electronics

MDPI

Article

Mean-C Thresholding with Noise-Adaptive Offset for OCR of Low-Contrast Receipts: Analysis, Benchmark, and Resource Model for Stream-Based Implementation

Roman Lishchuk [1], Volodymyr Kucheruk [1], Pavlo Kulakov [1], Oleksandr Lishchuk [2], Serhii Baraban [3,*], Andrii Semenov [4] and Andrii Rudyk [5]

[1] Department of Information Technologies, Uman National University, 1 Institutska Str., 20305 Uman, Ukraine; roman.lishchuk@unu.edu.ua (R.L.); volodymyr.kucheruk@unu.edu.ua (V.K.); pavlo.kulakov@unu.edu.ua (P.K.)
[2] Polish-Japanese Academy of Information Technology, 86 Koszykowa Str., 02-008 Warsaw, Poland; s33287@pjwstk.edu.pl
[3] Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 3, 60-965 Poznan, Poland
[4] Faculty of Information Electronic Systems, Vinnytsia National Technical University, Khmelnytske Highway 95, 21021 Vinnytsia, Ukraine; semenov.a.o@vntu.edu.ua
[5] Department of Energy, National University of Water and Environmental Engineering, 11 Soborna Str., 33028 Rivne, Ukraine; a.v.rudyk@nuwm.edu.ua
* Correspondence: serhii.baraban@put.poznan.pl; Tel.: +48-5151-79-906

Abstract

Binarization of fiscal receipts becomes difficult when the thermal printing has faded and lighting is uneven. We investigate the classical local rule Mean$-C$, $T = m - C$, which requires only a single integral representation or one set of line buffers. The authors analyze when its absolute shift outperforms the relative shift of the Bradley–Roth method or the variance-based shift of the Sauvola method. The image formation model defines the permissible band $z\sigma_n \le C < (1-p)\Delta_{\min}$ (noise level σ_n, ink fraction p, ink contrast $\Delta_{\min}$). The model explains why Mean$-C$ thresholding preserves faded print and why it fails in the presence of strong noise. Therefore, the authors determine the shift by a robust noise estimate, $C^* = \max(C_0, \kappa\hat{\sigma}_n)$ (noise-adaptive Mean-C, NA-Mean-C), without changing the rules for an individual pixel. In the benchmark, the parameters for 11 methods were selected using a separate validation set; the results were evaluated across five degradation categories without exceptions. On 150 synthetic receipts, NA-Mean-C achieved the highest average F-measure (91.61%) and the lowest average character error rate (CER) (6.89%), whereas Mean-C with a tuned fixed shift failed to handle high-frequency noise (F-measure 15.89%). The Wolf and Gatos methods remained the best in specific categories. On 80 public Scanned Receipt Optical Character Recognition (OCR) and Information Extraction (SROIE) receipts—across fourconditions—the NA-Mean-C method and other mean-based approaches successfully extracted the total amount from 55.6 to 57.5% of the documents compared to 48.8% for Gatos and 40.3% for Otsu. A bit-accurate model of the streaming implementation shows that, compared to Sauvola, Mean$-C$ saves only about 6% of on-chip memory but does not require a general-purpose multiplier, a divider, or the calculation of a square root. Evaluated independently of binarization and OCR, the keyword-based total amount extractor improved accuracy on the SROIE benchmark text from 42.9% to 83.0%.

Keywords: binarization of document images; local thresholding; Mean$-C$; noise estimation; OCR; receipts; integral image; streaming implementation.

Check for updates

Academic Editor: Dah-Jye Lee

Received: 28 September 2026
Revised: 10 November 2026
Accepted: 18 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Automatic recognition of data from fiscal receipts is a standard component of accounting, auditing, and expense control. Receipts are increasingly being photographed using built-in cameras from mobile devices rather than flatbed scanners, and processing images directly on the mobile device is advantageous in terms of minimizing latency and ensuring privacy [1]. However, mobile photography introduces lighting gradients, shadows cast by the device itself, and creases in the paper, while thermal prints fade over time due to heat and light. As a result, the absolute difference between the ink and the brightness of the paper becomes small [2–4]. In such images, the quality of optical character recognition (OCR) is strongly dependent on the binarization stage, which separates the text from the background [2,5].

