

PAPER • OPEN ACCESS

Analysis of Optical Character Recognition using EasyOCR under Image Degradation

To cite this article: M.A.M. Salehudin *et al* 2023 *J. Phys.: Conf. Ser.* **2641** 012001

View the [article online](#) for updates and enhancements.

You may also like

- [Smart Parking System based on Improved OCR Model](#)
Rami Bassam and Fars Samann
- [Insights into Charge-Redistribution in Double Layer Capacitors](#)
Ganesh Madabattula and Sanjeev Kumar
- [Food Code Breaker \(FCB\) – Developing the Multi-Lingual Food Code Translation Android Application Using Multi-Options Code Reader System](#)
Sharmini Abdullah, Mohamed Elshaikh, Muhammad Bazli Mahmood et al.

PRIME
PACIFIC RIM MEETING
ON ELECTROCHEMICAL
AND SOLID STATE SCIENCE

HONOLULU, HI
Oct 6–11, 2024

Abstract submission deadline:
April 12, 2024

Learn more and submit!

Joint Meeting of
The Electrochemical Society
•
The Electrochemical Society of Japan
•
Korea Electrochemical Society

Analysis of Optical Character Recognition using EasyOCR under Image Degradation

M.A.M. Salehudin¹, S.N. Basah^{1,2}, H. Yazid³, K.S. Basaruddin^{1,2},
M.J.A. Safar⁴, M.H. Mat Som¹, K.A. Sidek⁵

¹ Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis (UniMAP),
02600 Arau, Perlis, Malaysia

² Centre of Excellence for Intelligent Robotics and Autonomous Systems (CIRAS) Universiti
Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia,

³ Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis (UniMAP), 02600
Arau, Perlis, Malaysia.

⁴ Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis (UniMAP), 02600
Arau, Perlis, Malaysia.

⁵ Department of Electrical & Computer Engineering, International Islamic University,
Malaysia.

E-mail: m.aqilmirza399@gmail.com, shafriza@unimap.edu.my, hanizayazid@unimap.edu.my,
juhairi@unimap.edu.my, khsalleh@unimap.edu.my, mhanafi@unimap.edu.my,
azami@iiu.edu.my

Abstract. This project explores EasyOCR's performance with Latin characters under image degradation. Variables like character-background intensity difference, Gaussian blur, and relative character size were tested. EasyOCR excels in distinguishing unique lowercase and uppercase characters but tends to favor uppercase for similar shapes like C, S, U, or Z. Results showed that high character-background intensity differences affected OCR output, with confidence scores ranging from 3 % to 80%. Higher differences caused confusion between characters like o and 0, or i and l. Increased Gaussian blur hindered recognition but improved it for certain letters like v. Image size had a significant impact, with character detection failing as sizes decreased to 40% to 30% of the original. These findings provide insights into EasyOCR's capabilities and limitations with Latin characters under image degradation.

1. Introduction

This paper examines the advancements, limitations, and potential improvements in Optical Character Recognition (OCR). OCR has transformed data entry, document processing, and information retrieval. The research aims to understand OCR's limits and robustness by analyzing its performance under various parameters. It focuses on characters with similar structures, image size, intensity differences, and Gaussian blur. EasyOCR is used for analysis, employing Latin character images in Times New Roman font. Python and Visual Studio Code facilitate implementation. OCR's evolution, applications, and controlled experiments are discussed to enhance understanding and advancements in the field.

By undertaking this research, we aim to contribute to the existing knowledge on OCR performance analysis and provide insights into the strengths and weaknesses of OCR algorithms. Through our experiments and data analysis, we aim to shed light on the factors that influence



OCR accuracy and explore potential avenues for enhancing the robustness of OCR systems. Ultimately, this research holds the potential to drive advancements in OCR technology, facilitating more accurate and reliable character recognition in various applications.

