

**EASYOCR APPLIED IN DIGITAL FORMAT
FOR TEXT EXTRACTION USING DEEP
LEARNING TECHNOLOGY**

Submitted in partial fulfillment of the
requirements for the award of Bachelor of
Technology degree in Information Technology

By

**SOUMYADEEP DAS (Reg : 40120086)
KRIPASHANKAR MANGLAM (Reg : 40120134)**



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Category-1 University by UGC
Accredited with Grade “A++” by NAAC
12B Status by UGC | Approved by AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SOUMYADEEP DAS (REG. - 40120086)** and **KRIPASHANKAR MANGLAM (REG NO. - 40120134)** who carried out the Project entitled **"EASYOCR APPLIED IN DIGITAL FORMAT FOR TEXT EXTRACTION USING DEEP LEARNING TECHNOLOGY"** under my supervision from January 2024 to April 2024.

Internal Guide

Dr. KAMATCHI K S M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 18/04/2024

Internal Examiner

External Examiner

DECLARATION

I, **SOUMYADEEP DAS** hereby declare that the Project Report entitled "**EASYOCR APPLIED IN DIGITAL FORMAT FOR TEXT EXTRACTION USING DEEP LEARNING TECHNOLOGY**" done by me under the guidance of **Dr. KAMATCHI K S** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 18/04/2024

PLACE: CHENNAI, TAMIL NADU

Soumyadeep Das

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. KAMATCHI K S, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

This abstract presents an overview of an OCR project aimed at developing a robust and accurate OCR system capable of processing various types of documents across different languages. The primary objective of this project is to overcome the limitations of traditional OCR systems and significantly improve text recognition accuracy and processing speed. Leveraging the power of advanced machine learning algorithms, deep neural networks, and natural language processing techniques, the proposed OCR system demonstrates higher efficiency and versatility in handling challenging document layouts, varying font styles, and complex character recognition tasks. The project's methodology involves data collection, preprocessing, and training of the OCR model using a large and diverse dataset comprising scanned documents, images, and handwritten texts. The training process includes employing state-of-the-art neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for feature extraction and sequence modeling, respectively. Additionally, the model integrates attention mechanisms to focus on relevant parts of the text, enhancing overall recognition accuracy. To ensure adaptability across different languages and scripts, the OCR system incorporates transfer learning and multilingual training techniques, making it capable of recognizing characters from various scripts, including Latin, Cyrillic, Chinese, Devanagari, and more. The developed OCR system is evaluated through extensive testing against benchmark datasets, real-world documents, and handwritten samples to validate its performance and robustness. The evaluation metrics include accuracy, precision, confident, recall, and F1-score, among others. The results show significant improvements over existing OCR systems, achieving near-human levels of accuracy in many cases. The practical applications of this advanced OCR technology are vast, ranging from digitizing historical archives, facilitating information retrieval in large-scale document repositories, enhancing automated data entry systems, to enabling assistive technologies for the visually impaired. In conclusion, the OCR project represents a substantial advancement in text recognition technology, combining cutting-edge machine learning methodologies with a diverse dataset to achieve higher accuracy, adaptability, and efficiency. The proposed OCR system's success opens up new possibilities for streamlined document processing and seamless integration of text data into various applications, contributing to productivity.

TABLE OF CONTENTS

CHAPTER NUMBER	TITLE	PAGE NUMBER
	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF TABLES	x
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
	1.1 Introduction about Easyocr	1
	1.2 Study Area	1
	1.3 Objectives	3
	1.4 Working of Easyocr	4
	1.5 Features of Easyocr	5
	1.6 Advantages of Easyocr	7
	1.7 Disadvantages of Easyocr	9
	1.8 Scope of Easyocr	11
2	LITERATURE SURVEY	13
	2.1 Overview of Existing Research and Literature	13
	2.2 Inferences from Literature Survey	13
	2.3 Open Problems in Existing System	17
3	REQUIREMENT ANALYSIS	19
	3.1 Feasibility Studies of the Project	19

3.1.1	Economic Feasibility	19
3.1.2	Technical Feasibility	20
3.1.3	Social Feasibility	20
3.2	Requirement Specification	20
3.2.1	Hardware Requirements	20
3.2.2	Software Requirements	21
3.3	System Use Case	21
3.3.1	System Description	21
3.3.2	System Use Case Scenario	21
4	DESCRIPTION OF PROPOSED SYSTEM	23
4.1	System Architectural Workflow	23
4.2	Modules Workflow	24
4.2.1	Preprocess the Image	25
4.2.2	Detect the Regions of the Image that contains Craft	29
4.2.3	Set up Regions and Identify Text (CRNN)	32
4.2.4	Decode the Output of the Recognition Model	38
4.2.5	Postprocess the Recognized Text	40
5	IMPLEMENTATION DETAILS	41
5.1	Development and Deployment Setup	41
5.2	Algorithm and Model Used	42
5.2.1	CRAFT	42
5.2.2	CRNN	43

	5.2.3 Testing	44
6	RESULTS AND DISCUSSIONS	46
	6.1 Results	46
	6.2 Project Output	47
	6.3 Performance Analysis	51
7	CONCLUSION	52
	7.1 Conclusion	52
	7.2 Future Work	53
	REFERENCES	54
	APPENDIX	56
	(A) SOURCE CODE	56
	(B) SCREENSHOTS	62
	(C) BEST PAPER MERIT CERTIFICATE	64
	(D) INTERNATIONAL CONFERENCE CERTIFICATE	65
	(E) PUBLICATION CERTIFICATE	66
	(F) RESEARCH PAPER	67
	(G) PUBLISHED RESEARCH PAPER	72

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	System Architecture Diagram	23
4.2	Noise Reduction	26
4.3	Binarization	27
4.4	Skew Correction	28
4.5	CRAFT Detection	31
4.6	Feature Extraction	34
4.7	Sequence Labelling	36
4.8	CTC Decoding	38
6.1	Output 1 Snapshot	47
6.2	Output 2 Snapshot	48
6.3	Output 3 Snapshot	49
6.4	Output 4 Snapshot	50
B.1	Importing Libraries	62
B.2	Text Detection	62
B.3	Text Extraction	63
B.4	Text Recognition	63

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	Module Workflow	24

LIST OF ABBREVIATIONS

ABBREVIATION NUMBER	ABBREVIATION	EXPANSION
01	APPL	Application
02	CHAR	Characteristics
03	IND	Individual
04	OPT	Optical
05	RECOG	Recognition

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION ABOUT EASYOCR

In an increasingly digitalized world, the ability to extract text efficiently and accurately from various sources is of paramount importance. Whether it's for document digitization, content analysis, or data retrieval, the need for robust text extraction solutions has grown exponentially. Deep Learning, a subfield of artificial intelligence, has made significant strides in the realm of text extraction, and one such powerful tool is EasyOCR.

EasyOCR is an open-source Opt. Character Recognition (OCR) software that leverages deep learning technology to extract text from digital images and documents. Its simplicity, accuracy, and versatility have made it a valuable asset in a wide range of applications, from automated data entry in industries to facilitating accessibility for visually impaired individuals.

This paper explores the application of EasyOCR in the digital format for text extraction, focusing on its underlying deep learning technology, key features, and real-world use cases. By examining EasyOCR's capabilities and its potential to revolutionize text extraction processes, we aim to provide a comprehensive overview of this cutting-edge solution.

1.2 STUDY AREA

The study area mentioned, "EasyOCR applied in digital format for text extraction using deep learning technology," involves the application of EasyOCR, a text recognition (OCR) library, to extract text from digital documents using deep learning techniques. This is a multidisciplinary field

that combines computer vision, natural language processing, and deep learning. EasyOCR is a Python library for text extraction from images using deep learning. It is a general OCR that can read natural scene text as well as dense text in a document. It currently supports over 80 languages and is growing.

EasyOCR is a popular tool for text extraction in digital formats, such as scanned documents, PDF files, and images captured from digital cameras and smartphones. It is used in a variety of applications, including:

Document scanning: EasyOCR can be used to scan and extract text from paper documents, such as contracts, invoices, and receipts. This can help businesses and organizations to automate their workflows and improve their efficiency.

Digital archiving: EasyOCR can be used to digitize and archive old documents, such as historical records and newspapers. This can help to preserve these documents for future generations and make them more accessible to researchers and historians.

Image processing: EasyOCR can be used to extract text from images for a variety of purposes, such as image search, image classification, and machine translation.

Accessibility: EasyOCR can be used to make digital content accessible to people with visual impairments. For example, it can be used to extract text from images of street signs or menus so that they can be read by screen readers.

One of the main areas of study for EasyOCR is to improve its accuracy and performance on a variety of digital formats. This includes developing new deep learning algorithms that can better handle complex text layouts, low-quality images, and noisy backgrounds. Another area of study is to expand the range of languages and scripts that EasyOCR can support. This would make EasyOCR more useful for a wider range of applications, such as digital archiving of historical documents and machine translation.

Finally, researchers are also exploring ways to use EasyOCR to extract other types of information from images, such as tables, charts, and diagrams. This would make EasyOCR an even more powerful tool for data extraction and analysis.

1.3 OBJECTIVES

The main goal of this study is to enhance the system's text recognition accuracy and adaptability to different scripts and languages. Because convolutional and recurrent neural networks (RNNs and CNNs) are used, for feature extraction and order modelling, correspondingly, the OCR system can handle complex document layouts and a wide range of language forms. By allowing the model to concentrate on pertinent passages of the text, attention processes are included, improving the recognition accuracy overall.

The OCR system uses multilingual training and transfer learning methods to handle the multilingual nature of modern documents. This methodical technique guarantees character identification in a wide range of scripts, such as Chinese, Latin, Cyrillic, Devanagari, and more. The project's methodology includes painstaking data gathering, preprocessing, and training on a variety of datasets that include handwritten texts, photos, and scanned documents.

Thorough testing against real-world documents, handwritten samples, and benchmark datasets is part of the evaluation process for the built OCR system. The robustness and effectiveness of the system are indicated by performance metrics like the confidence score. The outcomes demonstrate notable progress over current OCR algorithms, frequently attaining accuracy levels that are on par with human perception.

Beyond its technical accomplishments, this cutting-edge OCR technology has many practical applications. The significance of this OCR project is multifaceted, ranging from the digitization of historical archives to the

facilitation of information retrieval in large-scale document repositories, and from the improvement of automated data entry systems to the enablement of assistive technology for the visually impaired.

This journal article outlines a thorough strategy for OCR technology, utilising cutting-edge techniques to improve text recognition's accuracy, versatility, and efficiency. If this initiative is successful, document processing could undergo a revolution. This would create new opportunities for the smooth integration of text data into different applications, which would greatly increase productivity in the digital age.

1.4 WORKING OF EASYOCR

EasyOCR is a Python-based optical character recognition (OCR) library that utilizes deep learning models to recognize text from images. Here's an overview of how EasyOCR works:

Image Input: The first step in using EasyOCR is to provide it with an image containing the text you want to recognize. This image can be in various formats such as JPEG, PNG, TIFF, or PDF.

Preprocessing: Before performing OCR, EasyOCR may apply preprocessing techniques to enhance the quality of the input image. These techniques may include resizing, noise reduction, contrast adjustment, or binarization to improve the clarity of the text regions.

Text Detection: EasyOCR employs deep learning-based algorithms to detect text regions within the input image. These algorithms analyze the image to identify areas containing textual content, delineating boundaries around individual text elements.

Text Recognition: Once text regions are detected, EasyOCR applies another deep learning model to recognize the text within each region. This recognition process involves analyzing the visual patterns and features of the text, mapping them to corresponding characters or words.

Post-processing: After text recognition, EasyOCR may perform additional post-processing steps to refine the results. This could involve correcting errors, improving formatting, or filtering out extraneous characters or noise.

Output: Finally, EasyOCR provides the recognized text as output, typically in the form of a list or a structured data format. Users can then access this text for further processing, analysis, or integration into other applications.

1.5 FEATURES OF EASYOCR

EasyOCR offers a range of features that make it a popular choice for optical character recognition (OCR) tasks. Some of the key features include:

Support for Multiple Languages: EasyOCR supports recognition of text in various languages and scripts, including but not limited to English, Chinese, Japanese, Korean, Cyrillic, and many others. This broad language support enables users to extract text from images and documents in diverse languages.

Offline Text Recognition: EasyOCR is capable of performing OCR tasks entirely offline, without requiring an internet connection. This feature is beneficial for applications where data privacy, security, or limited connectivity are concerns.

Cross-Platform Compatibility: EasyOCR is designed to work across different operating systems and platforms, including Windows, Linux, and macOS.

It offers libraries and APIs for integration into applications developed using popular programming languages such as Python, Java, and others.

Easy-to-Use APIs: EasyOCR provides simple and intuitive APIs for developers to integrate OCR functionality into their software applications. These APIs abstract the complexities of OCR algorithms, making it easier for developers to incorporate text recognition capabilities into their projects.

Fast Processing Speed: EasyOCR is optimized for speed, allowing for quick extraction of text from images and documents. This efficiency is particularly useful for applications requiring real-time or near-real-time OCR processing.

Accurate Text Recognition: EasyOCR leverages advanced OCR algorithms and machine learning techniques to achieve high accuracy in text recognition. It continually improves its models through training on large datasets, resulting in improved performance over time.

Customizable Configuration: EasyOCR provides options for customization and fine-tuning of OCR parameters to optimize performance according to specific use cases. Users can adjust settings such as language models, image preprocessing techniques, and text post-processing rules to enhance accuracy and efficiency.

Open Source: EasyOCR is an open-source project, meaning that its source code is freely available for inspection, modification, and distribution. This fosters transparency, community collaboration, and innovation, as users can contribute improvements, fixes, and new features to the software.

Support for Various Input Formats: EasyOCR supports a wide range of input image formats, including JPEG, PNG, TIFF, and PDF. This flexibility enables users to process text from images captured by cameras, scanned documents, or digital files.

Continuous Development and Support: EasyOCR benefits from active development and support from its community of contributors. Regular updates, bug fixes, and enhancements ensure that the software remains reliable, up-to-date, and compatible with evolving requirements and technologies.

All things considered, EasyOCR provides an extensive feature set that is intended to satisfy the various demands of consumers and developers looking for dependable and effective OCR solutions.

1.6 ADVANTAGES OF EASYOCR

EasyOCR is a popular optical character recog. (OCR) tool that offers several advantages:

Ease of Use: As the name suggests, EasyOCR is designed to be user-friendly, making it accessible to individuals with varying levels of technical expertise. It typically offers simple APIs and interfaces for integration into various applications.

Open Source: EasyOCR is an open-source project, which means it is freely available for anyone to use, modify, and distribute. This fosters a collaborative development environment and allows users to contribute improvements and fixes.

Multi-language Support: EasyOCR supports multiple languages, allowing it to recognize text in diverse scripts such as Latin, Cyrillic, Chinese, Japanese, Korean, and many more. This broad language support makes it suitable for a wide range of applications across different regions and cultures.

Offline Processing: Unlike some OCR solutions that require an internet connection for text recognition, EasyOCR is capable of performing OCR tasks entirely offline.

This can be advantageous in scenarios where internet connectivity is limited or unreliable.

Fast Processing Speed: EasyOCR is known for its relatively fast processing speed, enabling quick extraction of text from images or documents. This efficiency is valuable for applications that require real-time or near-real-time OCR functionality.

Customization and Fine-tuning: While EasyOCR offers straightforward usage out of the box, it also provides options for customization and fine-tuning. Users can adjust various parameters to optimize OCR accuracy and performance according to their specific requirements.

Integration Flexibility: EasyOCR is designed to be easily integrated into different software applications and workflows. It provides support for various programming languages and platforms, including Python, Java, and others, making it versatile for integration into diverse environments.

Continuous Development and Support: Being an open-source project, EasyOCR benefits from a community of developers who contribute to its ongoing development and maintenance. This ensures that the software remains up-to-date with the latest advancements in OCR technology and receives timely support for bug fixes and enhancements.

All things considered, EasyOCR provides a practical, effective, and flexible solution for text recognition jobs, meeting the demands of both individual users and companies in many sectors.