Global thresholding methods, such as the Otsu method [6], are fast but cannot account for spatially varying illumination [5]. Local (adaptive) thresholding methods calculate a threshold for each pixel based on the statistics of its neighborhood: the mean with a relative offset (Wellner [7], Bradley–Roth [8]), the mean with an absolute offset (the Mean$-C$ rule, available in OpenCV as ADAPTIVE_THRESH_MEAN_C [9,10]), or the mean and standard deviation (Niblack [11], Sauvola [12], Wolf [13], NICK [14]). When using integral images, the computational cost per pixel for all these methods is independent of the window size [15–17].

The article investigates the absolute-shift rule $T_{x,y} = m_{x,y} - C$ for the OCR of low-contrast receipts. The thresholding form itself is not new: it is the classic Mean$-C$ rule [9,10], and the implementation used here—based on an integral image—produces the same binary image as ADAPTIVE_THRESH_MEAN_C in OpenCV, subject to rounding of the local mean (Section 3.3).

In this case, the application of the Mean$-C$ binarization method lacks the following: a rationale for when an absolute offset is preferable to a relative or variance-based one, along with guidance on how to select the parameter C; a thorough comparison against tuned baseline methods across all types of degradation—specifically those where Mean$-C$ fails; and an accurate assessment of the hardware resources saved compared to variance-based methods.

Accordingly, the main scientific results of this article are as follows:

- Image formation, the results of which indicate that, in the presence of additive sensor noise and illumination that is approximately constant within a given window, the shift must satisfy the condition $z\sigma_n \leq C < (1-p)\Delta_{\min}$ (where σ_n is the noise level, p is the ink fraction, and $\Delta_{\min}$ is the absolute ink contrast)—a fact that explains both the superiority of Mean$-C$ for faded prints and its failure in the presence of significant noise;
- Noise-adaptive shift $C^* = \max(C_0, \kappa\hat{\sigma}_n)$, where $\hat{\sigma}_n$ is obtained using a robust variant of the Immerkær estimator [18]; this approach preserves the single-integral-image structure and adds one global statistic per frame (NA-Mean-C);
- A bit-accurate model of a streaming implementation (operating on line buffers) with precise register bit-widths and memory sizes for Mean$-C$, Bradley–Roth, and Sauvola, as well as a division-free comparison that does not require a general-purpose multiplier;
- Benchmark: 13 configurations of 11 methods with hyperparameters selected via grid search on a separate validation set; five synthetic degradation categories (none excluded); Document Image Binarization Contest (DIBCO)-style pixel-level metrics; Tesseract character error rate (CER); and end-to-end evaluation of 80 public Scanned Receipt OCR and Information Extraction (SROIE) receipts under four conditions [19], with binarization, OCR, and extraction of the total amount evaluated separately.

2. Related Works

The Otsu method selects a global threshold that maximizes the between-class variance:

$$\sigma_b^2(T) = \omega_0(T)\,\omega_1(T)\,[\mu_0(T) - \mu_1(T)]^2, \tag{1}$$

where $\omega_{0,1}$ denotes class probabilities, and $\mu_{0,1}$ denotes class mean values [6]. Its cost is linear with respect to the number of pixels; however, a single threshold cannot separate text from background when a shadow makes the background in one area darker than the ink in another: the histogram ceases to be bimodal, and the shadowed background is classified as text [5].

Local methods calculate $T_{x,y}$ over a $w \times w$ window. The Wellner method compares each pixel with a moving average of the just processed pixels and marks it as ink if it is darker by t percent [7]. Bradley and Roth replaced the moving average with an exact average within a window, calculated from an integral image using the following expression [8]:

$$T_{x,y} = m_{x,y}\left(1 - \frac{a}{100}\right). \tag{2}$$

In both methods (Wellner and Bradley–Roth), the threshold is relative: For a pixel to be considered ink, it must be darker than the local background by a certain fraction; thus, the required contrast increases along with the background brightness. The Mean-C rule uses an absolute offset—a constant value C that is subtracted from the local mean: $T_{x,y} = m_{x,y} - C$. This rule is described in standard image processing references [10] and implemented in OpenCV as ADAPTIVE_THRESH_MEAN_C, and with a Gaussian-weighted mean as ADAPTIVE_THRESH_GAUSSIAN_C [9].