2. Literature Review

2.1. Online OCR Tools

OCR has gained prominence for digitizing physical documents, but the accuracy of free online OCR services varies. In a study by Dr. Vijayarani and Ms. Sakila, OCR services achieved over 95% accuracy for normal characters but struggled with special characters, resulting in 100% error rates [1]. This reveals challenges in recognizing specialized characters. While OCR excels with normal Latin script, handling special characters remains a developing aspect. Advancements in OCR algorithms are expected to enhance special character recognition. Their findings in figure 3 underscores higher detection rates for normal characters compared to special symbols.

Algorithm 1 The algorithm of direct k -means clustering
Input: N objects in the R -dimensional space, and M (number of sub-cluster means)
Output: Initial M sub-clusters that contain all N objects
initialize M first means $r_0^1, r_0^2, \dots, r_0^M$
repeat
 for each input object $r_i, 1 \leq i \leq N$
 do
 assign r_i to D_j with nearest mean r_0^j , such as
 $\phi(r_i, r_0^j) \leq \phi(r_i, r_0^u), 1 \leq j, u \leq M$
 for each sub-cluster $D_j, 1 \leq j \leq M$
 do
 recalculate the mean (centroid) of objects $r_i \in D_j$,
 $r_0^j = \sum_{i \in D_j} r_i / |D_j|$, where $|D_j|$ defines the cardinality
 of D_j
 compute the quality function G
 until no object has changed sub-clusters (or G does not
 change)

Figure 2.1. Sample Input to an OnlineOCR test [1] . The figure is to test the OCRs capabilities to see if it can detect and output the correct transcription.

Algorithm 1 | he algorithm of direct k-means clustering Input: N objects in the R -dimensional space, and M (number of sub-cluster means) Output: initial AI sub-clusters that contain all N objects initialize AI first means (1), (4,2), (__, 4*) repeat for each input object $r_i, 1 \leq i < do$ assign r_i to p_i with nearest mean (4e), such as $q((r_i,40) cp(r_i,4^*)), 1, j, u <$ for each sub-cluster $D_i, 1 \leq j \leq M$ do recalculate the mean (centroid) of objects $r_i \in p_i, t_{0i} = \text{fill}(p_i)$, where $|D_j|$ defines the cardinality of p_i compute the quality function until no object has changed sub-clusters (or G does not change)

Figure 2.2. Output of figure 1. Figure 2 to shows that while the OCR can detect most of the words and letters, it failed to detect the mathematical notations.

S.No	OCR Tools	Character Accuracy (CA) (%)	Character Error Rate (CER) (%)	Special Symbols Accuracy (ESA) (%)	Special Symbol Error Rate (SER) (%)
1	Online OCR	95.9	4.10	0	100
2	Free Online OCR	98.64	1.36	0	100
3	OCR Convert	100	0	0	100
4	Convert image to text.net	100	0	0	100
5	Free OCR	100	0	0	100
6	i2OCR	100	0	0	100
7	Free OCR to Word Convert	23.29	76.71	0	100
8	Google Docs	100	0	0	100

Figure 2.3. Overall OnlineOCR Results [1] . The results of their assessment show that most OCR Tools available online have a high accuracy rate for normal characters but decline for special letters.

2.2. Document image OCR accuracy prediction via latent Dirichlet allocation

The paper "Document image OCR accuracy prediction via latent Dirichlet allocation" offers a novel approach to estimating OCR accuracy by focusing on document image quality. Unlike conventional methods, it utilizes unsupervised features derived from Latent Dirichlet Allocation (LDA) instead of predefined features for specific degradations. The model's strength lies in its ability to automatically predict OCR accuracy without reference images, presenting a departure from rule-based degradation approaches [2]. Although the study's primary focus is not EasyOCR or image degradation, its methodology of leveraging unsupervised features through LDA aligns with our exploration of "Analysis of Optical Character Recognition using EasyOCR under Image Degradation." By embracing this approach, we aim to gain insights into the performance of EasyOCR in the context of image degradation.