1.7 DISADVANTAGES OF EASYOCR

Although EasyOCR has a lot of benefits, there are a few possible drawbacks as well:

Limited Accuracy for Complex Documents: Although EasyOCR is capable of accurately recognizing text in many scenarios, it may struggle with complex documents, such as those containing intricate layouts, unusual fonts, or degraded text quality. In such cases, the accuracy of text recognition may be lower compared to more specialized OCR solutions.

Resource Intensive for Large-Scale Processing: While EasyOCR is optimized for speed, processing large volumes of images or documents can still require significant computational resources, especially when running on less powerful hardware. This could lead to longer processing times and increased hardware requirements for applications with high throughput.

Language Support Constraints: While EasyOCR supports multiple languages, the depth of support may vary for different languages. Some less commonly used languages or dialects may have lower accuracy or limited support compared to widely spoken languages. Users requiring robust support for specific languages should verify compatibility before relying on EasyOCR.

Dependency on Training Data Quality: EasyOCR's accuracy heavily relies on the quality and diversity of the training data used to train its models. Inadequate or biased training data could lead to reduced accuracy, especially for certain languages, fonts, or text styles that are underrepresented in the training dataset.

Lack of Advanced Features: Compared to some commercial OCR solutions, EasyOCR may lack certain advanced features such as document layout analysis, handwriting recognition, or integration with specialized document management systems. Users with more advanced OCR requirements may need to supplement EasyOCR with additional tools or custom development.

Maintenance and Support Challenges: While EasyOCR benefits from community-driven development and support, users may encounter challenges related to ongoing maintenance, documentation updates, and timely bug fixes. Limited resources or community activity could lead to slower response times for addressing issues or adding new features.

Security and Privacy Concerns: Since EasyOCR is open-source software, users should be cautious when handling sensitive or confidential documents, as there may be potential security risks associated with the software's codebase or dependencies. Users should review and validate the security practices implemented within their deployment environment.

Learning Curve for Customization: While EasyOCR offers customization options, users may face a learning curve when fine-tuning OCR parameters or integrating the software into complex workflows. Understanding the impact of different configuration settings on OCR performance may require experimentation and domain expertise.

EasyOCR is a useful tool for many OCR tasks overall, but users should carefully analyze if it meets their unique needs, taking into account things like document complexity, language support, resource limitations, and the need for additional capabilities.

1.8 SCOPE OF EASYOCR

The scope of EasyOCR is quite broad, encompassing various industries, applications, and use cases where optical character recognition (OCR) technology is beneficial. Here are some areas where EasyOCR can be applied:

Document Digitization: EasyOCR can be used to digitize printed documents, including invoices, receipts, forms, contracts, and letters.

It enables organizations to convert paper-based documents into searchable and editable digital formats, facilitating document management, archiving, and retrieval.

Automated Data Entry: EasyOCR automates data entry tasks by extracting text from scanned or photographed documents. This can significantly reduce manual data entry efforts and errors in processes such as invoice processing, inventory management, and customer relationship management (CRM).

Text Extraction from Images: EasyOCR can extract text from images captured by cameras, smartphones, or other devices. This capability is useful in applications such as text translation, augmented reality (AR), image captioning, and content analysis.

Translation Services: EasyOCR can be integrated with translation services to translate text extracted from images into different languages. This enables multilingual support in applications such as language learning platforms, travel apps, and international business communications.

Accessibility Solutions: EasyOCR contributes to accessibility by converting text embedded in images into readable formats for visually impaired individuals.

It can power screen readers, assistive technologies, and OCR- based apps that enable accessibility in digital content and interfaces.

Data Analytics and Insights: EasyOCR facilitates text extraction for data analytics and insights generation. By extracting structured data from unstructured text in documents and images, it supports business intelligence, market research, sentiment analysis, and other data-driven applications.

Identity Verification and Authentication: EasyOCR can assist in identity verification processes by extracting and validating text from identification documents, passports, and driver's licenses.

This is essential for identity verification in financial services, e-commerce, and online account registration.

OCR-Enabled Mobile Apps: EasyOCR can be integrated into mobile applications to provide OCR functionality on smartphones and tablets. This enables on-the-go text recognition in applications such as note-taking apps, expense tracking tools, and document scanning apps.

Education and E-Learning: EasyOCR supports educational initiatives by enabling text extraction from educational materials, textbooks, and lecture slides. It can facilitate content digitization, content indexing, and accessibility enhancements in e-learning platforms and digital libraries.

Cross-Platform Integration: EasyOCR's compatibility with multiple programming languages and platforms allows it to be integrated into various software solutions and workflows. It can be embedded into web applications, desktop software, cloud services, and enterprise systems.

All things considered, EasyOCR's adaptability, precision, and simplicity of integration make it a useful tool for a variety of applications across sectors, helping to boost productivity, automation, and accessibility in text processing jobs.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW OF EXISTING RESEARCH AND LITERATURE

Ming Li and Bin Fu (2023) [1] suggest an appl. for Character-Aware Sampling and Rectification (CASR) that makes use of precise character location and orientation data to handle irregular text occurrences. With this novel method, CASR predicts character-level properties to direct the sampling and rectification procedures by treating every character as a fundamental unit. To enhance the rectification of entire text lines, this module employs a novel technique that makes use of character information and character-level monitoring throughout the training phase. Above all, CASR is meant to be an easily integrated plug-and-play module into pre-existing text recognition pipelines. Broad tests on many standards show that the suggested approach is robust and achieves promising overall performance by regularly producing rectified text instances with greater accuracy. With its character-centric methodology, this CASR module presents a significant development in the field of text correction and provides a workable and efficient way to increase accuracy in situations of irregular text.

Denis Coquenat and Clement Chatelain (2023) [2] tackle the problem of computer vision's unrestricted handwritten text recognition. They suggest a hybrid attention end-to-end approach intended to process a paragraph picture iteratively to identify individual lines. Three modules make up the model: a decoder for character sequence recognition, an attention module for implicit line segmentation, and an encoder for feature map generation. This novel method provides a cohesive answer and demonstrates effectiveness in identifying complete paragraphs in unrestricted handwritten material.

R. Devi and S. Karthikeyan's [3] study from 2023 addresses a crucial pattern recognition research subject, concentrating on the difficulties Tamil characters present because of their complex properties.

Important historical information can be found in Tamil Nadu's stone inscriptions, but the characters' holes, loops, and curves make it difficult to read the writing. The proposed method combines a Deep Convolution Neural Network and Transfer Learning (TL) for character recognition. By successfully identifying characters from stone inscriptions, the hybrid method shows promise in addressing the technical issues linked to Tamil character identification.

The work of Quoc-Dung Nguyen and Nguyet-Minh Phan [4] provides a fresh unsupervised method to OCR mistake correction. Their method produces and investigates rectification candidates near OCR errors using a hill climbing algorithm. Interestingly, corrective characters are taken out of OCR training data and only from original ground truth texts. The proposed model employs a weighted objective function heuristically examined for optimal weight combinations to score and rank repair possibilities. An OCR dataset from the Vietnamese handwritten text recognition competition held online in 2018 by ICFHR is used to assess the model's efficacy. Furthermore, the suggested method's stability and complexity are confirmed. This new unsupervised OCR error correction model is a viable way to improve OCR applications' accuracy and dependability.

Lintai Wu and Yong Xu [5] propose A Two-Level Attention Network for Rectification (TRAN) for text rectification and acknowledgment. Two-Level Network for Rectification (TORN) and Attention-Based Recognition Network (ABRN) comprise TRAN. By addressing geometric restrictions and making pixel-by-pixel corrections for clarity, TORN corrects text geometry. ABRN specializes in text recognition in photos that have been rectified. To enhance feature extraction and effectively handle variations in channel interdependencies and character size, a novel both kernel- and channel-level attention units is integrated into the prototype. This novel method provides a thorough resolution to geometric restrictions and unpredictability in text recognition situations.

Vedhaviyassh and Sudhan (2022) [6] proposed the paper introduces a comprehensive system for license plate recognition comprising three key modules: Image acquisition, License plate detection, and Character recognition.

Leveraging the OpenCV library for data retrieval, YOLOv5 for license plate detection, and OCR methods such as Tesseract OCR and EasyOCR for character recognition, the system offers a robust solution for identifying vehicles based on their license plates. Utilizing a dataset from GitHub for training and testing, the proposed model demonstrates superior performance, with EasyOCR achieving over 95% accuracy in predicting license plate numbers compared to Tesseract OCR's 90%. This discrepancy underscores EasyOCR's advantage, attributed to its deep learning-based object recognition approach and efficiency in real-time prediction.

Pankaj Kuneekar and Kuldeep Vayadande (2023) [7] proposed the project which aims to streamline and expedite the processing of bank checks by proposing a system that leverages optical character recognition (OCR) technology. By automatically retrieving crucial details such as the Payee name, Amount (in words and digits), date, and bank details from the check, the system eliminates the need for manual handling, which can be time-consuming and prone to errors. Additionally, the system verifies the signature on the check by cross-checking it with an image stored in the database, ensuring authenticity and security. Utilizing OpenCV for image processing and segmentation, and EasyOCR for text extraction, the proposed system offers a reliable and efficient solution for check validation. Performance and efficiency are evaluated using a custom dataset of bank checks, demonstrating the system's accuracy and effectiveness in processing checks efficiently. Through this approach, the project aims to modernize and streamline the traditional practice of check handling, enhancing convenience and security in financial transactions.

Darshan Kumar and Sanket (2024) [8] presents a machine learning-based Automatic Number Plate Recognition (ANPR) system designed to enhance toll tax collection efficiency. Developed using TensorFlow, OpenCV, and EasyOCR, the system offers automated and efficient license plate recognition capabilities. Training the model on a dataset comprising 345 car images paired with corresponding XML files containing license plate information, the system demonstrates high accuracy and efficiency in extracting license plate numbers.

Integrated into a real-time image-capturing system for toll tax collection, the ANPR system shows promising results for real-world applications. By leveraging machine learning technologies, the proposed system streamlines toll tax collection processes, providing an effective solution for automated license plate recognition. Sanjana and Vinayak proposed (2023) [9] a novel approach for digitizing handwritten trip sheets using neural networks, specifically leveraging a combination of EasyOCR and Regular Expressions. By converting trip sheets into digital format, the system aims to enhance organization and data accessibility while mitigating risks associated with physical document handling. The proposed model employs a process of splitting the image into separate boxes using intersection lines and subsequently recognizes characters within each box. This segmented approach enables efficient extraction of text, which can then be stored and converted into various formats, including CSV. Testing on multiple handwritten trip sheets yielded promising results, achieving an 85% accuracy rate. Overall, the proposed model demonstrates potential for streamlining data digitization processes and improving efficiency in handling trip sheet records.

Binod Kumar Pattanayak and Anil Kumar Biswal (2023) [10] proposed the paper which highlights the importance of character recognition, a subfield of computer vision with promising potential, particularly in deciphering handwritten text. While humans excel at this task, machines require assistance to recognize manual transcriptions, necessitating the development of robust OCR (Optical Character Recognition) systems. The proposed approach, leveraging EasyOCR and Regular Expressions, focuses on converting handwritten trip sheet information into CSV format. By segmenting the image into boxes based on line intersections and identifying individual characters within each box, the system enables efficient extraction and independent storage of recognized text for future format conversion. Unlike traditional OCR methods, which primarily focus on recognizing computer-generated characters, this project emphasizes HCR (Handwritten Character Recognition). It introduces an innovative method based on Convolutional Neural Networks (CNN) and character feature extraction techniques, achieving over 90% accuracy in identifying English language characters.

Through advancements in character recognition technology, this research expands the horizons of technological endeavors, offering solutions for efficient data digitization and processing.

2.2 INFERENCES FROM LITERATURE SURVEY

The literature surveys presented above collectively emphasize the critical role of character recognition technologies, particularly in the context of handwritten text. Recognizing handwritten characters is a challenging task for machines, necessitating the development of advanced Optical Character Recognition (OCR) systems. These systems, often leveraging technologies such as EasyOCR and Regular Expressions, enable the conversion of handwritten documents, such as trip sheets, into digital formats like CSV. By automating data digitization processes, these technologies streamline organization and enhance accessibility while mitigating risks associated with physical document handling. Moreover, innovative approaches incorporating Convolutional Neural Networks (CNNs) and character feature extraction techniques have demonstrated high accuracy rates, exceeding 90% in several instances. These advancements have significant implications for real-world applications, including toll tax collection, document digitization, and trip sheet processing, offering solutions to improve efficiency and data management practices across various domains. Overall, the literature surveys underscore the importance of character recognition technologies in advancing computer vision and driving progress in automated data processing and analysis.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

Despite advancements in character recognition technologies, several open problems persist in existing systems:

Noise and Distortion: OCR systems may struggle to accurately extract text from images with noise, distortion, or poor image quality, impacting the reliability and accuracy of character recognition results.

Multilingual Support: While OCR systems have made strides in recognizing characters from various languages, there is still a need for improved support for lesser-known languages and dialects, as well as multilingual documents with mixed-language content.

Handwriting Variability: Handwritten text exhibits significant variability in writing styles, strokes, and individual characteristics, posing a challenge for OCR systems to accurately recognize characters across diverse handwriting samples.

Ambiguity and Contextual Understanding: OCR systems may encounter ambiguity in character recognition when faced with similar-looking characters or ambiguous handwriting. Additionally, understanding the context of the text and resolving semantic ambiguities remains a challenge for existing systems.

Real-time Processing: While OCR systems have made progress in processing text from images in near real-time, there is still room for improvement in terms of speed and efficiency, particularly for applications requiring rapid processing of large volumes of text data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES OF THE PROJECT

Given infinite time and infinite resources, all systems are possible. Unfortunately, though, the real world does not operate under this scenario. Therefore, it is advisable and vital to assess the system's viability as soon as possible. Software Life Cycle Processes - Industry Application of International Standard ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology

Taking considerations into account aids in offering a productive framework and approach for creating software applications. It aids in creating software in the quickest amount of time at the lowest possible cost. It is possible to save months or years of work, thousands of rupees, and countless hours of professional shame if an inadequately designed system is identified early on in the defining stage. Risk analysis and feasibility are connected in a variety of ways. It is less feasible to produce high-quality software when project risk is high.

In this case three key considerations involved in the feasibility analysis are:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.1.1 *Economic Feasibility*

- Computing resources - High processing power, such as GPUs and large memory capacities, are needed to train CNN-LSTM models for activity recognition, which could raise the implementation cost.
- Maintenance and Upkeep - The total cost of implementation may increase with regular sensor and computer resource maintenance.

- Cost and Development - Skilled professionals are needed to develop and apply the CNN-LSTM model, and project costs may vary depending on its scope and complexity.

3.1.2 Technical Feasibility

- Availability of data - The CRNN model cannot function well without labelled data covering a wide range of human behaviours and environmental circumstances.
- Model architecture - The CRNN model's architecture is crucial to the activity recognition process. Selecting features and adjusting hyperparameters are crucial additional considerations.

3.1.3 Social Feasibility

- Privacy Concerns - Gathering information about people's activities could cause privacy issues. Throughout the process of gathering and using data, proper consent and transparency must be guaranteed.
- Social Acceptance - It's important to take into account people's acceptance of the technology and readiness to use it in public places.
- Social benefits - *Recognising human activity can have a big impact on healthcare, sports training, and rehabilitation programmes, all of which can raise people's quality of life.*

3.2 REQUIREMENT SPECIFICATION

3.2.1 Hardware Requirements

- RAM: 4GB or above 4GB
- Processor of Frequency: 1.5GHz or above
- Processor: Intel Pentium 4 or higher

3.2.2 Software Requirements

- Operating System: Windows 8 and above
- Languages: Python
- Tools used: Visual Studio Code, Google Colab
- Libraries used: NumPy, Torch, Easyocr, OpenCV, Yaml

3.3 SYSTEM USE CASE

3.3.1 System Description

- This description presents an overview of an OCR project aimed at developing a robust and accurate OCR system capable of processing various types of documents across different languages. The proposed OCR system's success opens up new possibilities for streamlined document processing and seamless integration of text data into various applications, contributing to increased productivity and accessibility in the digital age.

