International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Real-Time OCR System for Multilingual Text Recognition Using Camera and Deep Learning

Akshay Gharpure
Department of Science and Technology,
G. H. Raisoni Skill Tech University, Nagpur, India

Abstract

Optical character recognition (OCR) technology plays an important role in the digitization of documents and intelligent data extraction. OCR allows computers to read text that appears inside images and videos. Traditional OCR systems have proven to have a high degree of accuracy when used to read text contained in non-moving (i.e., still) images; however, the use of OCR technology for real-time applications has been severely limited due to limitations on processing power, physical conditions (e.g., low light levels) and quick turnaround times. This paper provides details on the development and implementation of a real-time OCR system that can read text from live cameras using computer vision and convolutional neural networks (CNNs). This real-time OCR system uses CNN-based algorithms along with computer vision techniques to create an effective and scalable real-time OCR solution that can be used in real-life situations.

Using OpenCV, Frames from Live Camera Feed requiring Image Manupulation/ Pre-processing are Captured Then The Text Are Recognised Through (EasyOCR). The Pre-processing Steps Include Grayscale Conversion to Improve Magnitude of the Input Image, However Given The Time Constraints of 'Real Time' Optimisation Because of Frame-Skipping and Scaling Convert some of the Resistance Between Computation Cost and Output Cost While I'm In Multilingual Environments For Example; English and Hindi Throughout Data Collection/Evaluation of The System Using Various Performance Measurments As Well Using Accuracy Data as welll Omega 3 Oil Performance As Far As It's Effects On Cognitive Developement Based On The Results Puls Standars of Labour Cost Are Potential Future Development.

This study has developed an economically viable and viable OCR solution for use in many typical applications in everyday life, including improving assistive technologies, implementing automatic data entry systems, and performing document and image analysis through the use of modern deep learning algorithms and classical image processing methods to create robust OCR systems on platforms with limited resources, while achieving acceptable performance and accuracy levels.

1. Introduction

1.1 The Problem Context

In recent years, digital information systems have been growing rapidly, creating greater demand for automated techniques to extract text from images. Many images and video streams contain important text, such as road signs, business documents, product packaging, and electronic displays that must be quickly interpreted so they can be used in a variety of applications such as automating data entry and assistive technologies. Extracting these texts in real time is extremely difficult because of various factors including; lighting inconsistencies; motion blur; font variation; background noise; and

limitations imposed by computing equipment. In addition, many traditional OCR systems are designed for still images and do not perform well when extended to video.

1.2 Limitations of Established Techniques

Established OCR processes employ a great deal of hand-crafted features and use rule-based segmentation techniques making them sensitive to noise and conditions that change image quality. Often the systems require high-resolution images and a controlled environment in order to achieve reasonable accuracy; furthermore they lack the ability to adapt to different languages or complex image backgrounds. Established real-time OCR systems typically sacrifice accuracy for speed or require proprietary hardware, such as GPUs. None of the established systems can successfully balance performance, accuracy and computational requirements.

1.3 Role of Deep Learning and Computer Vision

1. Convolutional Neural Networks (CNNs), have led to improvements in the performance of Optical Character Recognition (OCR) Since CNNs automatically learn hierarchical feature representations from raw image data, they can be used to detect text in a variety of conditions, including fonts and orientations that may vary, as well as noise.
2. These benefits make CNNs far superior than other neural networks because they enable the use of image processing and object detection techniques to create deep learning-based OCR that is very accurate and can adapt well to different conditions.
3. For example, EasyOCR takes advantage of pretrained CNN models and incorporates the ability to perform text detection as well as text recognition. As a result, easy to use tools can now build end-to-end OCR solutions that require less than high-end computers to execute.

1.4 Purpose

The goals of this study are to design and develop an efficient and real-time OCR system that operates on regular computing devices and does not require the latest computer technology to run smoothly. By supporting both high accuracy for detecting and recognizing text, and good performance through lightweight preprocessing techniques and highly optimized CNN models, we believe that it is possible to bridge a gap between performance and accuracy. Additionally, by supporting a variety of languages, we hope to increase the relevance of the OCR system we develop for the Indian population.