2.3. OCR Quality on Researchers

Historical researchers express hesitance in relying on OCR technology for extracting information from old documents due to accuracy and bias concerns, given the challenges posed by document quality [3]. In a systematic review of OCR titled "Text extraction using OCR," advancements in machine-printed OCR, such as CNN, LSTM, and SRLSA, show promising improvements in detection rates. The focus remains on the Latin script, with continuous development and expansion of OCR into various sectors [4]. A survey on post-OCR processing shows a significant increase in context-dependent approaches, which is crucial in improving OCR development by evaluating output accuracy and performance [5]. These insights underscore the need for further research to enhance OCR accuracy and adaptability in diverse domains and applications. Figure 4 shows the growth of the OCR detection method over the years; throughout the years, context-dependent detection has seen a rise in preference.

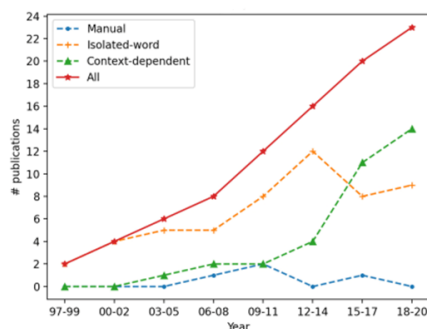


Figure 2.4. Growth of OCR approaches [5]. The growth of OCR detection method over the years; throughout the years, context-dependent detection has seen a rise in preferences.

2.4. Analysis of License Plate Recognition using EasyOCR and TesseractOCR

In research done by D.R.Vedhaviyassh, R.Sudhan, G.Saranya, M.Safa, and D. Arun that made comparisons using EasyOCR and TesseractOCR on License Plate Recognition using Deep Learning Algorithm, it was found the EasyOCR performed better at recognizing correct license plates 95% which is 5% better than TesseractOCR. This supports the claim of EasyOCR being used for smaller chunks of texts [6].

2.5. Robust Lexicon-Free Confidence Prediction for Text Recognition

The rapid advancement of deep learning has propelled Optical Character Recognition (OCR) to new heights. However, the vulnerability of text recognition results to even minor perturbations in input images necessitates a method for assessing result reliability. This paper introduces an innovative approach for measuring confidence in text recognition outcomes. The method consists of two stages: an initial stage employing a Single-Input Multi-Output network (SIMO) to generate multiple recognition candidates, followed by a refined confidence scoring stage that combines conditional probabilities and voting results. The proposed method is effective in both Latin and non-Latin languages, showcasing impressive performance on standard benchmarks [7]. The significance of this work lies in its ability to offer robust confidence measurement for text recognition results, enhancing the reliability of OCR systems in diverse scenarios.

3. Methodology

3.1. Research Tools

Visual Studio Code (VS Code) was selected as the primary integrated development environment (IDE) due to its flexibility and user-friendly interface. Python was chosen as the programming language for its extensive support and large developer community. OpenCV and Pillow were utilized as key libraries for image processing tasks, providing tools for manipulation and enhancement. EasyOCR was the preferred OCR tool, offering support for multiple languages and confidence values for evaluating OCR accuracy. These research tools aimed to leverage the features of VS Code, Python's ecosystem, OpenCV, Pillow, and EasyOCR to enhance coding efficiency and productivity in image processing and OCR applications.

3.2. Method Of Analysis

Character intensity was varied from 0 to 90% of the maximum intensity value (255) to analyze character faintness. Then, incremental application of Gaussian Blur from Intensity 1 to 10 simulated equipment noise, producing a range of blurred images for analysis. Finally, progressive adjustments to the image size, starting from 100% of the original size and reducing it in 10% increments, simulated different camera distances. Figure 5 shows the range of parameter severity from the least affected to the most severe.

3.3. Data Set

The data set generated from a to z all in lowercase is made by using Pillow Library from Python. Each letter will be induced with the parameters and the total generated images will be 780 with each letter having a total of 30.

3.4. Confidence Value

$$\text{custom_mean}(x) = \left(\prod_{i=1}^n x_i \right)^{\frac{2.0}{\sqrt{n}}} \quad (1)$$

In Equation 1 [8], the custom mean calculation involves confidence scores assigned by a machine learning model to its predictions. This approach combines the product of these scores with a scaling factor, where the scaling factor is based on the number of predictions. The resultant value, obtained through this process, yields a confidence score. A higher custom mean signifies greater prediction certainty, while a lower score indicates increased uncertainty.