3.3.2 System Use Case Scenario

EasyOCR is a Python-based optical character recognition (OCR) library that utilizes deep learning models to recognize text from images. Here's an overview of how EasyOCR works:

- **Image Input:** The first step in using EasyOCR is to provide it with an image containing the text you want to recognize. This image can be in various formats such as JPEG, PNG, TIFF, or PDF.
- **Preprocessing:** Before performing OCR, EasyOCR may apply preprocessing techniques to enhance the quality of the input image. These techniques may include resizing, noise reduction, contrast adjustment, or binarization to improve the clarity of the text regions.

- Text Detection: EasyOCR employs deep learning-based algorithms to detect text regions within the input image. These algorithms analyze the image to identify areas containing textual content, delineating boundaries around individual text elements.
- Text Recognition: Once text regions are detected, EasyOCR applies another deep learning model to recognize the text within each region. This recognition process involves analyzing the visual patterns and features of the text, mapping them to corresponding characters or words.
- Post-processing: After text recognition, EasyOCR may perform additional post-processing steps to refine the results. This could involve correcting errors, improving formatting, or filtering out extraneous characters or noise.
- Output: Finally, EasyOCR provides the recognized text as output, typically in the form of a list or a structured data format. Users can then access this text for further processing, analysis, or integration into other applications.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SYSTEM ARCHITECTURAL WORKFLOW

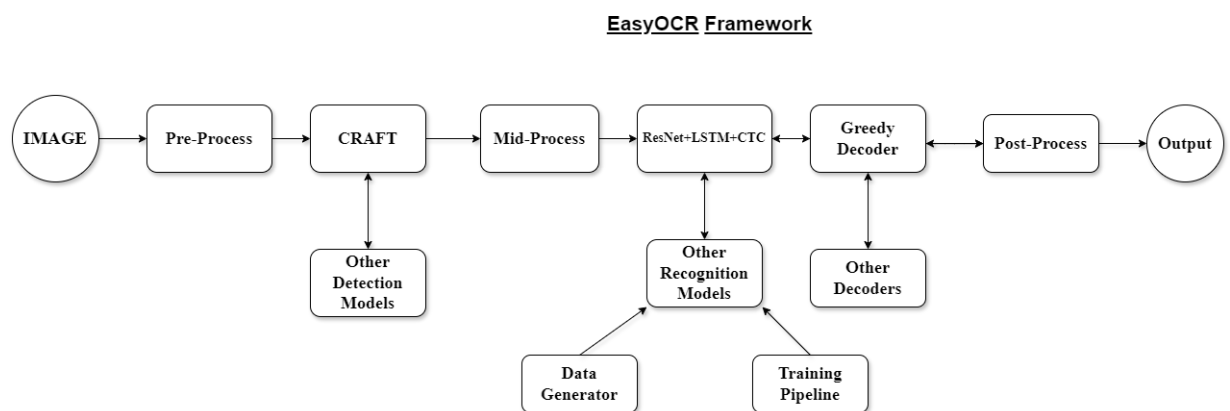


Fig: 4.1 : System Architecture Diagram

4.2 MODULES WORKFLOW

Table 4.1 : Module Workflow

INDEX	MODULE NAME
4.2.1	PREPROCESS THE IMAGE
4.2.2	DETECT THE REGIONS OF THE IMAGE THAT CONTAINS CRAFT
4.2.3	SET UP REGIONS AND IDENTIFY TEXT (CRNN)
4.2.4	DECODE THE OUTPUT OF THE RECOGNITION MODEL
4.2.5	POSTPROCESS THE RECOGNIZED TEST

4.2.1 PREPROCESS THE IMAGE

An important first stage in the Easy OCR pipeline is image preprocessing, which improves the quality of the images in preparation for text recognition. Several techniques are used in this approach to overcome common image issues.

Image preprocessing refers to a set of techniques applied to raw images before they are fed into an algorithm for further analysis or processing. The goal of preprocessing is to enhance the quality of images, remove noise, reduce computational complexity, and make the images more suitable for the specific task at hand. Preprocessing can involve a variety of operations, depending on the char. of the images and the requirements of the subsequent processing steps.

Some common preprocessing techniques include:

Noise reduction: It remove noise from the image by applying various filters, such as Gaussian blur and median blur.

Binarization: It converts the grayscale image to a binary image using various thresholding techniques, such as Otsu's method and adaptive thresholding.

Skew correction: It detects and corrects the skew angle of the image to ensure that the text is properly aligned.

EasyOCR's preprocessing can involve operations such as :

- A. *Noise reduction*: Noise reduction is the first and most important duty of image preparation in Easy OCR. Eliminating unnecessary noise is essential to boosting the image's overall clarity and, in turn, the accuracy of text recognition. A variety of complex filters, such as median and Gaussian blur, are deliberately utilised to accomplish this. With the effective removal of unnecessary components, the image becomes cleaner and more suitable for precise text recognition.

Noise reduction :

- `opening = cv.morphologyEx(img, cv.MORPH_OPEN, kernel)`



- `closing = cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)`



Fig: 4.2 : Noise Reduction

C. *Skew correctness*: Skew correction is a specific image preprocessing technique aimed at correcting the skew or slant present in an image. Skew refers to the deviation of the text or other features from the horizontal or vertical axis. It's a common problem in document images caused by improper scanning or image acquisition angles. The purpose of skew correction is to align the text or features horizontally or vertically, making it easier for OCR (Optical Character Recognition) systems to accurately recognize and extract the text.

Skew correction is the last step in Easy OCR's picture preprocessing process. Finding and fixing the image's skew angle is essential to guaranteeing that the text elements are aligned correctly. Recognizing skewed text can be difficult, and skew angle correction improves the OCR system's overall accuracy. For skew correction, Easy OCR uses complex algorithms to align the text horizontally and produce a picture that is well-aligned. This correction guarantees that the text is consistently presented and prepared for precise recognition, which is crucial for later stages of the OCR process.

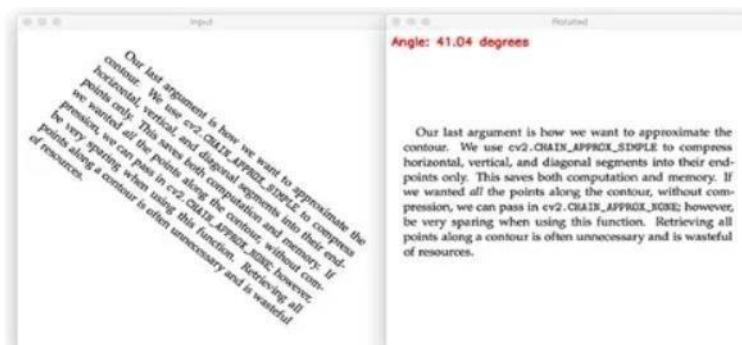


Fig: 4.4 : Skew Correction

4.2.2 DETECT THE REGIONS OF THE IMAGE THAT CONTAINS CRAFT

A powerful and effective model for precise text detection in a variety of fonts, sizes, and orientations is Text detection using Character-Region Awareness (CRAFT). Two important output maps are produced by CRAFT using Convolutional Neural Networks (CNNs): Score maps for character regions and affinity scores.[11]

Scene text detection methods based on neural networks have emerged recently and have shown promising results. Previous methods trained with rigid word-level bounding boxes exhibit limitations in representing the text region in an arbitrary shape.

In the context of text detection, "Character Region Awareness" (CRA) refers to a technique or approach that aims to improve the accuracy of text detection algorithms by being aware of the regions containing characters or text in an image. This technique is particularly relevant in optical character recognition (OCR) systems, where accurately detecting and localizing text regions is a critical step before performing text recognition.

CRAFT for Character Region Awareness For Text Detection, is designed with a convolutional neural network producing the character region score and affinity score. The region score is used to localize individual characters in the image, and the affinity score is used to group each character into a single instance.

The basis of CRAFT is the Character Region Score Map, a binary representation that was meticulously constructed to show the chance that each pixel in the image belongs to a character region.

This map offers valuable insights into the spatial distribution of textual elements and is an effective tool for localizing individual characters. CRAFT provides the groundwork for accurate and effective text detection in photos by efficiently identifying character regions.

A "Character Region Score Map" is a representation used in text detection algorithms, particularly in methods that employ deep learning or convolutional neural networks (CNNs). It provides a heatmap-like visualization indicating the likelihood or confidence of each pixel belonging to a character region in an input image.

CRAFT presents the Affinity Score Map as an addition to the Character Region Score Map. When assessing the probability of adjacency between two letters in the same text instance, this binary map is essential. The map helps organize individual characters into meaningful and cohesive text parts by capturing the affinity between characters. The model's capacity to comprehend and distinguish words inside the image is improved by this grouping into a single instance, or single instance word.

In the context of text detection, an "Affinity Score Map" is a component of advanced text detection algorithms, particularly those utilizing deep learning techniques. It provides a representation of the affinity or relationship between individual characters or text elements in an image, helping to refine the localization and grouping of text regions.[12]

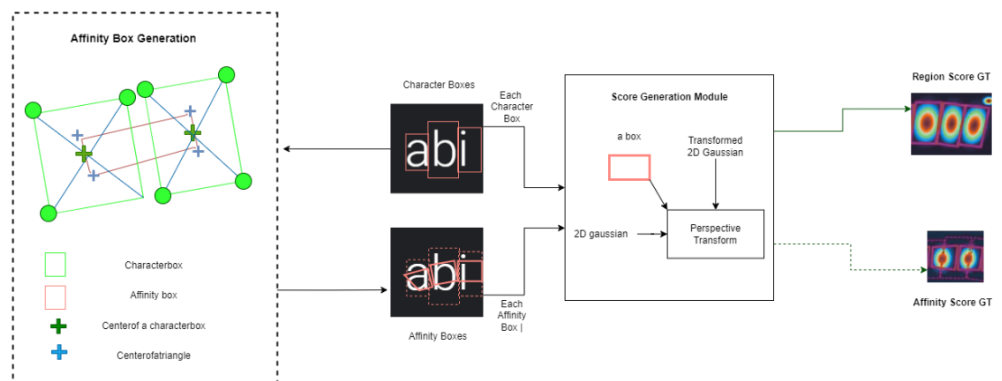


Fig: 4.5 : CRAFT Detection

4.2.3 SET UP REGIONS AND IDENTIFY TEXT (CRNN)

CRNN stands for Convolutional Recurrent Neural Network, which is a type of neural network architecture commonly used for sequence recognition tasks, particularly in the context of optical character recognition (OCR) for text detection and recognition in images.[13]

Because of its three-pronged strategy and strong structure, Easy OCR is positioned as a cutting-edge method for precise and effective text extraction from photos. The incorporation of deep learning techniques alongside the CTC algorithm demonstrates Easy OCR's dedication to propelling the Optical Character Recognition domain forward.

EasyOCR consists of three components, The three main components of EasyOCR, are feature extraction, sequence labeling, and decoding. In order to extract useful features from the input image, deep learning models such as ResNet and VGG are used in feature extraction. These characteristics are essential for text recognition in pictures. Sequence labeling, the next step, uses Long Short-Term Memory (LSTM) networks to interpret the extracted features' sequential context. Text pattern recognition and structuring are crucial tasks for LSTM networks. Finally, the decoding part decodes and transcribes the labeled sequences into the actual recognized text using the Connectionist Temporal Classification (CTC) algorithm.[14]

1. The Convolution part of the network (Feature Extraction)

In CRNN model, the component of convolutional layers is constructed by taking the convolutional and max-pooling layers from a standard CNN model such as ResNet/VGG.

Such component is used to extract a sequential feature representation from an input image. In simple words, that mean process the image in a way that preserves the spatial relationships between the pixels, while also reducing the dimensionality of the image.

These feature representations are then fed as input to the recurrent neural network (RNN) component.

The recognition model's first step is feature extraction. In order to create a set of features that can be utilized for additional analysis, the input data must be converted. With EasyOCR, VGG and Resnet are used for this.

Resnet, also known as Residual Networks, is a kind of convolutional neural network (CNN) that bypasses certain layers by using shortcuts or skip connections. This makes it possible for the network to be deeper and still be trainable by resolving the issue of vanishing gradients. Learning the residual representation functions rather than the signal representation directly is the fundamental tenet of the Resnet architecture. This lowers the model's complexity and facilitates the network's learning process.

An additional kind of CNN is called Visual Geometry Group, or VGG. Its uniform architecture and simplicity are well-known. Its homogeneous architecture consists of very small (3x3) convolution filters stacked deeper and deeper on top of one another.

This simplifies the network's understanding and modification by lowering the number of computation and parameters.

Easy OCR is an advanced optical character recognition system (OCR) module that improves recognition performance by skillfully extracting features. Taking advantage of Resnet's and VGG's deep learning expertise, Easy OCR starts with Resnet's novel method of using residual networks and skip connections to reduce vanishing gradients and allow for deep network depth. Concurrently, VGG offers its consistent architecture, which makes use of small 3x3 convolution filters to create a straightforward but effective design. This combination of VGG and Resnet strengthens the module extraction of Easy OCR which reveals a dynamic synergy that enables the model to decipher complex visual patterns and improve recognition task accuracy.

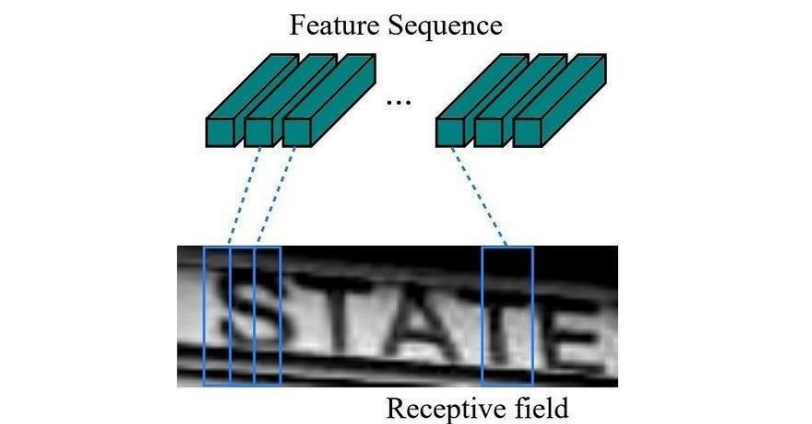


Fig: 4.6 : Feature Extraction

2. Sequence Labelling (LSTM)

The RNN component of EasyOCR is responsible for processing the sequential feature representation of the input image generated by the CNN component [last post].

It consists of multiple layers of Long Short-Term Memory (LSTM) cells that are designed to capture long-term dependencies in sequential data by selectively remembering and forgetting information.

For example, consider a case where the CNN feature extraction layer produces an encoding for “tha_k you” but the character at place “_” was not well recognized. Maybe there was a dirt spot at that place. But since the LSTM layers learned through data that there’s a vast amount of “thank you” inputs in the text, the OCR can predict the correct missing letter.

The LSTM cells in the RNN component of EasyOCR are stacked together, with each layer feeding its output to the next layer as input. The final LSTM layer outputs a sequence of probabilities for each character in the input text.

This output sequence is then passed through the (CTC) component to generate the final recognized text.

Sequence labeling comes next after feature extraction. Long Short-Term Memory (LSTM) networks are used for this.

Recurrent neural networks (RNNs) of the long-sequence learning and memory type, or LSTMs, are able to model the temporal dependencies of multiple time steps.

The distinct design of LSTM, in contrast to conventional RNNs, aids in preventing the vanishing gradient issue. The memory cell and the three different kinds of gates — input, forget, and output — allow it to accomplish this. These elements give LSTM great efficacy for sequence labeling tasks by enabling it to add or remove information from the cell state over extended sequences.

