two-phase approach, our application ensures proper processing for multiple uploaded document images, as well as propriety of the images during processing.

4.3 Text Detection

Text detection is accomplished through the EasyOCR engine, which is a deep learning-based Optical Character Recognition system. The engine detects bounding box coordinates, recognized text, and confidence levels for every object detected. On the original image, the areas of detected text are shown with green bounding boxes, to aid the user in recognizing textual components.

4.4 Generating a Text Mask

A binary mask is created to distinguish text and non-text objects that is the same size as the original image. The detected text bounding boxes are filled into the mask for the purpose of tagging all regions of text in the mask. This mask will be used in the next step as a spatial mask so that the graphical components can be detected without the influence of text regions.

4.5 Non-Text Detection

The non-text objects (logos, diagrams, or decorative shapes) are extracted using computer vision techniques. The first step is converting the input image to grayscale, and use the Gaussian filter to lessen noise. The next step is thresholding the 'bright' regions. By subtracting the 'text' mask from the thresholded image, we have only the regions with non-text content to analyze. Using the contours of the remaining region, if the area of the contour is more significant than a threshold, we will classify the shape as non-text. This will be drawn on the output image with blue bounding boxes.

4.6 Content Proportion Calculations

The proportion of image area taken up by text is calculated by totaling the area of all the bounding boxes found and comparing this to the total area of the image blank space. By doing this, any remaining area is classified automatically as non-text. This percentage-based analysis provides a quantitative sense of how much of the document is textual compared to graphical content.

4.7 Production and Presentation of Results

The final product of the automated text detection and annotation pipeline is an annotated output image that identifies text and non-text areas. In addition to the annotated image, statistical results are included that detail the number of detected text regions, the number of detected non-text regions, and the percentage of both text and non-text areas. A graphical output, such as a pie chart, is also included to provide the user with a visual representation of the document's composition.

4.8 Tools and Technologies Used

Tool/Library	Purpose
Python 3.8+	Core programming language for implementing logic and modules
Flask	Web framework for routing,request handling, and template rendering
OpenCV	Image processing and non-text detection using thresholding & contours
EasyOCR	Deep learning-based OCR engine for text detection and recognition
Numpy	Numerical operations and image array manipulation
HTML/CSS/JS	Frontend UI for file upload, preview,and result visualization

4.9 Work Flow of the Proposed System

The proposed workflow starts with the user uploading an image and pre-processing of that uploaded image. EasyOCR is used for text detection and mask generation. The system uses contour methods of analysis for identifying as non-text. The system subsequently annotates the results and provides the final representation in the web interface.

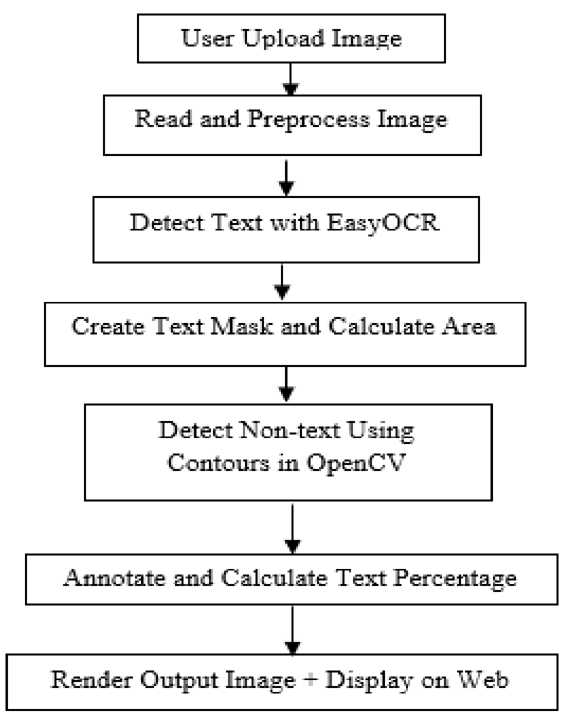

Fig.1 workflow of the proposed system

V. RESULTS AND DISCUSSION

The suggested system was evaluated on a variety of document images to assess the effectiveness of being able to separate text from non-text items. The outputs show clear, accurate detection of text regions, clear, accurate separation of non-text components, and the bounding box and graphical summary allow for easy interpretation.

