OCER and OCWER: Integrating Visual Similarity and Segmentation in OCR Evaluation

Samy Ouzerrout
Université d'Orléans

Abstract

Character Error Rate (CER) and Word Error Rate (WER) are the standard metrics for evaluating OCR, but their binary substitution cost ignores visual similarity between characters and over-penalizes segmentation errors.

We introduce the **Optical Character Error Rate (OCER)**, which weights substitutions by visual similarity, and the **Optical Character Word Error Rate (OCWER)**, which extends this principle to the word level and adds explicit split/union operations. These metrics provide evaluations that better reflect human perception and common OCR-specific errors.

Introduction

Optical Character Recognition (OCR) plays a central role in the digitization of written heritage, educational resources, and multilingual documents. Recent advances in deep learning have significantly improved OCR performance, yet challenges remain, particularly for handwritten documents, historical corpora, and low-resource scripts. Evaluating the quality of OCR outputs is therefore crucial to ensure the reliability of digitized text.

The most common metrics for OCR evaluation are the *Character Error Rate (CER)* and the *Word Error Rate (WER)*. Both rely on the Levenshtein distance, which computes the minimal number of insertions, deletions, and substitutions required to transform a hypothesis into a reference text. While simple and widely adopted, these metrics suffer from two fundamental limitations:

- **Binary substitution cost**: CER and WER treat any substitution as a complete error. Confusing visually similar characters, such as O and Q, is penalized as heavily as confusing A and Z. This binary logic disregards the degree of visual similarity, although it is crucial for OCR evaluation.
- **Segmentation errors**: Traditional CER and WER do not handle errors where a word is incorrectly split ("keyboard" → "key board") or merged ("ice cream" → "ice-cream"). These are double-counted as substitution and insertion/deletion, although they correspond to a single segmentation mistake.

Such limitations are particularly problematic for handwritten text and for languages with complex scripts or diacritics. In these cases, confusions often occur between visually close characters, or segmentation errors are frequent due to variable spacing in handwriting.

To address these issues, we propose two new metrics:

- The **Optical Character Error Rate (OCER)**, which replaces binary substitution costs with weighted costs based on visual similarity between characters.
- The **Optical Character Word Error Rate (OCWER)**, which extends this principle to the word level and integrates split and union operations to better capture segmentation errors.

These contributions aim to provide OCR evaluation metrics that are more aligned with human perception of errors, especially in handwritten and low-resource settings, and to open the way for fairer cross-linguistic evaluation of OCR systems.

This work focuses on the formal design of OCR-oriented evaluation metrics rather than on their empirical benchmarking. We aim to establish a principled framework that corrects known limitations of CER and WER and can later be validated across scripts, OCR engines, and human judgments.

Introducing CER/WER and the Levenshtein Distance

The *Character Error Rate (CER)* and the *Word Error Rate (WER)* are the two main metrics used to evaluate the performance of Optical Character Recognition (OCR) systems. Both compute an error rate based on the Levenshtein distance:

$$ErrorRate = \frac{S + D + I}{N},$$

where:

- S: number of substitutions,
- D: number of deletions,
- I: number of insertions,
- N: number of reference units (characters for CER, words for WER).

This calculation relies on the *Levenshtein distance*, which measures the dissimilarity between two sequences by counting the minimum number of operations required to transform a hypothesis into a reference sequence (Levenshtein 1966). It is computed through dynamic programming by filling an alignment matrix whose optimal path gives the minimal transformation cost between the two sequences.

		t	**a**	**c**
	0	1	2	3
c	1	1	2	2
a	2	2	1	2
t	3	2	2	2

Figure 1: Example of an alignment matrix for the sequences “cat” and “tac”.

Substitution Cost in Levenshtein Distance

In the Levenshtein formulation, insertion and deletion operations have a fixed cost of 1. The substitution cost is defined in a binary manner: it is 0 if the two units are identical and 1 otherwise. This mechanism is known as *binary substitution*.

Formally, each cell of the dynamic programming matrix is computed as:

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 & \text{(deletion)}, \\ D(i,j-1)+1 & \text{(insertion)}, \\ D(i-1,j-1)+ & \\ \qquad \text{sub\_cost}(A[i],B[j]) & \text{(substitution)}. \end{cases}$$

The substitution cost is given by:

$$\text{sub\_cost}(A[i],B[j]) = \begin{cases} 0 & \text{if } A[i] = B[j], \\ 1 & \text{if } A[i] \neq B[j]. \end{cases}$$

This binary formulation treats all substitutions equally, regardless of the degree of visual similarity between the compared units. While effective for general string matching, it constitutes a major limitation for OCR evaluation, where errors often arise from visual ambiguities between characters.