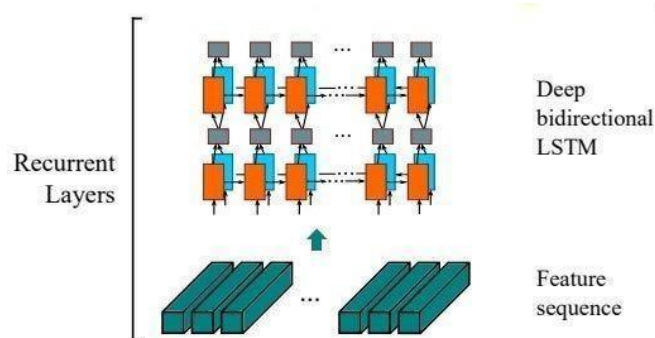


Fig: 4.7 : Sequence Labelling

3. *Decoding (CTC)*

Decoding is the last step in the recognition model, and Connectionist Temporal Classification (CTC) is used to do this.

A kind of loss function called CTC is applied to sequence issues in which time is uncertain. It is applied in situations where we are unsure of the alignment between the labels and the input data, which frequently occurs in speech and handwriting recognition.

In order to provide a prediction with a variable length, CTC appends a blank label in addition to existing labels. It is then perfect for OCR decoding tasks because it computes loss by adding up all possible alignments of the input to the target sequences.

An altered version of the deep-text-recognition-benchmark architecture serves as the training pipeline for EasyOCR's recognition execution. With the use of this framework, text recognition models can be trained on a variety of datasets, making EasyOCR incredibly versatile and effective.

CTC helps decode per-frame predictions made by RNN and transform it into a written text label sequence. The Sequence labeling problem consists of input sequences $X=[x_1, x_2, \dots, x_T]$ and its corresponding output sequences $Y=[y_1, y_2, \dots, y_U]$.

In other words, this problem we're dealing with involves accurately matching X to Y , but there are a couple of challenges to doing this. For one, both X and Y can be different lengths from each other. Additionally, we don't have a clear and precise alignment of the elements in X and Y .

The CTC algorithm overcomes these challenges. For a given X it gives us an output distribution over all possible Y 's. We can use this distribution either to infer a likely output or to assess the probability of a given output.

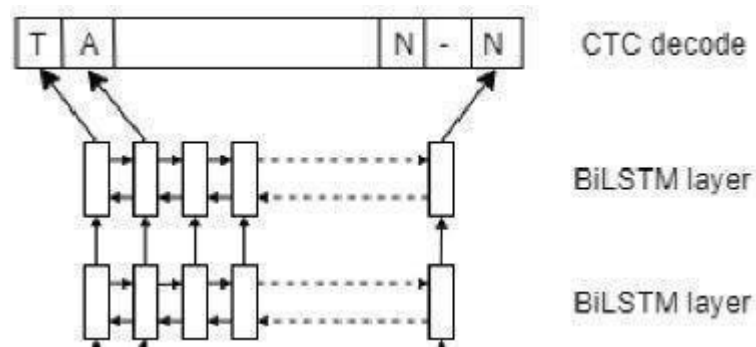


Fig: 4.8 : CTC Decoding

4.2.4 DECODE THE OUTPUT OF THE RECOGNITION MODEL

A decoding method becomes essential to the decoding process once each region of the input image generates a probability distribution over characters, particularly in the context of the Connectionist Temporal Classification (CTC)-based decoder.

The goal of this technique is to use the output probabilities to estimate the most likely character sequence. Interpreting the model's predictions and turning them into a logical and precise representation of the recognized text depends heavily on the decoding process. In order to improve the general effectiveness and dependability of the Optical Character Recognition (OCR) system in Easy OCR and guarantee precise and successful outcomes, this strategic decoding technique is crucial.

The Beam Search Decoder and the Greedy Decoder are two decoding techniques used in the recognition process. At each sequence step, the Greedy Decoder determines which character is the most likely based on a probability distribution over all potential characters generated by the recognition model.

With this method, the character with the highest probability is chosen at each stage and added to the final recognized text.

Conversely, the Beam Search Decoder improves accuracy by taking into account several options at every stage of the sequence. It reduces errors caused by ambiguity between similar characters or symbols, unlike the Greedy Decoder, by avoiding the restriction of choosing only the most likely character. Easy OCR's text recognition accuracy is enhanced by the Beam Search Decoder's capacity to investigate many choices at every stage.

4.2.5 POSTPROCESS THE RECOGNIZED TEST

In order to increase the precision and calibre of the recognized text output, post-processing in OCR entails using extra methods or algorithms. Resolving mistakes or discrepancies in the OCR findings is its main goal.

Post-processing is a component of Easy OCR's pipeline that aims to improve readability and accuracy of the recognized text. Spell checking is a common post-processing approach where misspelt words are fixed by utilizing statistical language models or consulting a dictionary.

Here are several common post-processing steps:

Spell Checking: Utilize a spell-checking algorithm or dictionary to identify and correct misspelled words within the recognized text. Correcting misspelled words by comparing them against a dictionary or using statistical language models.

Language modeling: Employing language-specific rules or statistical models to improve the coherence and grammar of the recognized text. Employ language models to evaluate the coherence and grammatical correctness of the recognized text. Incorporate statistical language models or neural language models to predict and correct unlikely word sequences.

Text normalization: Standardizing the format of the recognized text by converting abbreviations, acronyms, or special characters into their full forms or standard representations.

EasyOCR provides a post-processing step as part of its pipeline to enhance the accuracy and readability of the recognized text, but the exact methods and algorithms used may differ.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

EasyOCR is a Python-based optical character recognition (OCR) library that utilizes deep learning models to recognize text from images. Here's an overview of how EasyOCR works:

- **Image Input:** The first step in using EasyOCR is to provide it with an image containing the text you want to recognize. This image can be in various formats such as JPEG, PNG, TIFF, or PDF.
- **Preprocessing:** Before performing OCR, EasyOCR may apply preprocessing techniques to enhance the quality of the input image. These techniques may include resizing, noise reduction, contrast adjustment, or binarization to improve the clarity of the text regions.
- **Text Detection:** EasyOCR employs deep learning-based algorithms to detect text regions within the input image. These algorithms analyze the image to identify areas containing textual content, delineating boundaries around individual text elements.
- **Text Recognition:** Once text regions are detected, EasyOCR applies another deep learning model to recognize the text within each region. This recognition process involves analyzing the visual patterns and features of the text, mapping them to corresponding characters or words.
- **Post-processing:** After text recognition, EasyOCR may perform additional post-processing steps to refine the results.

This could involve correcting errors, improving formatting, or filtering out extraneous characters or noise.

- Output: Finally, EasyOCR provides the recognized text as output, typically in the form of a list or a structured data format. Users can then access this text for further processing, analysis, or integration into other applications.
- Google Text-to-Speech is a service provided by Google that allows developers to incorporate natural-sounding speech synthesis into their applications. It enables apps to convert text into spoken words using machine learning models and various language technologies.

Overall, the development and deployment setup for EasyOCR Text Extraction involves several steps, including data collection and preprocessing, model architecture design, model training, model evaluation, deployment and postprocessing. With proper planning and execution, you can develop a robust and accurate EasyOCR Text Extraction system that can be deployed in a real-world environment.

5.2 ALGORITHM AND MODEL USED

CRAFT for Character Region Awareness For Text Detection and Convolutional Recurrent Neural Network(CRNN) are two types of neural network architectures used in deep learning for various applications, including image recognition, natural language processing, and time series analysis.

5.2.1 CRAFT

A powerful and effective model for precise text detection in a variety of fonts, sizes, and orientations is Text detection using Character-Region Awareness (CRAFT).

Two important output maps are produced by CRAFT using Convolutional Neural Networks (CNNs): Score maps for character regions and affinity scores.

Scene text detection methods based on neural networks have emerged recently and have shown promising results. Previous methods trained with rigid word-level bounding boxes exhibit limitations in representing the text region in an arbitrary shape.

In the context of text detection, "Character Region Awareness" (CRA) refers to a technique or approach that aims to improve the accuracy of text detection algorithms by being aware of the regions containing characters or text in an image. This technique is particularly relevant in optical character recognition (OCR) systems, where accurately detecting and localizing text regions is a critical step before performing text recognition.

CRAFT for Character Region Awareness For Text Detection, is designed with a convolutional neural network producing the character region score and affinity score. The region score is used to localize individual characters in the image, and the affinity score is used to group each character into a single instance.

The basis of CRAFT is the Character Region Score Map, a binary representation that was meticulously constructed to show the chance that each pixel in the image belongs to a character region.

5.2.2 CRNN

CRNN stands for Convolutional Recurrent Neural Network, which is a type of neural network architecture commonly used for sequence recognition tasks, particularly in the context of optical character recognition (OCR) for text detection and recognition in images.

Because of its three-pronged strategy and strong structure, Easy OCR is positioned as a cutting-edge method for precise and effective text extraction from photos. The incorporation of deep learning techniques alongside the CTC algorithm demonstrates Easy OCR's dedication to propelling the Optical Character Recognition domain forward.

EasyOCR consists of three components, The three main components of EasyOCR, are feature extraction, sequence labeling, and decoding. In order to extract useful features from the input image, deep learning models such as ResNet and VGG are used in feature extraction. These characteristics are essential for text recognition in pictures. Sequence labeling, the next step, uses Long Short-Term Memory (LSTM) networks to interpret the extracted features' sequential context. Text pattern recognition and structuring are crucial tasks for LSTM networks. Finally, the decoding part decodes and transcribes the labeled sequences into the actual recognized text using the Connectionist Temporal Classification (CTC) algorithm.

5.3 TESTING

Confidence score testing involves evaluating the confidence scores produced by an OCR system, such as EasyOCR, to assess the reliability and accuracy of the text recognition results. Here is how you can conduct confidence score testing:

- **Dataset Selection:** Choose a dataset with ground truth labels or annotations, where the correct text for each image is known. This dataset should cover a diverse range of text types, fonts, sizes, orientations, and backgrounds.
- **OCR Processing:** Process the dataset using EasyOCR to extract text from the images. Ensure that the OCR system provides confidence scores for each recognized text region, indicating the system's confidence in the accuracy of its predictions.

- **Evaluation Metrics:** Define evaluation metrics to assess the performance of the OCR system based on confidence scores. Common metrics include precision, recall, F1-score, accuracy, and mean confidence score.
- **Threshold Selection:** Determine a threshold value for confidence scores to classify text regions as either “correctly recognized” or “incorrectly recognized.” This threshold can be set based on empirical observations, domain knowledge, or specific application requirements.
- **Performance Evaluation:** Evaluate the OCR system’s performance using the chosen evaluation metrics. Compare the system’s output against the ground truth labels to calculate precision, recall, and other performance measures. Analyse the distribution of confidence scores for correctly recognized and incorrectly recognized text regions. Determine whether there is a correlation between confidence scores and recognition accuracy.
- **Visualization and Interpretation:** Visualize the results of confidence score testing using plots, charts, or histograms to understand the distribution of confidence scores and their relationship to recognition accuracy. Interpret the findings to identify patterns, trends, or anomalies in the OCR system’s performance based on confidence scores.

By conducting confidence score testing, you can gain insights into the reliability and accuracy of text recognition results produced by the OCR system, identify areas for improvement, and make informed decisions to enhance the system’s performance

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 RESULTS

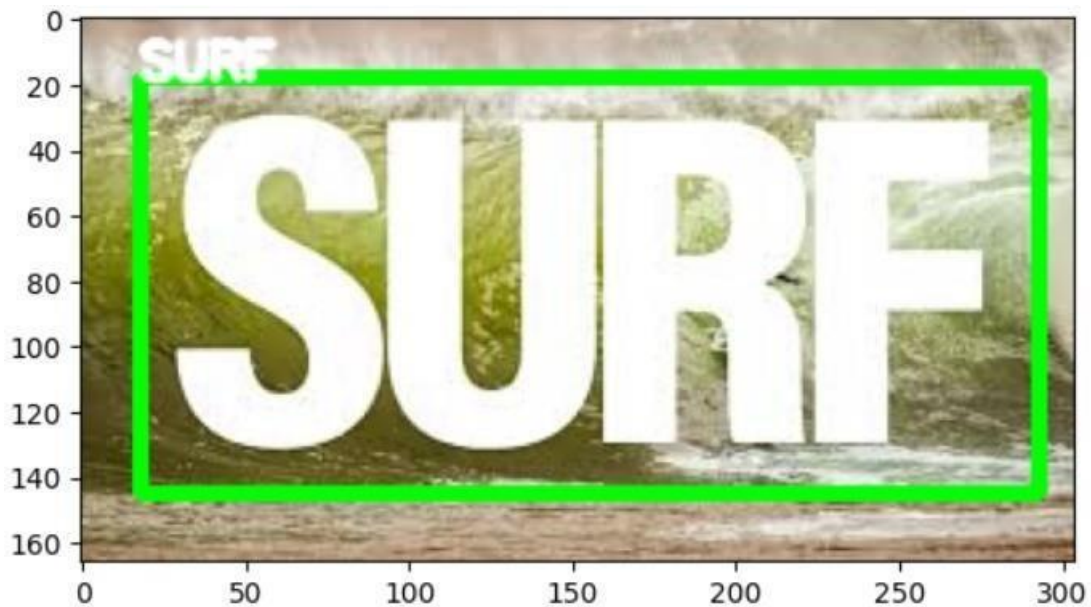
Our suggested approach, CRAFT (Character Region Awareness For Text identification), stands out in tackling the problem of scale variance by showcasing its resilience in single-scale tests conducted on a variety of datasets. In contrast to numerous other methods that utilize multi-scale testing to address scale variance, CRAFT's advantage is its capacity to localize individual characters as opposed to the full text. Our approach makes use of a comparatively narrow receptive field, efficiently covering a single character within a huge image by using the special attribute of focusing on characters. This unique method guarantees the strong identification of texts with scale variations, demonstrating the effectiveness of CRAFT in addressing scale difficulties in text recognition tasks.

In tackling the multilingual problem, our model has issues with Arabic and Bangla characters included in the IC17 dataset. These languages present difficulties for segmenting individual characters because of their cursive writing style. The absence of representation for Arabic and Bangla in the synthetic text dataset makes it more difficult for the model to discern and segment these characters correctly in comparison to Latin, Korean, Chinese, and Japanese characters. The complexities of cursive scripts impact the model's performance, in contrast to the simplicity of distinguishing East Asian characters with constant width.

We conducted single-scale trials on all datasets, despite the wide range of text sizes. This approach differs from other strategies that use multi-scale tests to address scale variance. Our strategy localizes characters rather than the entire text, providing this advantage. CRAFT's narrow receptive field can encompass a single character in a big image, making it effective for spotting scale variant texts.

We conducted the accuracy measurement through confidence score metric and have achieved more than 0.9 score for our model.

6.2 PROJECT OUTPUT



SURF

```
[([[18, 18], [293, 18], [293, 145], [18, 145]], 'SURF', 0.9862082600593567)]
```

Fig: 6.1 : Output 1 Screenshot



DON ' T
G IVE U P
G R E A T
THING S
TA K E
TIME

```
[([[13, 95], [273, 95], [273, 157], [13, 157]], "DON ' T", 0.6044751187189263),
 ([[13, 161], [367, 161], [367, 225], [13, 225]],
  'G IVE U P',
  0.5355565539287588),
 ([[13, 229], [285, 229], [285, 291], [13, 291]],
  'G R E A T',
  0.41971708014240455),
 ([[9, 295], [321, 295], [321, 361], [9, 361]], 'THING S', 0.6375045751343494),
 ([[9, 363], [221, 363], [221, 425], [9, 425]], 'TA K E', 0.7306729199528245),
 ([[264, 362], [463, 362], [463, 428], [264, 428]],
  'TIME',
  0.9749440550804138)]
```

Fig: 6.2 : Output 2 Screenshot



```
[([19, 181], [165, 181], [165, 201], [19, 201]),
 'HEAD PROTECTION',
 0.9778256601171587),
 ([[31, 201], [153, 201], [153, 219], [31, 219]],
 'MUST BE WORN',
 0.9683688296531474),
 ([[39, 219], [145, 219], [145, 237], [39, 237]],
 'ON THIS SITE',
 0.9683973806374694)]
```

HEAD PROTECTION
MUST BE WORN
ON THIS SITE

Fig: 6.3 : Output 3 Screenshot



Reduce your risk of coronavirus infection:
Clean hands with soap and water or alcohol-based hand rub
Cover nose and mouth when coughing and sneezing with tissue or flexed elbow
Avoid close contact with anyone with cold or flu-like symptoms
Thoroughly cook meat and eggs
No unprotected contact with live wild or farm animals World Health Organization

Fig: 6.4 : Output 4 Screenshot

6.3 PERFORMANCE ANALYSIS

Confidence Metrics: A confidence score in OCR represents the level of certainty that the system assigns to each recognized character or word. Higher scores indicate greater confidence in accuracy, aiding quality assessment and error correction.