5.1 Testing Process

The testing was performed just on document images, including reports, forms, and posters. Each of the uploaded images then followed through the proposed pipeline from document images to text information, by detecting the text regions using OCR, and then detecting non-text regions such as logos and shapes using contour analysis. The outputs were then visually checked against the ground truth labels that were created manually.

5.2 Performance Assessment

The assessment indicated that the proposed solution worked well with document images. The system correctly detected the text regions of the forms and reports, Contour analysis of the posters appropriately classified those components that were graphical (i.e. logos and shapes as non-text). The integration of OCR and contour analysis ensured robustness over different document images types.

5.3 Accuracy of text detection

The printed text present in this subset of document images was detected with nearly perfect accuracy. Bounding boxes were closely matched to words and paragraphs so that the textual content was clearly isolated. Some oddity was noted with certain more stylized headings but generally the recognition was stable and consistent for most document formats viewed.

5.4 Non-Text Isolation

The system was able to analyze the document images and identify and isolate the non-text components such as logos, shapes, stamps, and graphical objects (e.g. pie charts, or diagrams) as non-text regions using contour information. Given areas, which we used as thresholds, the system filtered out small background noise and small specks in the images, so we could develop results that were more meaningful, and to be able to tell the text that made it through the process from the graphics in the documents.

5.5 Output Visualization

The system produced intuitive and easy to interpret output.

- Text regions were defined in green bounding boxes labeled "Text."
- Non-text elements were defined in blue bounding boxes labeled "Non-Text."
- A pie chart was presented to provide a visual representation of the proportions of text and non-text in the image.

Fig. 2. Document image with annotations indicating detected text and non-text regions.

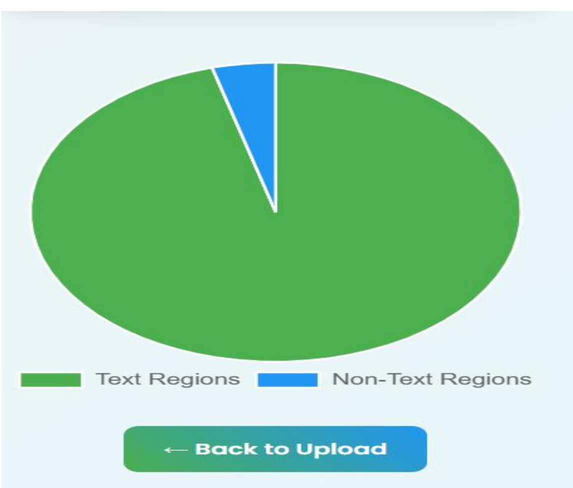

Fig. 3. Pie chart of distribution of text and non-text regions in the document

5.6 System Usability

Individuals with easy-to-use web interface that allowed them to upload those document images and view annotated results immediately. The output page contained both bounding box and graphical summaries making content interpretation a simple task. The web-based interface was designed to be simple and not require expertise from a technical perspective.

5.7 System Efficiency

The system was capable of processing each document image in just a few seconds, keeping the operation smooth and responsive. The software was lightweight, allowing for real-time analysis of uploaded files, making it feasible to operate in an ongoing state in document-based applications. The efficiency was consistent throughout since we have limited input images to document input images such as reports, forms, and posters. The efficiency was consistent as the layouts differed, the runtime did not vary significantly.

VI. CONCLUSION

This research has developed a functional system to isolate and differentiate between the text and non-text regions in document images. Through the integration of EasyOCR to recognize text and contour-based techniques to classify non-text regions, the system exhibited acceptable performance on various document types, including reports, forms, and posters. The annotated outputs were elegantly annotated with bounding boxes and summarized as pie charts, allowing for appropriate visual feedback of the process, while the Flask based interface increased usability and allowed users to quickly upload images or PDF files and provide immediate analysis from a web-based interface. While the overall assessment demonstrated that the system was accurate, lightweight and user-friendly in order to enable effective document analysis without technical expertise, this framework could easily support handwritten text and multilingual components, while accommodating for more intricate graphical elements that could expand usability in a larger variety of document processing applications.