Challenges of CER/WER Based on Levenshtein Distance

The cost calculation method of the Levenshtein distance is effective in general string matching, but when applied to OCR evaluation it exhibits important limitations, as it ignores both visual similarity between characters and OCR-specific segmentation errors.

Weaknesses of Binary Substitution

In CER, substitution costs are defined in a binary manner: two characters are either strictly identical (cost 0) or entirely different (cost 1). This oversimplification fails to reflect the visual similarity that often characterizes OCR confusions.

For example, confusions such as **O–Q**, **l–1**, or **m–n** are common in OCR because of their strong visual resemblance. Intuitively, such errors should be penalized less severely than more distant substitutions like **A–Z** or **L–X**. However, standard CER assigns them the same cost of 1, thereby ignoring the actual degree of visual proximity between characters.

This binary treatment prevents CER from distinguishing between minor and major character errors, which limits its adequacy for evaluating OCR quality, especially in handwritten and historical documents.

Segmentation Errors in OCR

OCR systems, particularly on handwritten or degraded documents, frequently produce segmentation errors. A single character may be split into several parts, or conversely, multiple characters may be merged into one. Whitespace errors leading to merged or split words are among the most frequent error types observed across OCR engines (Reul et al. 2019).

For example, the handwritten word *“chat”* may be incorrectly segmented into *“c hat”*, or the character sequence *“rn”* may be recognized as a single *“m”*. In standard CER/WER, these phenomena are treated as combinations of insertions, deletions, and substitutions, which results in a double counting of what is in fact a single segmentation mistake. Large-scale evaluations confirm the importance of this issue: while most errors correspond to simple character confusions, a non-negligible proportion involve separated or merged terms (Bazzo et al. 2020).

Impact on OCR Evaluation

By treating visually close characters as fully distinct and by over-penalizing segmentation errors, CER and WER tend to inflate the measured error rates of OCR systems, even when the recognized text remains largely readable for a human user. This binary view prevents a nuanced assessment of OCR quality and fails to capture the perceptual severity of errors.

Moreover, standardized OCR evaluation metrics based on CER and WER have been shown to lack robustness, as their values are not always comparable across evaluation tools and experimental setups (Neudecker et al. 2021). This limitation partly explains why quantitative studies on digitized documents are sometimes difficult to interpret or compare.

Overall, these shortcomings motivate the need for OCR-specific evaluation metrics that explicitly account for visual similarity between characters and for segmentation errors, which are central sources of degradation in OCR outputs.

Letter Similarity Exploration

To overcome the limitations of binary substitution in CER, we explored several approaches to quantify **visual similarity between characters**. The intuition is that confusing visually close characters (e.g., O–Q, l–1, m–n) should be penalized less than confusing clearly distinct ones (e.g., A–Z, L–X).

Methodology

Each character was rasterized into grayscale images across multiple fonts, cropped automatically around the glyph using bounding boxes, and resized to a fixed square resolution. Pairs of characters were then compared using different visual descriptors. For each pair, similarity scores were aggregated by taking the median across fonts, reducing typographic bias.

Tested Descriptors

Non-relevant measures. Simple shape descriptors such as Hu Moments, SSIM (Wang et al. 2004), Fourier descriptors, and Zernike moments (Tahmasbi, Saki, and Shokouhi 2011) failed to discriminate effectively: scores were either compressed in a narrow range or saturated at high values, offering little contrast between visually close and distant pairs.

Relevant measures. **HOG + cosine similarity** provided clear separation between close and distant characters, with a ranking that aligned well with human intuition, in line with prior findings on the effectiveness of HOG features for character recognition across multiple scripts (Venkateswarlu, Sudha, and Pavankumar 2022). By contrast, **CNN embeddings**, which have proven highly effective for scene text recognition (Jaderberg et al. 2014), produced consistently high similarity scores, even for unrelated letters, making them less discriminative in practice.