The prediction of text content from images using a designated machine learning model. It involves processing input images through the model to generate text predictions. These predictions undergo different decoding methods—greedy decoding, beam search, and word beam search—to transform model outputs into human-readable text. For each prediction, a confidence

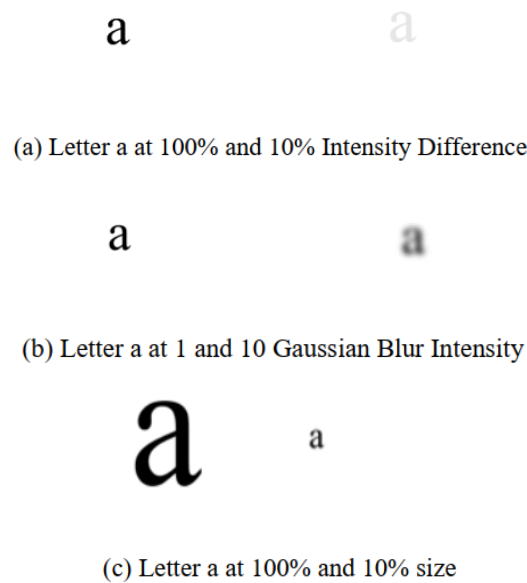


Figure 3.1. Letter a under variation of Intensity Difference, Gaussian Blur, and size.. These variations are to simulate equipment and environment aspects such as the faintness of the letter, noise from equipment and the distance between the object and the camera.

score is computed by analyzing prediction probabilities. This entails using softmax probabilities to identify the most likely characters, modifying probabilities based on defined criteria, and then calculating a confidence score using a specialized formula. The outcomes—text predictions accompanied by their respective confidence scores—are aggregated in an output list.

4. Results and Discussion

4.1. Results

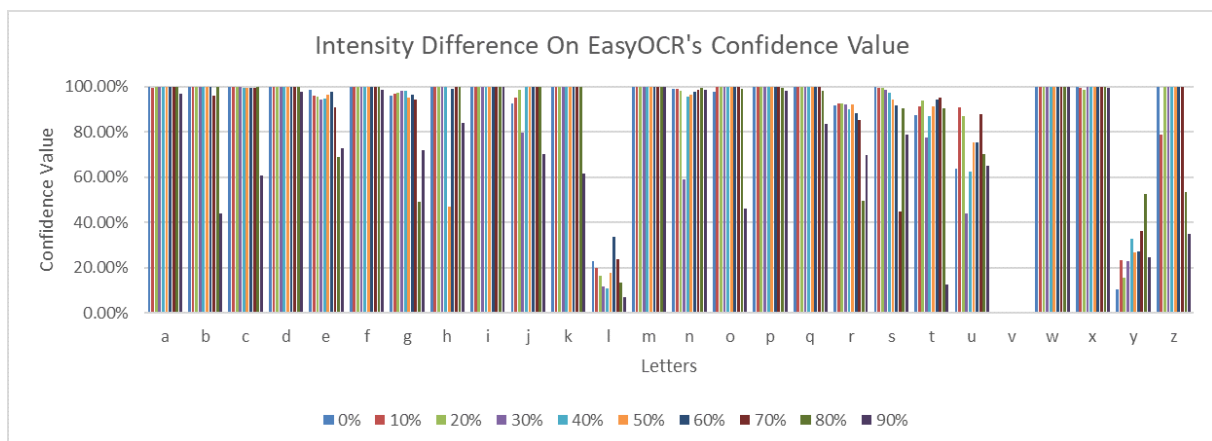


Figure 4.1. (Confidence vs. Background Intensity). Characters remain stable with slight shifts, but confidence drops notably at 80% and 90%, indicating a threshold around 70–80%. Overall, confidence reduces by an average of 4.56%, with 'g' having the least decline at -45.08%.