Another group of methods determines the shift based on the local standard deviation $\sigma_{x,y}$—that is, the degree to which the brightness varies within a window. The Niblack method uses a threshold $T = m + k\sigma$ [11]. The Sauvola method uses a threshold defined by the expression

$$T_{x,y} = m_{x,y}\left[1 + k\left(\frac{\sigma_{x,y}}{R} - 1\right)\right], \tag{3}$$

where $R = 128$ for 8-bit images and $k \in [0.1; 0.5]$ [12,20].

In [13], the authors normalize the contrast based on the minimum brightness of the entire image and the maximum local standard deviation. The NICK method lowers the threshold in light areas, resulting in background noise being less likely to be perceived as text [14]. If local sums are computed using integral images of I and I^2, all these methods require the same number of operations per pixel regardless of the window size—that is, $O(1)$ [12,15].

For damaged historical documents, methods have been developed that combine thresholding with background estimation and stroke analysis. The Gatos method first smooths the image using a Wiener filter, performs preliminary segmentation using the Sauvola method, and then estimates the background surface [2]. The Su method creates an image of local contrast (the difference between the maximum and minimum within a window) and applies a threshold to it [21,22]. The Lu method uses background estimation and stroke edges [23]. The Howe method minimizes an energy function based on the Laplacian, with automatic parameter selection [24]. Approaches based on phase features [25] and approaches for combined damage [4] have also been proposed. The methodology for evaluating document binarization—specifically metrics such as the F-measure and DRD (distance reciprocal distortion)—was shaped by the DIBCO competitions [26–28].

Learning-based methods—fully convolutional neural networks [29], selective autoencoders [30], hierarchical deep networks with supervised learning [31], and ensembles

trained on the results of classical methods [32]—achieve the best results in DIBCO. However, they require training data and perform orders of magnitude more computations per pixel than local thresholding methods. That is precisely why simple thresholding methods are still used on devices with limited computational resources [1,33].

Among public datasets for receipts are SROIE (scanned receipts with annotations for text and key fields, including the total amount) [19], CORD [34], and KORIE [35]. The most accurate modern methods for field extraction use transformers that take into account the placement of text on the page, such as LayoutLM [36] and DocFormer [37]. At the same time, in simple pipelines, extraction based on regular expression rules remains common [38]; this is the specific approach examined in this article.

In [39], real-time character recognition based on a field-programmable gate array (FPGA) is implemented for license plate recognition systems, while a methodology for hardware–software co-design of signal processing systems is discussed in [40]. To calculate statistics within a window as pixels arrive in a stream, a set of line buffers (memory storing the last few image rows) is typically used, along with running sums maintained across columns and rows. This eliminates the need to store the integral image of the entire frame. This structure is taken as the basis for the resource model in Section 3.4.

3. Materials and Methods

3.1. Mean-C Threshold and Conditions for the Appropriateness of an Absolute Shift

For a grayscale image I, let us denote the local mean within a $w \times w$ window by $m_{x,y}$; the binary result of the processing is determined by the following expression:

$$B_{x,y} = \begin{cases} 0 \text{ (text)}, & I_{x,y} < m_{x,y} - C, \\ 255 \text{ (background)}, & \text{otherwise.} \end{cases} \tag{4}$$

To determine the necessary shift, we will use an image formation model

$$I_{x,y} = L_{x,y}\,\rho_{x,y} + n_{x,y}, \tag{5}$$

where the variables are defined as follows: L—luminosity; ρ—reflection coefficient (ρ_b for paper, $\rho_t < \rho_b$ for ink); n—sensor noise with zero mean and standard deviation σ_n.