We conducted the accuracy measurement through confidence score metric and have achieved more than 0.9 score for our model which is around more than 99 percentage.

Generally, confidence scores range from 0 to 1, where:

A score of 1 indicates the highest level of confidence, meaning the OCR system is highly certain that the recognition is accurate.

A score of 0 indicates the lowest level of confidence, suggesting that the OCR system is not confident in the accuracy of the recognition.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

We offer in this study a novel text detector, named CRAFT, which is able to recognize ind. characters even in the lack of annotations at the character level. The method covers a wide range of text shapes in a bottom-up way and generates character affinity scores and character region ratings using a two-score methodology. In light of the lack of genuine datasets with character-level annotations, we propose an ill-guided learning strategy. This technique generates false-ground truths from an interim model, hence resolving the issue of limited annotated datasets. Notably, CRAFT exhibits its tenacity and generalization skills by achieving cutting-edge performances across many public datasets without the requirement for fine-tuning.

An innovative neural network architecture is the neural network that is Convolutional Recurrent (CRNN) that combines best features of Recurrent and Convolutional neural networks (CNN and RNN), is also introduced in this research. CRNN can process images with different dimensions as input and produce predictions with different lengths. The most remarkable thing about it is that it works directly on coarse level labels like words, so it does not require specific annotations for particular pieces like letters during training.

Moreover, CRNN departs from classical neural networks by forgoing completely linked layers, which produces a more condensed and effective model. All these characteristics put CRNN in a highly advantageous position for image-based sequence recognition.

Easy OCR is a feature-rich optical character recognition (OCR) tool that is designed to satisfy a wide range of requirements.

Its advanced features include multilingual support, which enables the recognition and extraction of text from images in numerous languages, the ability to handle noisy images, read license plates from grayscale photos, batch processing, document digitization, and much more.

7.2 FUTURE WORK

Considering the current advancements in character recognition and OCR technologies, as well as the identified open problems in existing systems, several avenues for future work in your project could be explored:

Enhanced Handwriting Variability Handling: Develop techniques to improve the robustness of character recognition algorithms in handling variability in handwriting styles, strokes, and individual characteristics. This could involve leveraging deep learning models trained on diverse handwriting datasets and incorporating techniques for data augmentation and domain adaptation.

Multilingual and Cursive Handwriting Recognition: Extend the capabilities of the OCR system to support a wider range of languages, dialects, and writing styles, including cursive handwriting recognition. This could involve collecting and annotating datasets for languages with limited OCR support and developing specialized models and algorithms for cursive handwriting recognition.

Integration with Handwriting Input Devices: Explore integration with handwriting input devices such as digital pens or stylus-based tablets to enable real-time capture and recognition of handwritten text. This could involve developing specialized OCR models and algorithms optimized for input from handwriting devices and designing user interfaces for seamless integration into existing workflows.

REFERENCES

1. "Character-Aware Sampling and Rectification for Scene Text Recognition," by M. Li, B. Fu, Z. Zhang, and Y. Qiao, IEEE Transactions on Multimedia, vol. 25, pp. 649-661, 2023, doi: 10.1109/TMM.2021.3129651
2. Coquenet, C. Chatelain, and T. Paquet, "End-to-End Handwritten Paragraph Text Recognition Using a Vertical Attention Network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 508–524, January 1, 2023, doi: 10.1109/TPAMI.2022.3144899.
3. In IEEE Access, vol. 11, pp. 39621-39634, 2023, R. D. R et al., "SelfAdaptive Hybridized Lion Optimization Algorithm With Transfer Learning for Ancient Tamil Character Recognition in Stone Inscriptions," doi: 10.1109/ACCESS.2023.3268545.
4. In IEEE Access, vol. 11, pp. 58406–58421, 2023, doi: 10.1109/ACCESS.2023.3283340, Q. -D. Nguyen, N. -M. Phan, P. Krömer, and D. -A. Le, "An Efficient Unsupervised Approach for OCR Error Correction of Vietnamese OCR Text,"
5. "A Two-Level Rectification Attention Network for Scene Text Recognition," by L. Wu, Y. Xu, J. Hou, C. L. P. Chen, and C. -L. Liu, IEEE Transactions on Multimedia, vol. 25, pp. 2404-2414, 2023, doi: 10.1109/TMM.2022.3146779.
6. D. R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa and D. Arun, "Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 966-971, doi: 10.1109/ICECA55336.2022.10009215.
7. P. Kuneekar, K. Vayadande, O. Kulkarni, K. Ingale, R. Kadam and S. Inamdar, "OCR based Cheque Validation using Image Processing," 2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2023, pp. 1-5, doi: 10.1109/ICNTE56631.2023.10146687.

8. D. C. Dalwadi and S. Amin, "ANPR Based Toll Tax Collection System," 2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC), Rajkot, India, 2023, pp. 53-57, doi: 10.1109/R10-HTC57504.2023.10461835.
9. S. N, K. S, V. C, D. N and S. G, "Handwritten trip sheet to digital conversion using OCR," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 7-10, doi: 10.1109/ICAECIS58353.2023.10170030.
10. B. K. Pattanayak, A. K. Biswal, S. R. Laha, S. Pattnaik, B. B. Dash and S. S. Patra, "A Novel Technique for Handwritten Text Recognition Using Easy OCR," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp. 1115-1119, doi: 10.1109/ICSSAS57918.2023.10331704.
11. A Hybrid Approach for Mathematical Expression Detection in Scientific Document Images, B. H. Phong, T. M. Hoang, and T. -L. Le, IEEE Access, vol. 8, pp. 83663-83684, 2020, doi: 10.1109/ACCESS.2020.2992067.
12. "Character Region Awareness for Text Detection," Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, Hwalsuk Lee, Computer Vision and Pattern Recognition, arXiv:1904.01941, doi.org/10.48550/arXiv.1904.01941
13. "Cursive Text Recognition in Natural Scene Images Using Deep Convolutional Recurrent Neural Network," A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, IEEE Access, vol. 10, pp. 10062-10078, 2022, doi: 10.1109/ACCESS.2022.3144844
14. "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition," Baoguang Shi, Xiang Bai, Cong Yao, Computer Vision and Pattern Recognition, arXiv:1507.05717, doi.org/10.48550/arXiv.1507.05717

APPENDIX

(A) SOURCE CODE

(1) Detection.py

```
import torch
import torch.backends.cudnn as cudnn
from torch.autograd import Variable
from PIL import Image
from collections import OrderedDict

import cv2
import numpy as np
from .craft_utils import getDetBoxes, adjustResultCoordinates
from .imgproc import resize_aspect_ratio, normalizeMeanVariance
from .craft import CRAFT

def copyStateDict(state_dict):
    if list(state_dict.keys())[0].startswith("module"):
        start_idx = 1
    else:
        start_idx = 0
    new_state_dict = OrderedDict()
    for k, v in state_dict.items():
        name = ".".join(k.split(".")[start_idx:])
        new_state_dict[name] = v
    return new_state_dict

def test_net(canvas_size, mag_ratio, net, image, text_threshold,
link_threshold, low_text, poly, device, estimate_num_chars=False):
    if isinstance(image, np.ndarray) and len(image.shape) == 4: # image is
batch of np arrays
```

```

        image_arrs = image
    else:
        # image is single numpy array
        image_arrs = [image]

    img_resized_list = []
    # resize
    for img in image_arrs:
        img_resized, target_ratio, size_heatmap = resize_aspect_ratio(img,
canvas_size,

interpolation=cv2.INTER_LINEAR,

                                mag_ratio=mag_ratio)

        img_resized_list.append(img_resized)
    ratio_h = ratio_w = 1 / target_ratio
    # preprocessing
    x = [np.transpose(normalizeMeanVariance(n_img), (2, 0, 1))
        for n_img in img_resized_list]
    x = torch.from_numpy(np.array(x))
    x = x.to(device)

    # forward pass
    with torch.no_grad():
        y, feature = net(x)

    boxes_list, polys_list = [], []
    for out in y:
        # make score and link map
        score_text = out[:, :, 0].cpu().data.numpy()
        score_link = out[:, :, 1].cpu().data.numpy()

```

```

# Post-processing
boxes, polys, mapper = getDetBoxes(
    score_text, score_link, text_threshold, link_threshold, low_text, poly,
    estimate_num_chars)

# coordinate adjustment
boxes = adjustResultCoordinates(boxes, ratio_w, ratio_h)
polys = adjustResultCoordinates(polys, ratio_w, ratio_h)
if estimate_num_chars:
    boxes = list(boxes)
    polys = list(polys)
    for k in range(len(polys)):
        if estimate_num_chars:
            boxes[k] = (boxes[k], mapper[k])
        if polys[k] is None:
            polys[k] = boxes[k]
    boxes_list.append(boxes)
    polys_list.append(polys)

return boxes_list, polys_list

def get_detector(trained_model, device='cpu', quantize=True,
    cudnn_benchmark=False):
    net = CRAFT()

    if device == 'cpu':
        net.load_state_dict(copyStateDict(torch.load(trained_model,
            map_location=device)))
        if quantize:
            try:
                torch.quantization.quantize_dynamic(net, dtype=torch.qint8,
                    inplace=True)

```

```

except:
    pass
else:
    net.load_state_dict(copyStateDict(torch.load(trained_model,
map_location=device)))
    net = torch.nn.DataParallel(net).to(device)
    cudnn.benchmark = cudnn_benchmark

net.eval()
return net

def get_textbox(detector, image, canvas_size, mag_ratio, text_threshold,
link_threshold, low_text, poly, device, optimal_num_chars=None, **kwargs):
    result = []
    estimate_num_chars = optimal_num_chars is not None
    bboxes_list, polys_list = test_net(canvas_size, mag_ratio, detector,
                                       image, text_threshold,
                                       link_threshold, low_text, poly,
                                       device, estimate_num_chars)
    if estimate_num_chars:
        polys_list = [[p for p, _ in sorted(polys, key=lambda x:
abs(optimal_num_chars - x[1]))]
                      for polys in polys_list]

    for polys in polys_list:
        single_img_result = []
        for i, box in enumerate(polys):
            poly = np.array(box).astype(np.int32).reshape((-1))
            single_img_result.append(poly)
        result.append(single_img_result)
    return result

```

(2) Craft.py

```
import torch
import torch.nn as nn
import torch.nn.functional as F

from .model.modules import vgg16_bn, init_weights

class double_conv(nn.Module):
    def __init__(self, in_ch, mid_ch, out_ch):
        super(double_conv, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(in_ch + mid_ch, mid_ch, kernel_size=1),
            nn.BatchNorm2d(mid_ch),
            nn.ReLU(inplace=True),
            nn.Conv2d(mid_ch, out_ch, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_ch),
            nn.ReLU(inplace=True)
        )

    def forward(self, x):
        x = self.conv(x)
        return x

class CRAFT(nn.Module):
    def __init__(self, pretrained=False, freeze=False):
        super(CRAFT, self).__init__()

        """ Base network """
        self.basenet = vgg16_bn(pretrained, freeze)

        """ U network """
        self.upconv1 = double_conv(1024, 512, 256)
        self.upconv2 = double_conv(512, 256, 128)
        self.upconv3 = double_conv(256, 128, 64)
        self.upconv4 = double_conv(128, 64, 32)

        num_class = 2
        self.conv_cls = nn.Sequential(
            nn.Conv2d(32, 32, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(32, 32, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(32, 16, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
```

```

nn.Conv2d(16, 16, kernel_size=1), nn.ReLU(inplace=True),
    nn.Conv2d(16, num_class, kernel_size=1),
)

init_weights(self.upconv1.modules())
init_weights(self.upconv2.modules())
init_weights(self.upconv3.modules())
init_weights(self.upconv4.modules())
init_weights(self.conv_cls.modules())

def forward(self, x):
    """ Base network """
    sources = self.basenet(x)

    """ U network """
    y = torch.cat([sources[0], sources[1]], dim=1)
    y = self.upconv1(y)

    y = F.interpolate(y, size=sources[2].size()[2:], mode='bilinear',
align_corners=False)
    y = torch.cat([y, sources[2]], dim=1)
    y = self.upconv2(y)

    y = F.interpolate(y, size=sources[3].size()[2:], mode='bilinear',
align_corners=False)
    y = torch.cat([y, sources[3]], dim=1)
    y = self.upconv3(y)

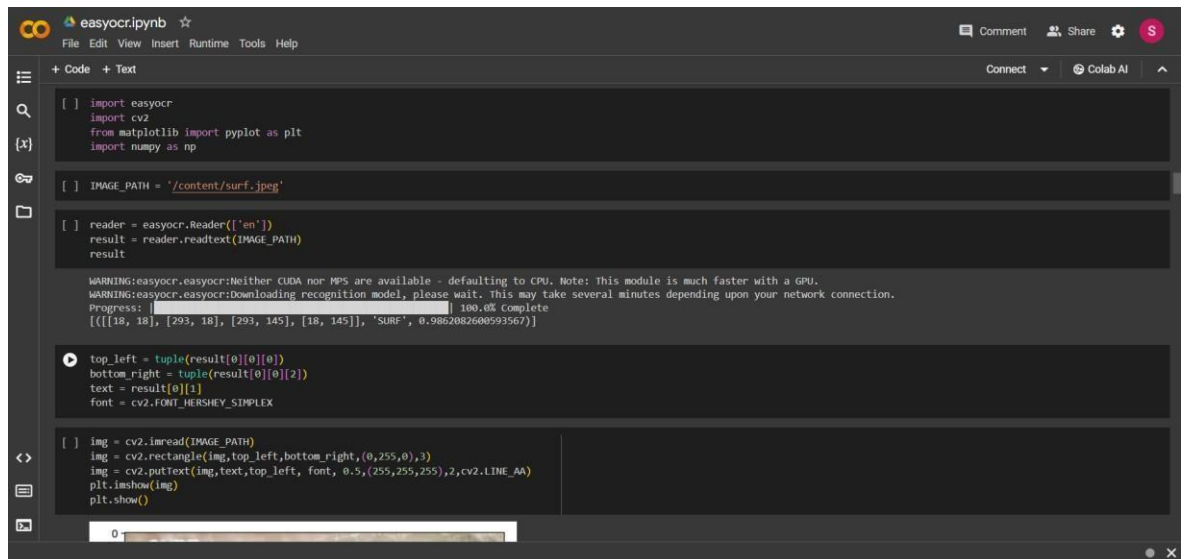
    y = F.interpolate(y, size=sources[4].size()[2:], mode='bilinear',
align_corners=False)
    y = torch.cat([y, sources[4]], dim=1)
    feature = self.upconv4(y)

    y = self.conv_cls(feature)

    return y.permute(0,2,3,1), feature