2. Related work

2.1 Traditional OCR Methods

Historically, the major types of Optical Character Recognition (OCR) have utilized two primary approaches: rule-based and feature engineering. They were rule-based systems which required the entire pipeline to be created with manual algorithms for character recognition, text detection, text segmentation, etc. Early OCR used template matching to recognize a character; for each character in the OCR model, there were templates of the characters in the database to compare the input character against. Template matching works well for structured documents (where the font of each character does not change) but provides no flexibility when it comes to different font types (e.g., Arial vs. Times New Roman), sizes (e.g., 12pt vs. 14pt), and orientations (e.g., straight vs. upside down) of the characters.

Another common type of OCR utilizes feature extraction techniques (e.g., Edge Detection, Zoning, Projection Histogram, Contour Analysis) to pull out the unique features of a character, which are then used to classify the character. The classifiers used with these features usually include k-Nearest Neighbors (k-NN) and/or Support Vector Machines (SVM). These methods of OCR do result in better generalization than template matching methods; however, they also require extensive prior

image processing including character binarization, character noise filtering, and character segmentation.

A significant limitation of traditional OCR systems is their reliance on precise segmentation of characters. In many real-world scenarios (but especially when using cameras for capture), segmenting characters becomes too difficult when the background is complex (e.g., a white piece of paper with full color images behind it), and/or characters are overlapping each other (e.g., an image with multiple lines of text). In addition, traditional OCR systems are sensitive to noise, light change, and distortion of the images captured therefore they cannot be used for real-time applications such as video stream-based OCR where the data is considered to be dynamic in nature.

2.2 Deep Learning-based OCR

Deep Learning has revolutionized OCR by introducing a data-driven approach to learning complex patterns from unprocessed images. Unlike traditional methods, which rely on hand-crafted features to extract data about text, deep-learning based OCR systems learn to automatically extract hierarchical representations of text from images via neural networks.

A key player in this transformation has been the use of Convolutional Neural Networks (CNNs) to extract significant features from images. At different layers of the CNN, low-level features such as edges and textures are combined with higher-level features such as character shape and word structure.

In addition to CNNs, Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM) can be built to model sequential features of text, allowing the OCR system to recognize entire words or sentences rather than individual characters.

A major advancement in utilizing deep learning for OCR is the implementation of Connectionist Temporal Classification (CTC). CTC is a sequence prediction technique that does not require direct segmentation of characters in order to recognize the sequence of letters, making it very useful for recognizing characters in natural scenes where characters are not clearly separated from each other. Attention Mechanisms have also been incorporated to increase recognition accuracy by allowing the model to look at smaller subsets of input to identify the desired output.

2.3 CNN Architectures for Text Recognition

Due to their ability to process spatial data from images, CNNs have become the primary technology for contemporary OCR systems. A number of CNN models have been developed specifically for use with text recognition systems that offer different compromises regarding accuracy, complexity, and speed.

In the 1990s, the introduction of the LeNet architecture made it clear that neural network architectures could be used for character classification. Other advanced architectures have been constructed, like VGGNet and ResNet and they provide higher-level features by increasing the number of CONV layers and continuing to use small CONV kernels (or filters) to create finer-resolution images when extracting text from photographs. For example, ResNet has incorporated residual connections to reduce vanishing gradient effect on creating an arbitrarily deep network.

Furthermore, there has been widespread use of hybrid architectures specifically designed for OCR applications, including Convolutional Recurrent Neural Networks (CRNNs). These systems combine traditional CNNs for feature detection with RNNs for sequential modeling to perform end-to-end recognition of text. This is particularly useful when processing images from naturally-occuring scenes where the characters may be joined or spaced irregularly.

Finally, lightweight architectures, like MobileNet, provide similar levels of accuracy while utilizing lower computational complexity. These lower-complexity models make them well-suited for real-time OCR applications on devices with limited resources.

2.4 Existing OCR Systems

Over time, a number of optical character recognition (OCR) systems have been developed using many differing optimization techniques to enhance performance overall. The OCR engine most frequently used is Tesseract OCR originally developed at Hewlett Packard then taken over and continually enhanced by Google. Tesseract has transitioned from a rule based to a deep learning based OCR engine utilizing Long Short Term Memory (LSTM) Networks to perform text recognition. Tesseract will perform highly accurately on multiple languages with structurally formatted documents and generally perform well but has performance restrictions relating to computational power requirements and sensitivity to the quality of incoming text images when used for real time.

EasyOCR is a new but popular framework for developing OCR systems which is very easy to use and provides an effective and efficient method of performing Optical Character Recognition (OCR) for Text Detection and Recognition designed to utilize pre-trained Deep Learning Models (DLM). EasyOCR is available in many languages including English and Hindi and can be readily integrated into any application written using the Python programming language. Due to the superior performance capabilities of EasyOCR and minimal configuration requirements and weight, EasyOCR is highly suited for real time OCR applications especially when using systems without Graphics Processing Unit (GPU) support.