HOG Parameter Exploration

The Histogram of Oriented Gradients (HOG) descriptor, originally introduced for robust human detection (Dalal and Triggs 2005), has since been widely adopted for shape and character recognition tasks (Venkateswarlu, Sudha, and Pavankumar 2022). Different HOG configurations were tested. Larger cells (16 × 16) gave the most stable discrimination. The configuration `out=64, pixels_per_cell=(16,16), cells_per_block=(2,2)` was selected as a practical balance between separation power and computational efficiency.

Generalization Across Alphabets

Preliminary experiments with Cyrillic suggested that HOG retains its discriminative capacity across scripts: visually close letters were consistently scored higher than distant ones, confirming the robustness of this approach beyond Latin.

Synthesis

In summary, Hu, SSIM, Fourier, and Zernike were unsuitable due to lack of contrast. CNN embeddings, while powerful, proved less practical as they inflated similarity values across the board. **HOG with cosine similarity was ultimately chosen** as the most effective descriptor: it offers good discrimination, robustness to font variation, and efficiency, making it the most practical choice for integration into a weighted CER.

OCER: Weighted Character Error Rate

The traditional Character Error Rate (CER) is a normalized Levenshtein distance at the character level with binary substitution costs: identical characters have cost 0, all others cost 1. While effective, this formulation ignores visual similarity between characters, leading to overestimation of errors in OCR evaluation. For example, confusing "O" with "Q" is penalized as harshly as confusing "A" with "Z," despite the former being visually closer.

Weighted Substitution Based on Letter Similarity

To overcome this limitation, we introduce **OCER (Optical Character Error Rate)**, which modifies CER by replacing the binary substitution rule with a weighted cost derived from the visual similarity of characters.

The substitution cost between two characters c_1 and c_2 is defined as:

$$sub\_cost(c_1, c_2) = \begin{cases} 0 & \text{if } c_1 = c_2, \\ dist(c_1, c_2) & \text{if } dist(c_1, c_2) \leq \tau, \\ 1 & \text{otherwise,} \end{cases}$$

where $\tau = 0.5$ and the visual distance $dist(c_1, c_2)$ is derived from the cosine similarity of the HOG descriptors:

$$dist(c_1, c_2) = \frac{1 - sim(c_1, c_2)}{2}.$$

Here, $sim(c_1, c_2) \in [-1, 1]$ denotes the cosine similarity between the HOG representations of characters c_1 and c_2, so that $dist(c_1, c_2) \in [0, 1]$, with 0 indicating identical shapes and 1 maximally dissimilar ones.

This formulation preserves the classical CER behavior for clearly distinct characters while softly penalizing visually plausible confusions, yielding an evaluation that is more consistent with human perception of OCR errors.

OCWER: Extending WER with Weighted Character Distance

While OCER refines character-level evaluation, most OCR evaluation still relies on the *Word Error Rate (WER)*, which measures errors at the word level. However, standard WER suffers from the same binary substitution limitation as CER and additionally fails to handle segmentation errors (splits and merges), which are particularly common in OCR due to irregular spacing, ligatures, or visual artifacts.

We propose **OCWER (Optical Character Word Error Rate)**, an extension of WER that integrates OCER as the substitution cost function and introduces explicit operations for splitting and merging words.

Integration of OCER at the Word Level

Instead of assigning a binary cost to word substitutions, OCWER computes their dissimilarity using OCER. For two words w_1 and w_2:

$$sub\_cost(w_1, w_2) = OCER(w_1, w_2),$$

where $w_1 = ref[i-1]$ and $w_2 = hyp[j-1]$.

This ensures that words differing only by visually close characters (e.g., "ordinateur" → "0rdinateur") receive a much lower penalty than completely different words.

Union and Split Operations

A frequent OCR-specific error is incorrect segmentation:

- **Split error:** a single word is erroneously divided into two (e.g., “keyboard” → “key board”).
- **Union error:** two adjacent words are incorrectly merged (e.g., “ice cream” → “icecream”).

Traditional WER counts such cases as two distinct errors (substitution + insertion/deletion). In contrast, OCWER introduces explicit union and split operations, each with a reduced cost:

$$union\_cost = \frac{1}{|w|}, \quad split\_cost = \frac{1}{|w|},$$

where $|w|$ is the length of the word involved. This formulation mirrors the principle of CER normalization, ensuring that the cost of segmentation errors scales with the word size.