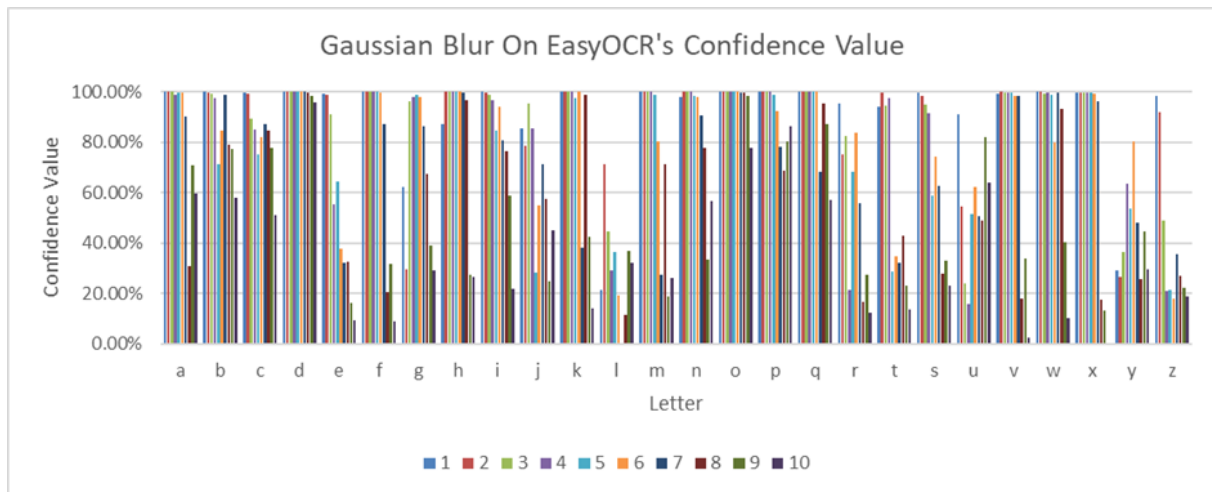


Figure 4.2. (Confidence vs. Gaussian Blur). Clear characters ('a', 'd', 'h', etc.) show consistent high confidence regardless of blur. However, 'e', 'l', and 'r' exhibit reduced confidence due to their intricate shapes.

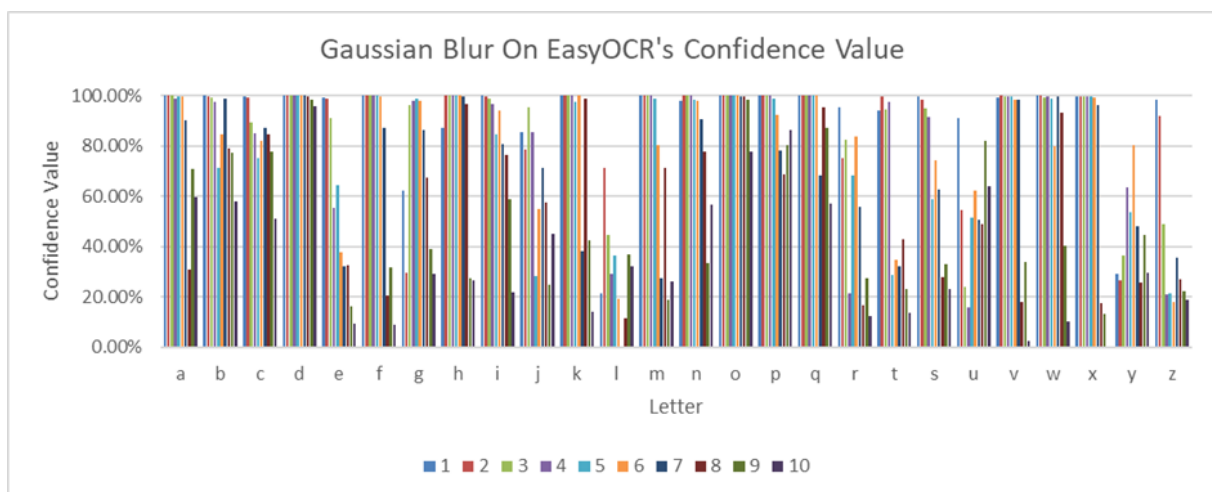


Figure 4.3. (Confidence vs. Character Size). Shrinking characters (100% to 10%) significantly affect scores. At 70% reduction, 'v' and 'x' score 0%, signifying detection failure. This trend highlights EasyOCR's challenge with very small characters.