REFERENCES

[1] Awasthi, P., et al. (2019). "Logo-assisted segmentation of advertisement overlays in document images." International Journal of Computer Vision and Pattern Recognition, 33(2), 112–119.

[2] Chaitra, Y. L., Kumar, R. D., & Kavyashree, M. (2023). "Text detection and recognition from scene images using RCNN and EasyOCR." International Journal for Research in Applied Science and Engineering Technology (IJRASET), 11(7), 1223–1228.

[3] Chen, Q., Zhang, L., & Wang, H. (2021). "Hybrid document analysis using semantic text detection and visual element segmentation." Journal of Intelligent Systems, 31(4), 327–336.

[4] Hassan, R., Batra, A., & Mehra, K. (2021). "Multi-modal document structure analysis using contour-based non-text detection and OCR." Journal of Document Intelligence, 18(3), 89–97.

[5] Kumar, S., Prasad, M., & Jain, A. (2020). "EAST and OpenCV based text and non-text classification from scanned reports." In Proceedings of the International Conference on Smart Vision Systems, 23(1), 45–52.

[6] Li, H., Yu, Z., & Feng, C. (2019). "Layout-aware text and image region segmentation for legal and resume documents." Pattern Recognition Letters, 129, 95–103.

[7] Patel, R., Shah, V., & Desai, A. (2019). "Edge-contour analysis for hybrid document layout processing." Journal of Information Processing, 27(1), 62–70.

[8] Reddy, P., & Srinivas, G. (2019). "Visual object detection in legal and financial documents using contour descriptors." International Journal of Legal Informatics, 10(2), 78–85.

[9] S. G. Santhosh *et al.*, "Real-Time Text Extraction and Classification from Bilingual Road Signboards Using OCR Engines," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 23s, pp. 3554–3563, 2024, ISSN: 2147-679921.

[10] S. G. Santhosh, *et al,* "Enhancing Machine Learning Methods for Robust Real-Time Text Classification of Bilingual Documents," International Journal of Innovative Research in Technology (IJIRT*)*, vol. 12, no. 3, pp. 42–48, Aug. 2025, ISSN: 2349-6002.

[11] Singh, A., & Awasthi, V. (2018). "Shape-based detection of brand logos in advertisement layouts using OpenCV." Journal of Image Processing and Recognition, 24(4), 211–219.

[12] Smelyakov, K., Ivanov, D., & Orlov, A. (2021). "A comparative study of EasyOCR and TesserOCR for text recognition in the presence of visual noise." Procedia Computer Science, 192, 34–41.

[13] Zhang, H., Wang, Y., & Lin, F. (2022). "Comparison of EasyOCR and Tesseract OCR on electronic and web-based documents." Journal of Image Processing and Applications, 32(3), 178–186.

[14] Zhang, J., Liu, T., & Chen, M. (2023). "Contour thresholding based on adaptive edge detection for document images." In Proceedings of the International Conference on Image Analysis and Processing, 47–54.

[15] Zhang, L., Zhao, Q., & Yu, M. (2022). "A comprehensive survey of segmentation methods such as thresholding, edge detection and contour segmentation." Journal of Digital Image Science, 29(1), 22–34.

[16] Zhang, L., Zhao, Q., & Yu, M. (2023). "Text detection and recognition based on deep learning: A survey of methods and applications." International Journal of Computer Vision Research, 31(2), 134–149.

[17] Zhang, W., Li, S., & Guo, K. (2022). "Evaluation of advanced text recognition tools against noisy and scanned documents." International Journal of Intelligent Vision, 18(4), 201–210.

[18] Zhang, W., Li, S., & Guo, K. (2023). "A comparative approach to evaluate properties of EasyOCR and Tesseract OCR capabilities across data structures." Procedia Computer Vision, 25(1), 95–103.

[19] Zhang, Y., Huang, L., & Chen, F. (2023). "Implementation and evaluation of EAST for text detection on natural scene images." Pattern Recognition Letters, 145, 78–85.

[20] Zhang, Y., Huang, L., & Chen, F. (2023). "Test assessment of EasyOCR extraction scenarios in document and web-based settings." Journal of Document Analysis, 12(2), 144–151.