Example. Reference: `keyboard` Hypothesis: `key board`.

Standard WER counts this as one substitution and one insertion (cost $= 2$), although it corresponds to a single segmentation error. OCWER handles it as a single *split* operation with cost:

$$\frac{1}{|\texttt{keyboard}|} = \frac{1}{8} \approx 0.125.$$

Similarly, for a merging error: Reference: `ice cream` Hypothesis: `icecream`, OCWER applies a single *union* operation instead of counting multiple independent errors.

This prevents the double counting of segmentation errors and provides a more faithful assessment of their actual impact on OCR quality.

Recursive Definition of OCWER

The dynamic programming formulation of OCWER extends the Levenshtein distance with additional transitions. Let $w_1 = ref[i-1]$ and $w_2 = hyp[j-1]$. The recurrence is defined as:

$$D(i,j) = \min \begin{cases} D(i-1,j)+1 & \text{(deletion)}, \\ D(i,j-1)+1 & \text{(insertion)}, \\ D(i-1,j-1)+OCER(w_1,w_2) & \text{(weighted substitution)}, \\ D(i-2,j-1)+split\_cost & \text{(split)}, \\ D(i-1,j-2)+union\_cost & \text{(union)}. \end{cases}$$

Finally, OCWER is normalized by the number of words in the reference:

$$OCWER = \frac{D(n,m)}{n},$$

where n and m are the numbers of words in the reference and hypothesis, respectively.

This design allows OCWER to better capture OCR-specific phenomena, balancing word-level similarity with segmentation flexibility.

Computational Complexity

With precomputed character distances, OCER has the same asymptotic complexity as CER, namely $O(nm)$, where n and m are the lengths of the reference and hypothesis strings. The only difference lies in the substitution cost, which involves a constant-time lookup and floating-point operations.

Similarly, OCWER remains $O(nm)$, like standard WER. The introduction of the two additional transitions (split and union) only adds a constant number of operations per dynamic programming cell, resulting in a constant-factor overhead in practice.

Together, OCER and OCWER provide a unified framework for OCR evaluation that accounts for both visual character similarity and segmentation errors.

Conclusion and Future Work

In this work, we introduced two new evaluation metrics for OCR: **OCER**, a character-level metric that integrates visual similarity into substitution costs, and **OCWER**, a word-level extension that combines OCER with explicit handling of split and union errors. This paper focuses on the formal design of these metrics rather than on their empirical benchmarking. Both metrics aim to provide a more perceptually aligned evaluation than traditional CER and WER, by distinguishing minor confusions (e.g., “O” vs. “Q”) from major recognition mistakes, and by treating segmentation errors more appropriately.

Future work will focus on extending this approach to non-Latin alphabets such as Arabic, Cyrillic, or scripts with complex diacritics, where visual similarity and segmentation errors are even more prominent. This generalization would further validate the robustness of the proposed metrics and their suitability for fair OCR evaluation across diverse writing systems.

References

Bazzo, G. T.; Lorentz, G. A.; Vargas, D. S.; and Moreira, V. P. 2020. Assessing the Impact of OCR Errors in Information Retrieval. In *Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020)*, volume 12036 of *Lecture Notes in Computer Science*, 102–109. Springer.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, 886–893. IEEE.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *arXiv preprint arXiv:1406.2227.*

Levenshtein, V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10: 707–710.

Neudecker, C.; Baierer, K.; Gerber, M.; Clausner, C.; Antonacopoulos, A.; and Pletschacher, S. 2021. A survey of OCR evaluation tools and metrics. In *Proceedings of the*

6th International Workshop on Historical Document Imaging and Processing (HIP '21), 13–18. New York, NY, USA: ACM.

Reul, C.; Springmann, U.; Wick, C.; and Puppe, F. 2019. State of the Art Optical Character Recognition of 19th Century Fraktur Scripts using Open Source Engines. In *Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage (DATeCH)*, 99–103. ACM.

Tahmasbi, A.; Saki, F.; and Shokouhi, S. B. 2011. Classification of benign and malignant masses based on Zernike moments. *Computers in Biology and Medicine*, 41(8): 726–735.

Venkateswarlu, K.; Sudha, N.; and Pavankumar, P. 2022. Implementing HOG features to Recognize Multilingual Characters in Machine Learning. *Science Technology and Development*, 11(10): 106–115.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.