Other optical character recognition (OCR) systems exist such as Google Vision API and Microsoft Azure OCR that provide cloud based OCR solutions that are highly accurate and scalable. However, while very effective, these cloud-based OCR solutions require internet connectivity to function; thus, OCR systems developed using these methods may

2.5 Research Gap

While there has been much progress made in developing OCR technologies there still exists a significant gap in developing systems which are capable of performing optimally in a real-time environment without requiring extensive computing resources. Most currently available "traditional" systems do not have sufficient robustness to be able to handle dynamic input; furthermore, many of the "deep learning" based systems require large amounts of computational power (which requires the use of GPUs). Furthermore, the current solutions have tended to focus either on improving accuracy or improving speed but very rarely provide both in sufficient measure.

As such, this shortfall becomes increasingly important for real-time applications that demand both real-time response times and dependability from an end-user perspective. Another important limitation is the inconsistent multi-lingual capabilities generally exhibited across the different OCR systems when it comes to languages that utilize complex scripts such as Hindi. Finally, there is a significant lack of usable implementations of complete OCR systems (including input acquisition, pre-processing, detection, recognition, and visualization of data) as part of a unified real-time system. Most research projects that have been undertaken tend to be focused on only one or two individual components as opposed to the totality of the OCR process as part of an end-to-end system, which creates significant difficulties with developing usable solutions within real-world scenarios.

This research intends to address those issues by developing a real-time OCR Capability that combines efficient image preprocessing with a CNN-based recognition model configured to operate effectively on standard hardware. This capability will provide multi-lingual recognition capabilities. Additionally this will facilitate developing an integrated development environment for future ‘long-range hands-free’ OCR solution.

3. Research Methodology

3.1 Problem Statement

The primary goal of this project is to develop a real-time OCR system that can automatically identify characters and words displayed in a continuous video stream from an input video camera. An OCR System is a tool that allows the computer to identify and capture text from images, such as those that may be viewed on a printed page or displayed on a computer screen. In this research, the OCR system should function as if it were a camera located at a point in either an indoor or outdoor environment and operating as part of a computer vision project.

The OCR system must take on challenges inherent to dynamically produced input (such as multiple users moving through the same location), including differences in lighting level (brightness), motion artifacts (changes in pixel position over time due to camera defects), complex backgrounds (furniture, other people), different font styles (fonts used for "Hello"), etc. This necessitates that the proposed architecture include an optimum design for achieving the best possible computed recognition accuracy in conjunction with minimal processing time when executed on resource-constrained platforms without any form of graphical processing unit (GPU) accelerators.

The proposed system must process video frames at a speed near real-time and achieve an acceptable level of recognition accuracy when processing video frame images containing both English and Hindi text as input. The system must also deal with different sized text (small/large), rotated text, and noise (i.e., interference with the reception of good data) with no human assistance.

The intent of this project can be formally stated as follows: Given a sequence of frames from a continuous video, $F=\{f_1,f_2,f_3, \ldots ,f_n\}$; generate a set of textual representations, $T=\{t_1,t_2,t_3,\ldots,t_m\}$; thereby, for each t_i, such that t_i represents a valid text region within the original cycle producing a set of T, while, at the same time, minimizing processing delay and maximizing recognition accuracy.

3.2 System Design Architecture

The overall design of the proposed system consists of modularized components to provide greater flexibility and ease of maintenance than traditional systems. These components will provide:

The first phase of the process consists of capturing a live feed of video pictures from a camera (either an external IP camera or a built-in webcam). The video stream is processed as each frame is taken; so, therefore, each frame is treated separately by the system. The second phase is represented by preprocessing of the raw frames; in this phase, the raw frames are modified to allow them to be more appropriate for recognition of text (i.e., grayscale, resize, and reduce noise).

The third stage utilizes a Convolution Neural Network architecture for both text detection as well as recognition; the detection portion of EasyOCR identifies text in specific regions of the image while the recognition portion decodes the detected text to read the content of the detected text in the respective regions of the image. The final stage overlays the bounding boxes, recognized text, and scale of confidence values onto the source image (i.e., displayed with image) providing a visual reference for the result of the pipeline.