Assume that the value of L is approximately constant within the window (this holds true when the window is small relative to the illumination gradient), and let p be the proportion of ink pixels in the window. If the absolute contrast in ink is $\Delta = L(\rho_b - \rho_t)$, then the local mean is defined as $m \approx L\rho_b - p\Delta$. Taking Equation (4) into account, we obtain the following:

- Ink pixel is detected if $L\rho_t + n < L\rho_b - p\Delta - C$—that is, with a probability of $1 - \Phi\left(\frac{C-(1-p)\Delta}{\sigma_n}\right)$;
- The background pixel is incorrectly labeled as ink with a probability of $\Phi\left(-\frac{C+p\Delta}{\sigma_n}\right) \leq \Phi\left(-\frac{C}{\sigma_n}\right)$.

The following variable is defined: Φ—standard normal distribution function (Gaussian noise is assumed solely for the explicit expression of probabilities). The requirement that the probability of a false alarm not exceed $\Phi(-z)$ and that the faintest ink be detectable in the absence of noise determines an admissible band defined by the inequality.

$$z\,\sigma_n \;\leq\; C \;<\; (1-p)\,\Delta_{\min}. \tag{6}$$

Thus, the shift C must be large enough to ignore background noise but small enough not to cut off faint ink pixels.

Three consequences, which are experimentally verified, follow from inequality (6):

1. **Faded Print.** Fading reduces the ink's contrast Δ—that is, the absolute difference in brightness between the paper and the ink. The Mean$-C$ applies a uniform absolute shift C across all parts of the image, regardless of the background brightness. Consequently, faded ink remains visible as long as the condition $(1-p)\,\Delta > C$ is met.
2. **Relative Shifts.** In the Bradley–Roth method, the shift is not constant but proportional to the local mean: $C_{BR} = \frac{a}{100}\,m \approx \frac{a}{100}\,L\rho_b$. In this case, the condition for the detection of ink is defined by the expression $\frac{\rho_b - \rho_t}{\rho_b} > \frac{a}{100\,(1-p)}$. This is the condition for Weber contrast—that is, for the relative difference—and it does not depend on the illumination L. This approach is valid when the ink contrast varies in proportion to the illumination; however, in light areas, it requires a high absolute contrast value. For example, with $m = 240$ and $a = 15$, the threshold is $T = 204$; therefore, an ink pixel with a brightness of 215 will be lost. For an absolute shift of $C = 12$, the threshold is $T = 228$, and this pixel is retained. In the case of low local contrast ($\sigma_{x,y} \ll R$), the Sauvola method, in accordance with Equation (3), has a threshold of $T \to m\,(1-k)$. This is also a relative shift, in this case, of $100k$ percent. The threshold of the Sauvola method approaches the mean (i.e., the method becomes less strict) only as the local deviation $\sigma_{x,y}$ increases. Therefore, for faded text, Sauvola requires a small value of k (Section 5.2).
3. **Noise.** If the noise level σ_n is large, a fixed small offset C violates the left-hand side of inequality (6), $z\sigma_n \le C$, and the background noise is labeled as ink. If $z\sigma_n \ge (1-p)\,\Delta_{\min}$, then there is no value of C, which satisfies both parts of the inequality, and the method inevitably fails. In this case, a variance-based method, a global method, or prior noise suppression is required.
   If the noise is additive, its level σ_n does not depend on the illumination L. Therefore, it is the absolute, rather than the relative, shift that corresponds to the noise level. If the noise depends on the signal (shot noise), σ_n increases in proportion to $\sqrt{L}$—that is, this case is located between the two previous ones.