High confidence scores arise from distinct letter shapes ('a', 'b', 'd', 'h', 'k', 'm', 'p', 'q', 't', 's', 'w', 'x') that allow accurate recognition across degradation levels. A bias toward lowercase outputs for similar uppercase-lowercase letters ('c', 'u', 'w', etc.) offers insights into EasyOCR's decision process.

Low confidence stems from severe degradation, like intense blur or extreme size reduction. These conditions compromise character features, challenging EasyOCR's matching against training data and revealing OCR's limitations under extreme degradation.

4.2. Bias

An intriguing observation emerged during the analysis, revealing the presence of an uppercase bias in EasyOCR. Even with high confidence values, when faced with characters that share identical shapes in both uppercase and lowercase forms, EasyOCR tended to output the uppercase letter, suggesting a preference for uppercase characters. This bias highlights the need for further investigation and potential adjustments in the OCR algorithm to ensure fair and accurate recognition of both uppercase and lowercase characters.

4.3. Discussion

The discussion analyzes EasyOCR's performance on a Latin character dataset, uncovering insights for character recognition. While generally proficient, EasyOCR favoured uppercase interpretations for similar-shaped characters. Unexpectedly, Gaussian Blur sometimes improved accuracy. The study emphasized EasyOCR's sensitivity to character size variations, affecting accuracy. Despite limitations, EasyOCR excelled at distinguishing characters with distinct shapes. Future research could explore alternative OCR models and algorithm optimization for improved accuracy with small characters.

5. Conclusion

The study assessed EasyOCR's letter detection robustness using Confidence Value. It accurately detected distinct letters with high confidence, but struggled with similar uppercase and lowercase letters, resulting in frequent misinterpretation. Notably, a 10% intensity difference led to a 13.17% confidence decrease, except for 'v' which failed detection. Gaussian blur degradation showed an intensity-confidence correlation, except for intensity 6 with a 1.48% increase. Confidence dropped from 88.22% to 35.68% for intensities 9 to 10. Resizing to 20% width caused confidence drops: 70% to 60%, 50% to 40%, 40% to 30%, and 20% to 10%. 'v' remained undetected, except in Gaussian blur maybe due to blurred edges. The study highlights the need for diverse OCR training and suggests exploring new models. Challenges include case distinction. Future research could consider OCRs with word prediction for comprehensive evaluation.

References

- [1] Vijayarani S and Sakila A 2015 *International Journal of UbiComp (IJU)* **6** 19–30
- [2] Peng X, Cao H and Natarajan P 2015 *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (IEEE) pp 771–775
- [3] Traub M C, Van Ossenbruggen J and Hardman L 2015 *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries, TPD L 2015, Poznań, Poland, September 14–18, 2015, Proceedings 19* (Springer) pp 252–263
- [4] Mittal R and Garg A 2020 *2020 second international conference on inventive research in computing applications (ICIRCA)* (IEEE) pp 357–362
- [5] Nguyen T T H, Jatowt A, Coustaty M and Doucet A 2021 *ACM Computing Surveys (CSUR)* **54** 1–37
- [6] Vedhaviyassh D, Sudhan R, Saranya G, Safa M and Arun D 2022 *2022 6th International Conference on Electronics, Communication and Aerospace Technology* (IEEE) pp 966–971
- [7] Song Q, Jiang Q, Zhang R and Wei X 2021 *2020 25th International Conference on Pattern Recognition (ICPR)* (IEEE) pp 3232–3239
- [8] Jaided.AI 2022 Easyocr confidence value computation <https://github.com/JaidedAI/EasyOCR/blob/master/easyocr/recognition.py> accessed: September 17, 2023