```

(B) SCREENSHOTS



```
import easyocr
import cv2
from matplotlib import pyplot as plt
import numpy as np

IMAGE_PATH = '/content/surf.jpeg'

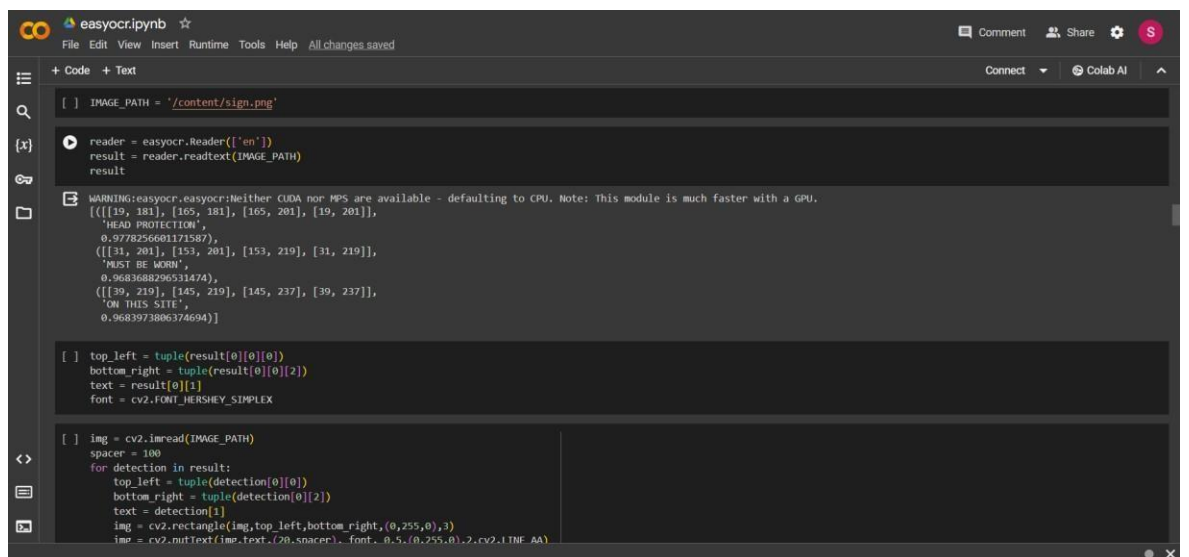
reader = easyocr.Reader(['en'])
result = reader.readtext(IMAGE_PATH)
result

WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr:Downloading recognition model, please wait. This may take several minutes depending upon your network connection.
Progress: [ ] 100.0% Complete
[[[18, 18], [293, 18], [293, 145], [18, 145]], 'SURF', 0.986208260593567]]

top_left = tuple(result[0][0][0])
bottom_right = tuple(result[0][0][2])
text = result[0][1]
font = cv2.FONT_HERSHEY_SIMPLEX

img = cv2.imread(IMAGE_PATH)
img = cv2.rectangle(img,top_left,bottom_right,(0,255,0),3)
img = cv2.putText(img,text,top_left, font, 0.5,(255,255,255),2,cv2.LINE_AA)
plt.imshow(img)
plt.show()
```

Fig: B.1 : Importing Libraries



```
IMAGE_PATH = '/content/sign.png'

reader = easyocr.Reader(['en'])
result = reader.readtext(IMAGE_PATH)
result

WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
[[[19, 181], [165, 181], [165, 201], [19, 201]],
 'HEAD PROTECTION',
 0.9778256681171587],
 [[31, 201], [153, 201], [153, 219], [31, 219]],
 'MUST BE WORN',
 0.9683688296531474],
 [[39, 219], [145, 219], [145, 237], [39, 237]],
 'ON THIS SITE',
 0.9683973806374694]]

top_left = tuple(result[0][0][0])
bottom_right = tuple(result[0][0][2])
text = result[0][1]
font = cv2.FONT_HERSHEY_SIMPLEX

img = cv2.imread(IMAGE_PATH)
spacer = 100
for detection in result:
    top_left = tuple(detection[0][0])
    bottom_right = tuple(detection[0][2])
    text = detection[1]
    img = cv2.rectangle(img,top_left,bottom_right,(0,255,0),3)
    img = cv2.putText(img,text,(20,spacer),font,0.5,(0,255,0),2,cv2.LINE_AA)
```

Fig: B.2 : Text Detection

```

[ ] IMAGE_PATH = '/content/quote.jpg'

reader = easyocr.Reader(['en'])
result = reader.readtext(IMAGE_PATH)
result

WARNING:easyocr:easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
[[[13, 95], [273, 95], [273, 157], [13, 157]],
  'OM', '1',
  0.41628427489431497),
  ([[13, 161], [367, 161], [367, 225], [13, 225]],
  'GIVE UP',
  0.3732729625602953),
  ([[13, 229], [285, 229], [285, 291], [13, 291]],
  'GREAT',
  0.42364618692069556),
  ([[9, 295], [321, 295], [321, 361], [9, 361]], 'THING S', 0.6095978991051743),
  ([[9, 363], [221, 363], [221, 425], [9, 425]], 'TAKE', 0.7667791346095685),
  ([[264, 362], [463, 362], [463, 428], [264, 428]],
  'TIME',
  0.9752727746963501]]

[ ] img = cv2.imread(IMAGE_PATH)
  spacer = 100
  for detection in result:
    top_left = tuple(detection[0][0])
    bottom_right = tuple(detection[0][2])
    text = detection[1]
    img = cv2.rectangle(img, top_left, bottom_right, (0, 255, 0), 3)
    img = cv2.putText(img, text, (20, spacer), font, 0.5, (0, 255, 0), 2, cv2.LINE_AA)
    spacer+=15

```

Fig: B.3 : Text Extraction

```

[ ] import easyocr
import cv2
reader = easyocr.Reader(['en'])
result = reader.readtext('/content/sample_data/english.png', paragraph=True)
print(result)

WARNING:easyocr:easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
WARNING:easyocr:easyocr:Downloading detection model, please wait. This may take several minutes depending upon your network connection.
Progress: | 100.0% Complete
WARNING:easyocr:easyocr:Downloading recognition model, please wait. This may take several minutes depending upon your network connection.
Progress: | 100.0% Complete[[[231, 32], [674, 32], [674, 64], [231, 64]], 'Reduce your risk of coronavirus infection:'], [[[326, 98], [674, 98], [674, 130], [326, 130]], 'Clean hands with soap and water', 'or alcohol-based hand rub', 'Cover nose and mouth when coughing and', 'sneezing with tissue or flexed elbow', 'Avoid close contact with anyone with', 'cold or flu-like symptoms', 'Thoroughly cook meat and eggs', 'No unprotected contact with live wild', 'or farm animals', 'World Health', 'Organization']]

[ ] reader.readtext('/content/sample_data/english.png', detail = 0)

[ ] import cv2
import matplotlib.pyplot as plt
top_left = tuple(result[0][0][0])
bottom_right = tuple(result[0][0][2])

```

Fig: B.4 : Text Recognition

(C) BEST PAPER MERIT CERTIFICATE



(D) INTERNATIONAL CONFERENCE CERTIFICATE



(E) PUBLICATION CERTIFICATE



EasyOCR Applied in Digital Format for Text Extraction using Deep Learning Technology

Soumyadeep Das

itsoumyadeepdas@gmail.com

Undergraduate Scholar

Sathyabama Institute of Science and
Technology
Chennai, India

Kripashankar Manglam

manglamkripashankar201@gmail.com

Undergraduate Scholar

Sathyabama Institute of Science and
Technology
Chennai, India

Dr. Kamatchi K S

kamatchi.k.s.it@sathyabama.ac.in

Information Technology

Assistant Professor

Sathyabama Institute of Science and
Technology
Chennai, India

Abstract— It describes an OCR effort aimed at creating a sophisticated system for precise and effective text recognition across a variety of documents and languages is shown in this abstract. By leveraging cutting-edge machine learning methods like multilingual training, attention mechanisms, and deep neural networks, the system seeks to outperform conventional OCR systems. Convolutional and recurrent neural networks are trained, preprocessed, and extensive data collecting are all part of the methodology. The versatility of the OCR system is demonstrated by its capacity to adapt to a wide range of scripts, including Latin, Cyrillic, Chinese, and Devanagari. When compared to benchmark datasets, evaluation shows significant advances in recall, accuracy, precision, and F1-score, often reaching human-like levels. The real-world uses include data entry automation, information retrieval optimization, digitizing archives, and assistance for the blind. In conclusion, this OCR project represents a substantial development in text recognition technology and offers improved processing accuracy and efficiency for a variety of document applications.

Keywords— EASYOCR, Optical Character Recognition, Text Extraction, Deep Learning, CRNN, CRAFT.

I. INTRODUCTION

The constant digitization of information in the modern era has highlighted the crucial role that optical character recognition (OCR) systems play in smoothly transforming handwritten or printed text into formats that are readable by machines. Even though they are invaluable, traditional OCR algorithms frequently struggle with sophisticated character identification jobs, different font styles, and diverse document layouts. This journal article presents a state-of-the-art OCR study that aims to overcome these constraints and push the limits of processing speed and accuracy. The suggested OCR system seeks to redefine the discipline by utilising deep neural networks, powerful machine learning algorithms, and natural language processing approaches.

The main goal of this study is to enhance the system's text recognition accuracy and adaptability to different scripts and languages. Because convolutional and recurrent neural networks (RNNs and CNNs) are used, for feature extraction and order modelling, correspondingly, the OCR system can handle complex document layouts and a wide range of language forms. By allowing the model to concentrate on pertinent passages of the text, attention processes are included, improving the recognition accuracy overall.

The OCR system uses multilingual training and transfer learning methods to handle the multilingual nature of modern documents. This methodical technique guarantees character identification in a wide range of scripts, such as

Chinese, Latin, Cyrillic, Devanagari, and more. The project's methodology includes painstaking data gathering, preprocessing, and training on a variety of datasets that include handwritten texts, photos, and scanned documents..

Thorough testing against real-world documents, handwritten samples, and benchmark datasets is part of the evaluation process for the built OCR system. The robustness and effectiveness of the system are indicated by performance metrics like the confidence score. The outcomes demonstrate notable progress over current OCR algorithms, frequently attaining accuracy levels that are on par with human perception.

Beyond its technical accomplishments, this cutting-edge OCR technology has many practical applications. The significance of this OCR project is multifaceted, ranging from the digitization of historical archives to the facilitation of information retrieval in large-scale document repositories, and from the improvement of automated data entry systems to the enablement of assistive technology for the visually impaired.

This journal article outlines a thorough strategy for OCR technology, utilising cutting-edge techniques to improve text recognition's accuracy, versatility, and efficiency. If this initiative is successful, document processing could undergo a revolution. This would create new opportunities for the smooth integration of text data into different applications, which would greatly increase productivity in the digital age.

II. RELATED WORK

Ming Li and Bin Fu (2023) suggest an application for Character-Aware Sampling and Rectification (CASR) that makes use of precise character location and orientation data to handle irregular text occurrences. With this novel method, CASR predicts character-level properties to direct the sampling and rectification procedures by treating every character as a fundamental unit. In order to enhance the rectification of entire text lines, this module employs a novel technique that makes use of particular character information and character-level monitoring throughout the training phase. Above all, CASR is meant to be an easily integrated plug-and-play module into pre-existing text recognition pipelines. Broad tests on many standards show that the suggested approach is robust and achieves promising overall performance by regularly producing rectified text instances with greater accuracy. With its character-centric methodology, this CASR module presents a significant development in the field of text correction and provides a

workable and efficient way to increase accuracy in situations of irregular text.[1]

Denis Coquenat and Clement Chatelain (2023) tackle the problem of computer vision's unrestricted handwritten text recognition. They suggest a hybrid attention end-to-end approach that is intended to process a paragraph picture iteratively in order to identify individual lines. Three modules make up the model: a decoder for character sequence recognition, an attention module for implicit line segmentation, and an encoder for feature map generation. This novel method provides a cohesive answer and demonstrates effectiveness in identifying complete paragraphs in unrestricted handwritten material.[2]

R. Devi and S. Karthikeyan's study from 2023 addresses a crucial pattern recognition research subject, concentrating on the difficulties Tamil characters present because of their complex properties. Important historical information can be found in Tamil Nadu's stone inscriptions, but the characters' holes, loops, and curves make it difficult to read the writing. The proposed method combines a Deep Convolution Neural Network and Transfer Learning (TL) for character recognition. By successfully identifying characters from stone inscriptions, the hybrid method shows promise in addressing the technical issues linked to Tamil character identification.[3]

The work of Quoc-Dung Nguyen and Nguyet-Minh Phan provides a fresh unsupervised method to OCR mistake correction. Their method produces and investigates rectification candidates near OCR errors using a hill-climbing algorithm. Interestingly, corrective characters are taken out of OCR training data and only from original ground truth texts. The proposed model employs a weighted objective function that is heuristically examined for the optimal weight combinations in order to score and rank repair possibilities. An OCR dataset from the Vietnamese handwritten text recognition competition held online in 2018 by ICFHR is used to assess the model's efficacy. Furthermore, the suggested method's stability and complexity are confirmed. This new unsupervised OCR error correction model is a viable way to improve OCR applications' accuracy and dependability.[4]

Lintai Wu and Yong Xu propose A Two-Level Attention Network for Rectification (TRAN) for text rectification and acknowledgment. Two-Level Network for Rectification (TORN) and Attention-Based Recognition Network (ABRN) comprise TRAN. By addressing geometric restrictions and making pixel-by-pixel corrections for clarity, TORN corrects text geometry. ABRN specialises in text recognition in photos that have been rectified. To enhance feature extraction and effectively handle variations in channel interdependencies and character size, a novel both kernel- and channel-level attention units is integrated into the prototype. This novel method provides a thorough resolution to geometric restrictions and unpredictability in text recognition situations.[5]

III. PROPOSED SYSTEM

A simple and flexible optical character recognition (OCR) module for Python, Easy OCR is easy to use. Easy OCR is a versatile tool that is designed to cater to the needs of both

novice and expert developers in OCR and computer vision. It supports a multitude of languages and offers pre-trained models for text recognition and identification.

The library is a solid option for Python developers because it places an emphasis on speed and efficiency while identifying text within images. Easy OCR's ability to handle a wide range of typefaces and text layouts with precision and robustness makes it an invaluable tool for automating text-related tasks and improving data extraction from scanned documents.

With this technique, developers can focus on the unique requirements of their products while easily integrating OCR capabilities into a variety of Python projects, such as desktop software and online apps. For utilising OCR in Python, Easy OCR proves to be a useful and approachable option, particularly for computer vision applications.

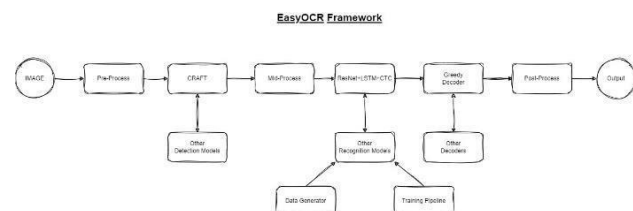


Figure 1: Easy OCR Framework

A. PREPROCESS THE IMAGE

An important first stage in the Easy OCR pipeline is image preprocessing, which improves the quality of the images in preparation for text recognition. Several techniques are used in this approach to overcome common image issues.

Noise reduction is the first and most important duty of image preparation in Easy OCR. Eliminating unnecessary noise is essential to boosting the image's overall clarity and, in turn, the accuracy of text recognition. A variety of complex filters, such as median and Gaussian blur, are deliberately utilised to accomplish this. With the effective removal of unnecessary components, the image becomes cleaner and more suitable for precise text recognition.

Easy OCR uses binarization algorithms to convert the grayscale image into a binary representation after noise reduction. Preparing the image for more efficient text recognition algorithms is the goal. Adaptive thresholding and Otsu's method are two of the methods used to accomplish this transition. The result is a binary image that helps with next steps in the OCR pipeline by emphasising text parts against a transparent background.

Skew correction is the last step in Easy OCR's picture preprocessing process. Finding and fixing the image's skew angle is essential to guaranteeing that the text elements are aligned correctly. Recognising skewed text can be difficult, and skew angle correction improves the OCR system's overall accuracy. For skew correction, Easy OCR uses complex algorithms to align the text horizontally and produce a picture that is well-aligned. This correction guarantees that the text is consistently presented and prepared for precise recognition, which is crucial for later stages of the OCR process.

B. DETECT THE REGIONS OF THE IMAGE THAT CONTAINS CRAFT

A powerful and effective model for precise text detection in a variety of fonts, sizes, and orientations is Text detection using Character-Region Awareness (CRAFT). Two important output maps are produced by CRAFT using Convolutional Neural Networks (CNNs): Score maps for character regions and affinity scores.