3.2. Noise-Adaptive Shift (NA-Mean-C, Noise-Adaptive Mean$-C$)

It follows from inequality (6) that C must be chosen to be proportional to the noise level rather than fixed. We estimate σ_n once per image using a robust form of the Immerkær estimator [18]. The image is filtered by a kernel

$$N = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \tag{7}$$

which suppresses smooth illumination and has $\| N \|_2 = 6$. Thus, for noise with a standard deviation of σ_n, the response has a standard deviation of $6\sigma_n$.Immerkær used the mean absolute response. Since the edges of the text produce large responses, we use the median of the absolute response, which remains robust as long as the edges account for less than half of the pixels:

$$\hat{\sigma}_n = \frac{\operatorname{median}\left|\,(I * N)_{x,y}\right|}{0.6745 \cdot 6}. \tag{8}$$

Then, the shift is defined by the expression

$$C^* = \max(C_0,\ \kappa\,\hat{\sigma}_n), \tag{9}$$

where κ plays the role of z in inequality (6), and the lower bound C_0 accounts for quantization, compression, and paper texture, which the evaluator does not see.

Both parameters were selected in the validation set (Section 4.3). The solution for an individual pixel remains unchanged, so the method still requires only a single integral image (or a set of line buffers). Additional costs comprise one 3×3 filter and one median operation per frame. In a streaming implementation, the median can be selected from a 256-bin histogram of saturated responses; for videos, values from the previous frame can be used, while for individual images, a preliminary pass with subsampling is sufficient.

3.3. Computation Using an Integral Image

The integral image (table of cumulative sums) [16,17] has the form

$$S_{x,y} = \sum_{x' \le x,\ y' \le y} I_{x',y'}, \tag{10}$$

and is calculated in a single pass. The sum within a window with corners (x_1, y_1) and (x_2, y_2) is equal to

$$\Sigma_{x,y} = S_{x_2,y_2} - S_{x_1-1,y_2} - S_{x_2,y_1-1} + S_{x_1-1,y_1-1}, \tag{11}$$

and the local mean is:

$$m_{x,y} = \frac{\Sigma_{x,y}}{A_{x,y}}, \qquad A_{x,y} = (x_2 - x_1 + 1)(y_2 - y_1 + 1). \tag{12}$$

Multiplying Equation (4) by $A_{x,y} > 0$ forms an equivalent check without division.

$$I_{x,y}\, A_{x,y} + C\, A_{x,y} < \Sigma_{x,y}. \tag{13}$$

For replication padding at the image boundary, $A_{x,y} = w^2$ is a constant; thus, $I \cdot A$ and $C \cdot A$ amount to multiplication by constants (shift-and-add networks), and neither a divider nor a general-purpose multiplier is required. If the window is clipped at the boundary, $A_{x,y}$ changes, and a multiplier with a bit-width of $8 \times \lceil \log_2 w^2 \rceil$ bits is required. The algorithm for the corresponding procedure is shown in Figure 1. The cost per pixel $O(1)$ relative to w is a property of every local thresholding method that uses an integral image, specifically including the Bradley–Roth and Sauvola methods [15].

OpenCV calculates m using a normalized block filter (with border replication), rounds it to an integer, and marks a pixel as background if $I > m - C$ [9]. Therefore, an implementation based on an integral image—using the exact window area at the boundaries and floating-point averages—differs from OpenCV only for pixels located within the rounding error margin of the threshold. On 150 synthetic test images, the two binary results differ in 0.20% of the pixels, while the integer-based form without division (13) matches the floating-point form bit for bit. Therefore, moving forward, we refer to the fixed-shift method as Mean-C. The results are presented for two configurations: a fixed setting ($w = 45$, $C = 12$, implementation based on an integral image) and an OpenCV implementation with parameters tuned on a validation set.

3.4. Analytical Model of Streaming Implementation Resources

Two architectures are distinguished: an integral image of the entire frame, typical for central processing unit (CPU)-based implementations, and a w line buffer, typical for streaming (FPGA/application-specific integrated circuit (ASIC)) implementations. Estimating memory as $w \cdot W$ words of a fixed bit-width conflates these architectures and overestimates the cost of squared-brightness statistics. The bit-widths below have been calculated for both architectures and verified using a bit-accurate software model.