Using this methodology allows for a continuous stream of data for the system to use for real-time functions while providing the opportunity for modularity for future improvements to the system.

3.3 Image Processing Methodologies

The types of operations performed on the raw images prior to the recognition of text in the images is important to the success of OCR systems due to the fact that the quality of the images being inputted into the system cannot be controlled as it would be in a real-time fashion. The suggested approach to perform image enhancement techniques on images to improve the quality of those images.

The first step involves converting the input frame from RGB to grayscale. This transformation reduces computational complexity by eliminating color information while retaining intensity variations that are essential for text detection. The grayscale image is then resized to a lower resolution to reduce processing time. However, the resizing operation is carefully chosen to avoid excessive loss of detail, which could negatively impact recognition accuracy.

Filtering methods, such as Gaussian blur, are used for reducing noise in images. Gaussian blur will smooth the image and help erase most of the high-frequency noise. In addition to filtering, thresholding may also be applied to improve contrast of the text and the background. These two processes should improve clarity of the text regions and allow for easier extraction of meaningful features within the text by the CNN model.

A very important consideration is that the frames are optimized before being passed to the model for processing. Rather than process every frame, the system uses frame skipping, processing only a subset of frames that are spaced by a regular interval of time. This greatly reduces the amount of computation needed while maintaining an adequate temporal continuity of the output.

3.4 CNN Model Architecture

The Convolutional Neural Network (CNN) is the primary method used to detect text and recognize text for the OCR system; and the EasyOCR framework uses pre-trained models, built upon Deep Learning, to combine convolutional layers for feature extraction with sequence modeling for recognition.

The CNN structure is made up of multiple Convolutional layers, each followed by an Activation function (such as ReLU) and Pooling layer for reducing the dimensionality of the data. As you go through each of these layers, the CNN will extract progressively more complex hierarchical features from the input image by capturing Edges, Textures and Complex Patterns associated with text characters. Finally, the extracted features are sent to one or more Fully Connected layers which classify the features by associating them with the likelihood of them being a certain character.

Along with the convolutional component of the network itself, it also utilizes a sequence modeling component that allows the network to recognize complete words or lines of text instead of just single characters. This provides additional accuracy through the use of contextual information.

The data to train the model comes from a large-scale, multilingual dataset of text, allowing the model to generalize well across varied languages and scripts.

Using a pre-trained model allows the implementation of real-time applications on standard computational resources due to less training and lower computational requirements.

3.5 Mathematical Formulation

Mathematically, the OCR process can be expressed in function form as an image I, which is transformed into a sequence of characters through the feature extraction from the CNN, represented by the function $\phi(I)$. Finally, the output generated by the network is a probability distribution defined over the possible character sequences.

The predicted output, Y, is obtained by applying a softmax function to the output of the network as follows:

$Y = \arg\max_c P(c \mid \phi(I))$

where $P(c \mid \phi(I))$ is equal to the probability of character sequence given the features extracted from the input image.

For sequence-based recognition, it is common practice to utilize Connectionist Temporal Classification (CTC) loss to predict values of outputs that vary in length while not requiring to perform explicit segmentation. The calculated accuracy is:

Accuracy = (Number of Correctly Predicted Outputs / Total Number of Outputs)

Step 2: Change the picture to black and white

Step 3: Resize image so it can be processed more quickly

Step 4: Use software to reduce the noise and apply a threshold

Step 5: Send the image to the EasyOCR for text recognition.

Step 6: Identify the regions of the image to detect text and recognize the text characters.

Step 7: Draw bounding boxes around the characters and display the results.

Step 8: Go back to step 3 for the next image and use any optimizations that can be accomplished.

4. Performance Evaluation

4.1. Performance *Evaluation Measures*

Evaluation of the performance of a real-time Optical Character Recognition system should be based on an analysis that incorporates many dimensions (e.g., speed, accuracy, reliability, etc.) of performance in addition to the actual ability of the system to recognize text characters or words from video images. To accomplish a complete evaluation of this system, the use of both quantitative (number of correct words or characters out of the total number) and qualitative (descriptive) measurement techniques (metrics) will allow one to evaluate how the system performs in a complete manner.

Recognition accuracy is the main metric used for this evaluation as it provides a measure of how many words (or characters) have been recognized by the system when compared to the total number of words (or characters) present in the video images fed into the system. Recognition accuracy is very sensitive to error and variations in font style and amount of noise or brightness which makes it an important performance measure when evaluating the reliability of an OCR system.