The basis of CRAFT is the Character Region Score Map, a binary representation that was meticulously constructed to show the chance that each pixel in the image belongs to a character region. This map offers valuable insights into the spatial distribution of textual elements and is an effective tool for localising individual characters. CRAFT provides the groundwork for accurate and effective text detection in photos by efficiently identifying character regions.

CRAFT presents the Affinity Score Map as an addition to the Character Region Score Map. When assessing the probability of adjacency between two letters in the same text instance, this binary map is essential. The map helps organise individual characters into meaningful and cohesive text parts by capturing the affinity between characters. The model's capacity to comprehend and distinguish words inside the image is improved by this grouping into a single instance, or single instance word.[6]

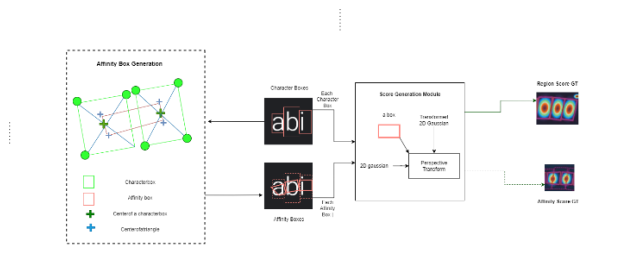


Figure 2: Score maps for character regions and affinity scores

C. SET UP REGIONS AND IDENTIFY TEXT (CRNN)

The Python-based OCR programme EasyOCR presents a robust three-component system that includes sequence labelling, decoding, and feature extraction. This novel method uses deep learning models—ResNet and VGG in particular—to extract important features from input images, which serve as the foundation for reliable word detection in a variety of visual settings.

Because of its three-pronged strategy and strong structure, Easy OCR is positioned as a cutting-edge method for precise and effective text extraction from photos. The incorporation of deep learning techniques alongside the CTC algorithm demonstrates Easy OCR's dedication to propelling the Optical Character Recognition domain forward.

Easy OCR is an advanced optical character recognition system (OCR) module that improves recognition performance by skillfully extracting features. Taking

advantage of Resnet's and VGG's deep learning expertise, Easy OCR starts with Resnet's novel method of using residual networks and skip connections to reduce vanishing gradients and allow for deep network depth. Concurrently, VGG offers its consistent architecture, which makes use of small 3x3 convolution filters to create a straightforward but effective design. This combination of VGG and Resnet strengthens the module extraction of Easy OCR which reveals a dynamic synergy that enables the model to decipher complex visual patterns and improve recognition task accuracy.

Easy OCR smoothly moves from feature extraction to sequence labelling, where Long Short-Term Memory (LSTM) networks are the main players. Specialized recurrent neural networks, or LSTMs, are excellent at simulating temporal dependencies across long sequences. Their unique architecture solves the vanishing gradient issue that plagues conventional RNNs. It includes memory cells and three essential gates (input, forget, and output). This enables LSTMs in Easy OCR to perform sequence labelling jobs effectively, enabling more accurate and efficient nuanced information processing over longer sequences.

Easy OCR uses Connectionist Temporal Classification (CTC) for decoding at the end of the recognition model. For sequences with ambiguous temporal alignments, such as those seen in speech and handwriting recognition tasks, CTC functions as a specialized loss function. By inserting a blank label alongside preexisting labels, CTC allows for variable-length predictions, which is especially helpful in situations when there is uncertainty about the alignment between labels and input data. Because of this feature, CTC is perfect for OCR decoding because it calculates loss by adding up all potential alignments between the input and target sequences. The training pipeline, which is based on the deep-text-recognition-benchmark architecture, increases the adaptability and efficiency of Easy OCR and permits reliable training on a variety of datasets.[7]

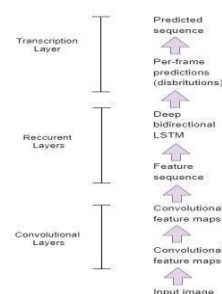


Figure 3: Network Architecture

D. DECODE THE OUTPUT OF THE RECOGNITION MODEL

A decoding method becomes essential to the decoding process once each region of the input image generates a probability distribution over characters, particularly in the context of the Connectionist Temporal Classification (CTC)-based decoder.

The goal of this technique is to use the output probabilities to estimate the most likely character sequence. Interpreting the model's predictions and turning them into a logical and precise representation of the recognized text depends heavily on the decoding process. In order to improve the general effectiveness and dependability of the Optical Character Recognition (OCR) system in Easy OCR and guarantee precise and successful outcomes, this strategic decoding technique is crucial.

The Beam Search Decoder and the Greedy Decoder are two decoding techniques used in the recognition process. At each sequence step, the Greedy Decoder determines which character is the most likely based on a probability distribution over all potential characters generated by the recognition model. With this method, the character with the highest probability is chosen at each stage and added to the final recognized text.

Conversely, the Beam Search Decoder improves accuracy by taking into account several options at every stage of the sequence. It reduces errors caused by ambiguity between similar characters or symbols, unlike the Greedy Decoder, by avoiding the restriction of choosing only the most likely character. Easy OCR's text recognition accuracy is enhanced by the Beam Search Decoder's capacity to investigate many choices at every stage.

E. POSTPROCESS THE RECOGNIZED TEST

In order to increase the precision and calibre of the recognized text output, post-processing in OCR entails using extra methods or algorithms. Resolving mistakes or discrepancies in the OCR findings is its main goal.

Post-processing is a component of Easy OCR's pipeline that aims to improve readability and accuracy of the recognized text. Spell checking is a common post-processing approach where misspelt words are fixed by utilizing statistical language models or consulting a dictionary.

Language-specific rules or statistical models are used in language modelling to enhance coherence and grammar. Furthermore, by transforming acronyms, special characters, and abbreviations into their whole names or standard representations, text normalization guarantees that the accepted text format is standardized. Although Easy OCR incorporates post-processing for text enhancement, several techniques and algorithms may be used to achieve the best outcomes.

IV. RESULTS AND DISCUSSIONS

Our suggested approach, CRAFT (Character Region Awareness For Text identification), stands out in tackling the problem of scale variance by showcasing its resilience in single-scale tests conducted on a variety of datasets. In contrast to numerous other methods that utilise multi-scale testing to address scale variance, CRAFT's advantage is its capacity to localise individual characters as opposed to the full text. Our approach makes use of a comparatively narrow

receptive field, efficiently covering a single character within a huge image by using the special attribute of focusing on characters. This unique method guarantees the strong identification of texts with scale variations, demonstrating the effectiveness of CRAFT in addressing scale difficulties in text recognition tasks.

In tackling the multilingual problem, our model has issues with Arabic and Bangla characters included in the IC17 dataset. These languages present difficulties for segmenting individual characters because of their cursive writing style. The absence of representation for Arabic and Bangla in the synthetic text dataset makes it more difficult for the model to discern and segment these characters correctly in comparison to Latin, Korean, Chinese, and Japanese characters. The complexities of cursive scripts impact the model's performance, in contrast to the simplicity of distinguishing East Asian characters with constant width.

We conducted single-scale trials on all datasets, despite the wide range of text sizes. This approach differs from other strategies that use multi-scale tests to address scale variance. Our strategy localises particular characters rather than the entire text, providing this advantage. CRAFT's narrow receptive field can encompass a single character in a big image, making it effective for spotting scale variant texts.

We conducted the accuracy measurement through confidence score metric and have achieved more than 0.9 score for our model.

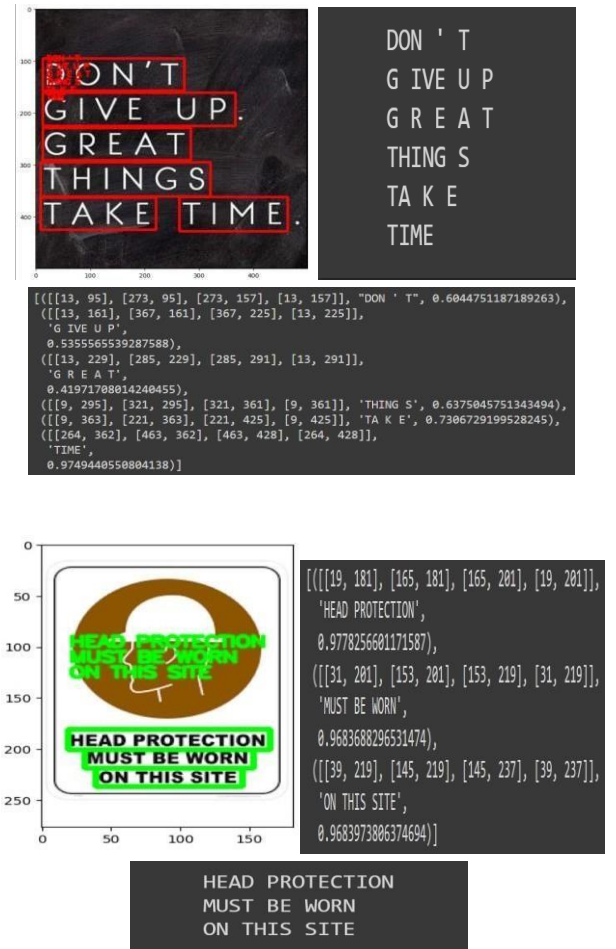


Figure 4: Output

V. CONCLUSION

We offer in this study a novel text detector, named CRAFT, which is able to recognise individual characters even in the lack of annotations at the character level. The method covers a wide range of text shapes in a bottom-up way and generates character affinity scores and character region ratings using a two-score methodology. In light of the lack of genuine datasets with character-level annotations, we propose an ill-guided learning strategy.

This technique generates false-ground truths from an interim model, hence resolving the issue of limited annotated datasets. Notably, CRAFT exhibits its tenacity and generalisation skills by achieving cutting-edge performances across many public datasets without the requirement for fine-tuning.

An innovative neural network architecture is The neural network that is Convolutional Recurrent (CRNN) that combines best features of Recurrent and Convolutional neural networks (CNN and RNN), is also introduced in this research. CRNN can process images with different dimensions as input and produce predictions with different lengths. The most remarkable thing about it is that it works directly on coarse-level labels like words, so it doesn't require specific annotations for particular pieces like letters during training.

Moreover, CRNN departs from classical neural networks by forgoing completely linked layers, which produces a more condensed and effective model. All of these characteristics put CRNN in a highly advantageous position for image-based sequence recognition.

Easy OCR is a feature-rich optical character recognition (OCR) tool that is designed to satisfy a wide range of

requirements. Its advanced features include multilingual support, which enables the recognition and extraction of text from images in numerous languages, the ability to handle noisy images, read licence plates from grayscale photos, batch processing, document digitization, and much more.

REFERENCES

- [1] "Character-Aware Sampling and Rectification for Scene Text Recognition," by M. Li, B. Fu, Z. Zhang, and Y. Qiao, *IEEE Transactions on Multimedia*, vol. 25, pp. 649-661, 2023, doi: 10.1109/TMM.2021.3129651.
- [2] D. Coquenat, C. Chatelain, and T. Paquet, "End-to-End Handwritten Paragraph Text Recognition Using a Vertical Attention Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 508-524, January 1, 2023, doi: 10.1109/TPAMI.2022.3144899.
- [3] In *IEEE Access*, vol. 11, pp. 39621-39634, 2023, R. D. R et al., "Self-Adaptive Hybridised Lion Optimisation Algorithm With Transfer Learning for Ancient Tamil Character Recognition in Stone Inscriptions," doi: 10.1109/ACCESS.2023.3268545.
- [4] In *IEEE Access*, vol. 11, pp. 58406-58421, 2023, doi: 10.1109/ACCESS.2023.3283340, Q. -D. Nguyen, N. -M. Phan, P. Krömer, and D. -A. Le, "An Efficient Unsupervised Approach for OCR Error Correction of Vietnamese OCR Text,"
- [5] "A Two-Level Rectification Attention Network for Scene Text Recognition," by L. Wu, Y. Xu, J. Hou, C. L. P. Chen, and C. -L. Liu, *IEEE Transactions on Multimedia*, vol. 25, pp. 2404-2414, 2023, doi: 10.1109/TMM.2022.3146779.
- [6] A Hybrid Approach for Mathematical Expression Detection in Scientific Document Images, B. H. Phong, T. M. Hoang, and T. -L. Le, *IEEE Access*, vol. 8, pp. 83663-83684, 2020, doi: 10.1109/ACCESS.2020.2992067.
- [7] "Cursive Text Recognition in Natural Scene Images Using Deep Convolutional Recurrent Neural Network," A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, *IEEE Access*, vol. 10, pp. 10062-10078, 2022, doi: 10.1109/ACCESS.2022.3144844.

Easy OCR Applied in Digital Format for Text Extraction Using Deep Learning Technology

Soumyadeep Das, Kripashankar Manglam, Dr. Kamatchi K S

Sathyabama Institute of Science and Technology Chennai, India

Abstract- It describes an OCR effort aimed at creating a sophisticated system for precise and effective text recognition across a variety of documents and languages is shown in this abstract. By leveraging cutting-edge machine learning methods like multilingual training, attention mechanisms, and deep neural networks, the system seeks to outperform conventional OCR systems. Convolutional and recurrent neural networks are trained, preprocessed, and extensive data collecting are all part of the methodology. The versatility of the OCR system is demonstrated by its capacity to adapt to a wide range of scripts, including Latin, Cyrillic, Chinese, and Devanagari. When compared to benchmark datasets, evaluation shows significant advances in recall, accuracy, precision, and F1-score, often reaching human-like levels. The real-world uses include data entry automation, information retrieval optimization, digitizing archives, and assistance for the blind. In conclusion, this OCR project represents a substantial development in text recognition technology and offers improved processing accuracy and efficiency for a variety of document applications.

Index Terms- EASYOCR, Optical Character Recognition, Text Extraction, Deep Learning, CRNN, CRAFT.

I. INTRODUCTION

The constant digitization of information in the modern era has highlighted the crucial role that optical character recognition (OCR) systems play in smoothly transforming handwritten or printed text into formats that are readable by machines. Even though they are invaluable, traditional OCR algorithms frequently struggle with sophisticated character identification jobs, different font styles, and diverse document layouts. This journal article presents a state-of-the-art OCR study that aims to overcome these constraints and push the limits of processing speed and accuracy. The suggested OCR system seeks to redefine the discipline by utilising deep neural networks, powerful machine learning algorithms, and natural language processing approaches.

The main goal of this study is to enhance the system's text recognition accuracy and adaptability to different scripts and languages. Because convolutional and recurrent neural networks (RNNs and CNNs) are used, for feature extraction and order modelling, correspondingly, the OCR system can handle complex document layouts and a wide range of language forms. By allowing the model to concentrate on pertinent passages of the text, attention processes are included, improving the recognition accuracy overall.

The OCR system uses multilingual training and transfer learning methods to handle the multilingual nature of modern

documents. This methodical technique guarantees character identification in a wide range of scripts, such as

Chinese, Latin, Cyrillic, Devanagari, and more. The project's methodology includes painstaking data gathering, preprocessing, and training on a variety of datasets that include handwritten texts, photos, and scanned documents.

Thorough testing against real-world documents, handwritten samples, and benchmark datasets is part of the evaluation process for the built OCR system. The robustness and effectiveness of the system are indicated by performance metrics like the confidence score. The outcomes demonstrate notable progress over current OCR algorithms, frequently attaining accuracy levels that are on par with human perception.

Beyond its technical accomplishments, this cutting-edge OCR technology has many practical applications. The significance of this OCR project is multifaceted, ranging from the digitization of historical archives to the facilitation of information retrieval in large-scale document repositories, and from the improvement of automated data entry systems to the enablement of assistive technology for the visually impaired.

This journal article outlines a thorough strategy for OCR technology, utilising cutting-edge techniques to improve text recognition's accuracy, versatility, and efficiency. If this initiative is successful, document processing could undergo a revolution. This would create new opportunities for the

smooth integration of text data into different applications, which would greatly increase productivity in the digital age.