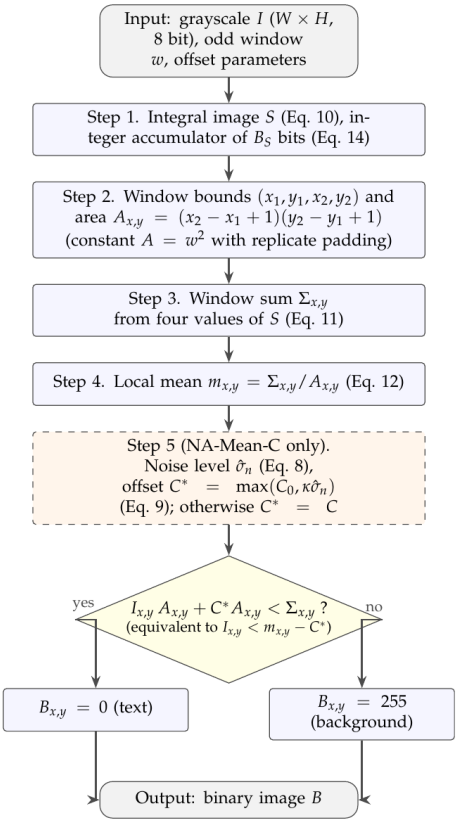


Figure 1. Block diagram of Mean$-C$ binarization with an optional noise-adaptive offset (step 5). The solution is presented in a form without division, as shown in Equation (13).

Integral Image of the Entire Frame (CPU Implementation). The largest element S is equal to $255\,WH$. Therefore,

$$B_S = \lceil \log_2(255\,WH + 1) \rceil, \qquad B_{S^2} = \left\lceil \log_2\left(255^2\,WH + 1\right) \right\rceil. \tag{14}$$

For $W = 1000$ and $H = 300$, $B_S = 27$ and $B_{S^2} = 35$ bits (not 64). Given specific bit-depths, Sauvola requires $(27 + 35)/27 = 2.30$ times more memory than Mean$-C$; with CPU word sizes (32 and 64 bits), the ratio is 3.0. For the receipt image with dimensions of 1000×2000, the bit depths are 29 and 37 bits. A 32-bit accumulator for S is enough only for $WH < 2^{32}/255 \approx 16.8$ megapixels (unsigned type).

Streaming Implementation (FPGA or Sequential Code for a CPU). The table is not needed for the entire frame. The pipeline stores w row buffers of 8-bit pixels, an array of column sums c_x for the last w rows, and a sliding horizontal sum of w column sums. Register bit widths are defined by expressions

$$B_c = \lceil \log_2(255\,w + 1) \rceil, \qquad B_\Sigma = \left\lceil \log_2\left(255\,w^2 + 1\right) \right\rceil. \tag{15}$$

The Sauvola method additionally stores column sums of squares with $B_{c^2} = \lceil \log_2(255^2 w + 1) \rceil$ bits and window sums of squares with $B_{\Sigma^2} = \lceil \log_2(255^2 w^2 + 1) \rceil$ bits. For $w = 45$, $B_c = 14$, $B_\Sigma = 19$, $B_{c^2} = 22$, and $B_{\Sigma^2} = 27$. Memory size is determined by expressions

$$V_{\text{Mean-C}} = V_{\text{BR}} = 8wW + B_c W, \qquad V_{\text{Sauvola}} = 8wW + (B_c + B_{c^2})W. \tag{16}$$

The numerical values are presented in Table 1. Line buffers dominate in streaming architecture, and they are the same for all three methods. Therefore, the memory savings of Mean$-C$ compared to Sauvola amount to approximately 6%. The significant difference lies

in the means required to perform arithmetic operations: Mean-C and Bradley–Roth require only adders and multiplication by constants, whereas Sauvola requires a squaring unit, a general multiplier for Σ^2 operation, and a square root operation unit (or additional general multipliers, if the inequality needs to be squared). The bit-accurate model was run on an all-white frame (the worst-case scenario) and on test images; all registers remained within the bit-width limits defined by Equation (15), and the results matched the floating-point reference. Analytical estimates are provided in this case; post-synthesis data regarding look-up tables (LUTs), block random access memory (BRAM), and digital signal processing (DSP) blocks fall outside the scope of this article.