Latency will also be measured during the performance evaluation to provide a measure of how long it takes the system to acquire a video frame through the process of capturing, processing and displaying that same video frame for output. The amount of time it takes the system to process each video frame is referred to latency, which has a detrimental effect on real-time use of this OCR system.

Moreover, the use of a percentage of confidence score will enable evaluation of reliability of EasyOCR model's outputs. For each piece of text that EasyOCR identifies, there is a confidence score associated with it; this gives insight into how certain EasyOCR is that it has correctly identified what it thinks it has detected. These confidence scores are then aggregated together and examined so that patterns can be recognized whereby one can analyse the reliability of EasyOCR across different situations/scenarios. The error rate is also taken into account as an additional metric; this is defined by the ratio of number of characters that have not been correctly recognized by EasyOCR to total number of characters processed.

The evaluation framework mentioned above uses both of these metrics as they offer a rigorous way to measure EasyOCR for both its efficiency and effectiveness.

4.2. Experimental *Methodology*

The experimental process has been specifically designed to mirror an actual work environment as closely as possible while ensuring reproducible results through consistency in how the evaluation occurs. The system will be tested on an ordinary computer set up consisting of an Intel Core i5 processor, 8 GB of RAM, integrated video graphics (without GPU acceleration). This system has

been specifically selected because it is commonly available to students/small scale users and therefore demonstrates the viability of implementing this application on devices that have limited resources. The software environment will consist of writing program code in Python, using OpenCV for processing images, and using EasyOCR as the deep learning framework for identifying text within an image.

The input into the system will come through an application that provides an IP webcam view.

For efficient evaluation of how various environmental settings and attributes of the action/character recognition system (e.g., speed) affect operation, numerous test cases were specified to allow for systematic analysis of test results. Each of the multiple instances of a test case (the major variables that were changed for the tests included: pre-processing variable set up, e.g., frame size/type of processing, etc.) will provide an average value to minimize the effect of random variation. As an example of this approach, numerous tests will be performed using a resolution of 640×480 and then again using a resolution of 320×240 to compare/contrast the quality of detection accuracy versus processing speed.

Similarly, multiple frame skipping values (1, 2 or 3) will be used to evaluate the relationship between the amount of data processed in the system to the processing speed of the system. The overall goal of this approach is to create a number of experiments based on the selection of frame skipping and pre-processing values that will allow the researcher to complete a thorough review of how well the system behaves through a variety of configurations.

4.3. Results

The results from the experimental study were subjected to an in-depth analysis to develop a clear understanding of the factors influencing the characteristics of performance of the proposed system. As part of this analysis, rates of recognition accuracy for the proposed action/character recognition system were collected for multiple configurations of operating conditions. An overview of the accuracy based on environmental illumination conditions can be viewed in Table 1.

Table 1: Recognition Accuracy Based on Environmental Illumination Condition

Lighting Condition	Character Accuracy (%)	Word Accuracy (%)
Good Lighting	92.5	89.2
Moderate Lighting	85.3	81.7
Low Lighting	72.1	68.4

Overall, the results suggest that the action/character recognition system operated best under conditions when there was sufficient contrast between the text and its background.

Performance Metrics for Real-Time Processing

Table 2 shows the performance metrics of different FPS (Frame Per Second) ranges for video processing at different resolutions and averaging the latency (in milliseconds) experienced.

Resolution	FPS Range	Average Latency (ms)
640×480	10–12	85–110
320×240	15–18	55–75

This is consistent with the other metrics which show positive interaction between video resolution and the FPS range and latency that are experienced in the application for processing video images in real-time. When using lower resolution video images, the user will experience increased FPS and decreased latency, allowing the user to achieve much smoother and more reliable operation during real-time activities. At the same time, due to the lower resolution they are utilizing, the accuracy of the images being processed for smaller or finer text on the video frame will be reduced. As such, the video resolution selected for processing video will depend on the requirements of the individual application.

Confidence Score Distribution

Further analysis into the confidence scores of the detections (clearly identified in Table 3) also support this conclusion.

Table 3 shows the confidence score distribution of the detections.

Confidence Range	Percentage of Detections (%)
0.90 – 1.00	58
0.75 – 0.89	27
0.50 – 0.74	10
Below 0.50	5

The confidence range for the majority (58%) of detections are between 0.90 and 1.00, suggesting that the model provides a reliable model for processing video images. However, lower confidence detections were found to occur when processing video images of complex backgrounds and/or text with stylized fonts. These results provide opportunities for model enhancement.