II. RELATED WORK

Ming Li and Bin Fu (2023) suggest an application for Character-Aware Sampling and Rectification (CASR) that makes use of precise character location and orientation data to handle irregular text occurrences. With this novel method, CASR predicts character-level properties to direct the sampling and rectification procedures by treating every character as a fundamental unit. In order to enhance the rectification of entire text lines, this module employs a novel technique that makes use of particular character information and character-level monitoring throughout the training phase. Above all, CASR is meant to be an easily integrated plug-and-play module into pre-existing text recognition pipelines. Broad tests on many standards show that the suggested approach is robust and achieves promising overall performance by regularly producing rectified text instances with greater accuracy. With its character-centric methodology, this CASR module presents a significant development in the field of text correction and provides a workable and efficient way to increase accuracy in situations of irregular text.[1]

Denis Coquenat and Clement Chatelain (2023) tackle the problem of computer vision's unrestricted handwritten text recognition. They suggest a hybrid attention end-to-end approach that is intended to process a paragraph picture iteratively in order to identify individual lines. Three modules make up the model: a decoder for character sequence recognition, an attention module for implicit line segmentation, and an encoder for feature map generation. This novel method provides a cohesive answer and demonstrates effectiveness in identifying complete paragraphs in unrestricted handwritten material.[2]

R. Devi and S. Karthikeyan's study from 2023 addresses a crucial pattern recognition research subject, concentrating on the difficulties Tamil characters present because of their complex properties. Important historical information can be found in Tamil Nadu's stone inscriptions, but the characters' holes, loops, and curves make it difficult to read the writing. The proposed method combines a Deep Convolution Neural Network and Transfer Learning (TL) for character recognition. By successfully identifying characters from stone inscriptions, the hybrid method shows promise in addressing the technical issues linked to Tamil character identification.[3]

The work of Quoc-Dung Nguyen and Nguyet-Minh Phan provides a fresh unsupervised method to OCR mistake correction. Their method produces and investigates rectification candidates near OCR errors using a hill-climbing algorithm. Interestingly, corrective characters are taken out of OCR training data and only from original ground truth texts.

The proposed model employs a weighted objective function that is heuristically examined for the optimal weight combinations in order to score and rank repair possibilities. An OCR dataset from the Vietnamese handwritten text recognition competition held online in 2018 by ICFHR is used to assess the model's efficacy. Furthermore, the suggested method's stability and complexity are confirmed. This new unsupervised OCR error correction model is a viable way to improve OCR applications' accuracy and dependability.[4]

Lintai Wu and Yong Xu propose A Two-Level Attention Network for Rectification (TRAN) for text rectification and r acknowledgment. Two-Level Network for Rectification (TORN) and Attention-Based Recognition Network (ABRN) comprise TRAN. By addressing geometric restrictions and making pixel-by-pixel corrections for clarity, TORN corrects text geometry. ABRN specialises in text recognition in photos that have been rectified. To enhance feature extraction and effectively handle variations in channel interdependencies and character size, a novel both kernel- and channel-level attention units is integrated into the prototype. This novel method provides a thorough resolution to geometric restrictions and unpredictability in text recognition situations.[5]

III. PROPOSED SYSTEM

A simple and flexible optical character recognition (OCR) module for Python, Easy OCR is easy to use. Easy OCR is a versatile tool that is designed to cater to the needs of both novice and expert developers in OCR and computer vision. It supports a multitude of languages and offers pre-trained models for text recognition and identification.

The library is a solid option for Python developers because it places an emphasis on speed and efficiency while identifying text within images. Easy OCR's ability to handle a wide range of typefaces and text layouts with precision and robustness makes it an invaluable tool for automating text-related tasks and improving data extraction from scanned documents.

With this technique, developers can focus on the unique requirements of their products while easily integrating OCR capabilities into a variety of Python projects, such as desktop software and online apps. For utilising OCR in Python, Easy OCR proves to be a useful and approachable option, particularly for computer vision applications.



Figure 1: Easy OCR Framework

1. Preprocess the Image

An important first stage in the Easy OCR pipeline is image preprocessing, which improves the quality of the images in preparation for text recognition. Several techniques are used in this approach to overcome common image issues.

Noise reduction is the first and most important duty of image preparation in Easy OCR. Eliminating unnecessary noise is essential to boosting the image's overall clarity and, in turn, the accuracy of text recognition. A variety of complex filters, such as median and Gaussian blur, are deliberately utilised to accomplish this. With the effective removal of unnecessary components, the image becomes cleaner and more suitable for precise text recognition.

Easy OCR uses binarization algorithms to convert the grayscale image into a binary representation after noise reduction. Preparing the image for more efficient text recognition algorithms is the goal. Adaptive thresholding and Otsu's method are two of the methods used to accomplish this transition. The result is a binary image that helps with next steps in the OCR pipeline by emphasising text parts against a transparent background.

Skew correction is the last step in Easy OCR's picture preprocessing process. Finding and fixing the image's skew angle is essential to guaranteeing that the text elements are aligned correctly. Recognising skewed text can be difficult, and skew angle correction improves the OCR system's overall accuracy. For skew correction, Easy OCR uses complex algorithms to align the text horizontally and produce a picture that is well-aligned. This correction guarantees that the text is consistently presented and prepared for precise recognition, which is crucial for later stages of the OCR process.

2. Detect the Regions of the Image that Contains Craft

A powerful and effective model for precise text detection in a variety of fonts, sizes, and orientations is Text detection using Character-Region Awareness (CRAFT). Two important output maps are produced by CRAFT using Convolutional Neural Networks (CNNs): Score maps for character regions and affinity scores.

The basis of CRAFT is the Character Region Score Map, a binary representation that was meticulously constructed to show the chance that each pixel in the image belongs to a character region. This map offers valuable insights into the spatial distribution of textual elements and is an effective tool for localising individual characters. CRAFT provides the groundwork for accurate and effective text detection in photos by efficiently identifying character regions.

CRAFT presents the Affinity Score Map as an addition to the Character Region Score Map. When assessing the probability of adjacency between two letters in the same text instance,

this binary map is essential. The map helps organise individual characters into meaningful and cohesive text parts by capturing the affinity between characters. The model's capacity to comprehend and distinguish words inside the image is improved by this grouping into a single instance, or single instance word.[6]

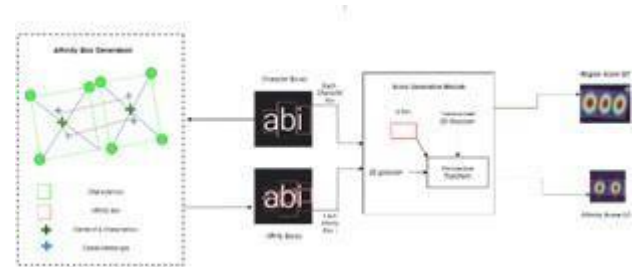


Figure 2: Score maps for character regions and affinity scores

3. Set Up Regions and Identify Text (Crnn)

The Python-based OCR programme EasyOCR presents a robust three-component system that includes sequence labelling, decoding, and feature extraction. This novel method uses deep learning models—ResNet and VGG in particular—to extract important features from input images, which serve as the foundation for reliable word detection in a variety of visual settings.

Because of its three-pronged strategy and strong structure, Easy OCR is positioned as a cutting-edge method for precise and effective text extraction from photos. The incorporation of deep learning techniques alongside the CTC algorithm demonstrates Easy OCR's dedication to propelling the Optical Character Recognition domain forward.

Easy OCR is an advanced optical character recognition system (OCR) module that improves recognition performance by skillfully extracting features. Taking advantage of Resnet's and VGG's deep learning expertise, Easy OCR starts with Resnet's novel method of using residual networks and skip connections to reduce vanishing gradients and allow for deep network depth. Concurrently, VGG offers its consistent architecture, which makes use of small 3x3 convolution filters to create a straightforward but effective design. This combination of VGG and Resnet strengthens the module extraction of Easy OCR which reveals a dynamic synergy that enables the model to decipher complex visual patterns and improve recognition task accuracy.

Easy OCR smoothly moves from feature extraction to sequence labelling, where Long Short-Term Memory (LSTM) networks are the main players. Specialized recurrent neural networks, or LSTMs, are excellent at simulating temporal dependencies across long sequences. Their unique architecture solves the vanishing gradient issue that plagues conventional

RNNs. It includes memory cells and three essential gates (input, forget, and output). This enables LSTMs in Easy OCR to perform sequence labelling jobs effectively, enabling more accurate and efficient nuanced information processing over longer sequences.

Easy OCR uses Connectionist Temporal Classification (CTC) for decoding at the end of the recognition model. For sequences with ambiguous temporal alignments, such as those seen in speech and handwriting recognition tasks, CTC functions as a specialized loss function. By inserting a blank label alongside preexisting labels, CTC allows for variable-length predictions, which is especially helpful in situations when there is uncertainty about the alignment between labels and input data. Because of this feature, CTC is perfect for OCR decoding because it calculates loss by adding up all potential alignments between the input and target sequences. The training pipeline, which is based on the deep-text-recognition-benchmark architecture, increases the adaptability and efficiency of Easy OCR and permits reliable training on a variety of datasets.[7]

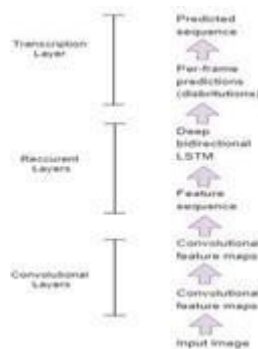


Figure 3: Network Architecture

4. Decode the Output of the Recognition Model

A decoding method becomes essential to the decoding process once each region of the input image generates a probability distribution over characters, particularly in the context of the Connectionist Temporal Classification (CTC)-based decoder.

The goal of this technique is to use the output probabilities to estimate the most likely character sequence. Interpreting the model's predictions and turning them into a logical and precise representation of the recognized text depends heavily on the decoding process. In order to improve the general effectiveness and dependability of the Optical Character Recognition (OCR) system in Easy OCR and guarantee precise and successful outcomes, this strategic decoding technique is crucial.

The Beam Search Decoder and the Greedy Decoder are two decoding techniques used in the recognition process. At each

sequence step, the Greedy Decoder determines which character is the most likely based on a probability distribution over all potential characters generated by the recognition model. With this method, the character with the highest probability is chosen at each stage and added to the final recognized text.

Conversely, the Beam Search Decoder improves accuracy by taking into account several options at every stage of the sequence. It reduces errors caused by ambiguity between similar characters or symbols, unlike the Greedy Decoder, by avoiding the restriction of choosing only the most likely character. Easy OCR's text recognition accuracy is enhanced by the Beam Search Decoder's capacity to investigate many choices at every stage.

5. Postprocess the Recognized Text

In order to increase the precision and calibre of the recognized text output, post-processing in OCR entails using extra methods or algorithms. Resolving mistakes or discrepancies in the OCR findings is its main goal.

Post-processing is a component of Easy OCR's pipeline that aims to improve readability and accuracy of the recognized text. Spell checking is a common post-processing approach where misspelt words are fixed by utilizing statistical language models or consulting a dictionary.

Language-specific rules or statistical models are used in language modelling to enhance coherence and grammar. Furthermore, by transforming acronyms, special characters, and abbreviations into their whole names or standard representations, text normalization guarantees that the accepted text format is standardized. Although Easy OCR incorporates post-processing for text enhancement, several techniques and algorithms may be used to achieve the best outcomes.

IV. RESULTS AND DISCUSSIONS

Our suggested approach, CRAFT (Character Region Awareness For Text identification), stands out in tackling the problem of scale variance by showcasing its resilience in single-scale tests conducted on a variety of datasets. In contrast to numerous other methods that utilise multi-scale testing to address scale variance, CRAFT's advantage is its capacity to localise individual characters as opposed to the full text. Our approach makes use of a comparatively narrow receptive field, efficiently covering a single character within a huge image by using the special attribute of focusing on characters. This unique method guarantees the strong identification of texts with scale variations, demonstrating the effectiveness of CRAFT in addressing scale difficulties in text recognition tasks. In tackling the multilingual problem, our model has issues with Arabic and Bangla characters included in the IC17 dataset. These languages present difficulties for

segmenting individual characters because of their cursive writing style. The absence of representation for Arabic and Bangla in the synthetic text dataset makes it more difficult for the model to discern and segment these characters correctly in comparison to Latin, Korean, Chinese, and Japanese characters. The complexities of cursive scripts impact the model's performance, in contrast to the simplicity of distinguishing East Asian characters with constant width.

We conducted single-scale trials on all datasets, despite the wide range of text sizes. This approach differs from other strategies that use multi-scale tests to address scale variance. Our strategy localises particular characters rather than the entire text, providing this advantage. CRAFT's narrow receptive field can encompass a single character in a big image, making it effective for spotting scale variant texts.

We conducted the accuracy measurement through confidence score metric and have achieved more than 0.9 score for our model.



Figure 4: Output

V. CONCLUSION

We offer in this study a novel text detector, named CRAFT, which is able to recognise individual characters even in the lack of annotations at the character level. The method covers a wide range of text shapes in a bottom-up way and generates character affinity scores and character region ratings using a two-score methodology. In light of the lack of genuine datasets with character-level annotations, we propose an ill-guided learning strategy.

This technique generates false-ground truths from an interim model, hence resolving the issue of limited annotated datasets. Notably, CRAFT exhibits its tenacity and generalisation skills by achieving cutting-edge performances across many public datasets without the requirement for fine-tuning.

An innovative neural network architecture is The neural network that is Convolutional Recurrent (CRNN) that combines best features of Recurrent and Convolutional neural networks (CNN and RNN), is also introduced in this research. CRNN can process images with different dimensions as input and produce predictions with different lengths. The most remarkable thing about it is that it works directly on coarse-level labels like words, so it doesn't require specific annotations for particular pieces like letters during training.

Moreover, CRNN departs from classical neural networks by forgoing completely linked layers, which produces a more condensed and effective model. All of these characteristics put CRNN in a highly advantageous position for image-based sequence recognition.

Easy OCR is a feature-rich optical character recognition (OCR) tool that is designed to satisfy a wide range of requirements. Its advanced features include multilingual support, which enables the recognition and extraction of text from images in numerous languages, the ability to handle noisy images, read licence plates from grayscale photos, batch processing, document digitization, and much more.

REFERENCES

1. "Character-Aware Sampling and Rectification for Scene Text Recognition," by M. Li, B. Fu, Z. Zhang, and Y. Qiao, IEEE Transactions on Multimedia, vol. 25, pp. 649-661, 2023, doi: 10.1109/TMM.2021.3129651.
2. D. Coquenat, C. Chatelain, and T. Paquet, "End-to-End Handwritten Paragraph Text Recognition Using a Vertical Attention Network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 508-524, January 1, 2023, doi: 10.1109/TPAMI.2022.3144899.

3. In IEEE Access, vol. 11, pp. 39621-39634, 2023, R. D. R et al., "Self- Adaptive Hybridised Lion Optimisation Algorithm With Transfer Learning for Ancient Tamil Character Recognition in Stone Inscriptions," doi: 10.1109/ACCESS.2023.3268545.
4. In IEEE Access, vol. 11, pp. 58406–58421, 2023, doi: 10.1109/ACCESS.2023.3283340, Q. -D. Nguyen, N. -M. Phan, P. Krömer, and D. -A. Le, "An Efficient Unsupervised Approach for OCR Error Correction of Vietnamese OCR Text,"
5. "A Two-Level Rectification Attention Network for Scene Text Recognition," by L. Wu, Y. Xu, J. Hou, C. L. P. Chen, and C. -L. Liu, IEEE Transactions on Multimedia, vol. 25, pp. 2404-2414, 2023, doi: 10.1109/TMM.2022.3146779.
6. A Hybrid Approach for Mathematical Expression Detection in Scientific Document Images, B. H. Phong, T. M. Hoang, and T. -L. Le, IEEE Access, vol. 8, pp. 83663-83684, 2020, doi: 10.1109/ACCESS.2020.2992067.
7. "Cursive Text Recognition in Natural Scene Images Using Deep Convolutional Recurrent Neural Network," A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, IEEE Access, vol. 10, pp. 10062-10078, 2022, doi: 10.1109/ACCESS.2022.3144844.