4.4. Comparative Analysis

A comparative analysis of the proposed system was conducted to an alternative methods which includes manually entering information into a database and/or using conventional OCR methods. This analysis provides additional information on the advantages and disadvantages of the proposed system when compared to other methods across multiple types of applications.

Table 4. Comparison of Different Techniques

Technology	Precision (%)	Speed of Processing	Usefulness
Manual Input	100	Very Slow	Unlikely
Traditional OCR	78	Moderate	Moderate
Deep Learning OCR (GPU)	94	Fast	High
Proposed System	88	Fast	High

Overall performance of the different technologies is as follows: While 100% accuracy is achieved through manual methods, this is not suitable for any type of "real time" application because this method is much slower than all other methods; Traditional OCR technology provides moderate accuracy; however, the system does not perform well when there is a dynamic environment; Deep Learning OCR technology with the use of a GPU provides much higher levels of accuracy and requires high price hardware to do so; where as, the technology used in the proposed prototype system provides good precision and high levels of speed without the need for any special type of hardware.

4.5. Evaluation of Robustness

To evaluate the robustness of this system against difficult testing environments, the system has been tested under different conditions, including motion blur, various orientations of text (the text will be at different angles) and complex backgrounds. The results of the test show that this system is quite resilient to moderate amounts of noise and distortion and continues to perform at an adequate level in the majority of testing environments. However, extreme environmental conditions (rapid movement of camera or very low levels of light) created problems with performance of this system.

Another test to evaluate robustness was performed by having the system process English and Hindi language (multilingual) text. The results indicate that the proposed system is capable of processing English and Hindi language text, but that the accuracy of reading the Hindi language is lower than the accuracy achieved when reading English text due to the complex nature of the Hindi letters/characters and the limited amount of training data available to train the system.

5. Conclusion and Future Work

The research presented in this paper describes how to create a working real-time OCR system that can extract textual data from live streaming video using ordinary computer resources. By using OpenCV based image preprocessing techniques with a CNN based recognition framework developed by EasyOCR, our proposed system provides a compromise between computing efficiency and recognition accuracy. This combination of technologies provides a proof-of-concept for performing real-time text detection and recognition without using high-end hardware like GPUs; hence, our system is available for an extensive array of use cases and user populations.

The results of the experiments conducted during this research show that our system is effective across multiple environmental conditions as evidenced by its ability to remain accurate and responsive through several different types of environmental challenges. The use of optimization techniques such as frame skipping and resolution reduction will yield real-time performance; however, these techniques will create certain compromises that will have to be considered. An analysis of performance metrics such as accuracy, latency, and FPS provides an all-encompassing picture of how the system functions, and highlights areas where further improvement may be required.

The system itself has some limitations such as its inability to handle extreme lighting issues, motion blur, and fonts that are excessively stylized; however, the system can work with text blocks up to 500 characters long.

5.1. Future *Work*

1. By adding GPU acceleration, processing speeds will significantly increase, allowing more extensive, sophisticated deep learning models to be utilized, resulting in increased accuracy and real-time performance.
2. Advanced preprocessing methods, such as adaptive thresholding and image enhancement, would help make the system more reliable when used in difficult environments.
3. Expanding support for other languages, including regional Indian languages, would make it possible to deploy the system in a greater variety of circumstances.
4. Using text tracking algorithms between frames will reduce unnecessary work and increase the overall efficiency of the system.
5. Using more advanced deep learning architectures, such as transformers, will increase overall recognition accuracy regarding difficult and stylized text.
6. Implementing a mobile or embedded version of the system will provide more access to users and enable its usage in real-world scenarios.
7. Developing user feedback systems for correcting errors will improve the system as a whole through continuous learning via feedback.

References

1. R. Smith, “Tesseract OCR Engine Overview,” IEEE, 2007.
2. A. Graves, et al.; “Connectionist Temporal Classification,” ICML, 2006.
3. K. Simonyan and A. Zisserman; "Very Deep CNNs," 2014.
4. OpenCV Documentation (2024).
5. EasyOCR Documentation (2024).
6. Tensorflow Documentation (2024).
7. J. Redmon, et al.; “YOLO: Real-Time Object Detection,” 2016.
8. Baek, et al.; "What is Wrong with Scene Text Recognition Model," 2019.
9. Shi, et al.; "CRNN for OCR"