Table 1. Exact register bit-widths, memory, and arithmetic per pixel (W = 1000, w = 45), memory 10^3 bits.

Value	Mean−*C*	Bradley–Roth	Sauvola
String buffers ($8wW$)	360.0	360.0	360.0
Column sums	14.0 (14 bits)	14.0 (14 bits)	14.0 + 22.0 (14 + 22 bits)
Window sum registers	19 bits	19 bits	19 + 27 bits
Total memory of the stream implementation	374.0	374.0	396.0 (+5.9%)
Integral image of the frame, $H = 300$, exact bit-widths	8135	8135	18,681 ($\times$2.30)
Integral image of the frame, $H = 300$, words 32/64 bits	9642	9642	28,925 ($\times$3.00)
Additions/subtractions per pixel (sums)	4	4	8
Decision	Multiplication by a constant + comparison	2 multiplications by a constant + comparison	Squaring, general multiplier, $\sqrt{\cdot}$, multiplication
General-purpose multipliers, dividers, and square-root units	0/0/0	0/0/0	$\geq$ 2/0–1/1

3.5. Determination of the Total Amount

To determine whether binarization preserves information, the output of Tesseract is passed to a total-amount extractor. It is evaluated separately from binarization and OCR (Section 5.6). We use rule-based extraction—a method that employs regular expressions and simple statistics, without involving natural language processing.

Frequency Rule (P-freq). All lexemes matching `\b(\d+[.,]\d{2})\b` are collected, values below an empirical threshold of 10.0 are discarded, and the most frequent value is returned. This rule has three drawbacks, which are quantified through experiments. First, the word boundary `\b` does not work between a currency prefix and a digit (for example, RM9.00), because both are word characters. Secondly, the 10.0 threshold discards any total sum less than 10; this applies to 28% of SROIE receipts. Third, the mode is incorrect when the total sum is printed once, while another sum is repeated—for example, in the line items, the cash total, or the tax total. Furthermore, the rule explicitly does not account for discounts, service fees, and digit merge (99.00→9900).

Keyword Rule (P-kw). The parser searches for the total amount near the words typically used to label it on a receipt. Its operating algorithm is as follows. First, the parser searches for lines that contain a total-amount label. The parser selects lines that include one

of the keywords indicating a total amount (e.g., TOTAL, AMOUNT, PAYABLE, etc.). OCR often distorts labels (for example, recognizing "TOTL" instead of "TOTAL"). Therefore, a word from a line is accepted as a label if it differs from the target label by no more than one change—specifically, the substitution, insertion, or deletion of a single character (i.e., a Levenshtein distance of no more than 1). Before comparison, common confusions between digits and letters—such as 0/O, 1/I, and 5/S—are corrected. Subsequently, similar strings are filtered out. Strings with labels that merely resemble a "total" but actually represent a different value—such as SUBTOTAL, CHANGE, CASH, DISCOUNT, EXCL., TOTAL GST, etc.—are discarded. Next, an amount is selected from the line. The rightmost amount is taken from each selected line. Typically, amounts on a receipt appear in the right-hand column. If there is no amount in a line, it is sought in the following line. The response is then selected. If at least one row contains a priority label (PAYABLE, GRAND, NETT, ROUNDED, INCL.), the response is selected only from the amounts in such rows. Among the candidates, the value that occurs the most frequently is returned; if the frequencies are equal, the larger value is returned. Fallback Option: If no tag is found, the value that appears most frequently throughout the entire text of the receipt is returned. There is no minimum threshold value.

Amounts are recognized by a pattern that allows for thousands of separators and a currency symbol preceding the number (e.g., RM9.00). The parser rules were developed on the basis of only 100 individual SROIE receipts, none of which are used in any of the presented results. For comparison, the simplest maximum sum rule (P-max)—which returns the largest sum found in the text—has also been included.

4. Experiment

4.1. Synthetic Receipts with Reference Labeling

For real-world photographs, there is no pixel-level ground truth—that is, it is unknown which specific pixels belong to the text. Therefore, a synthetic receipt generator has been created, for which three things are known: the pixel mask of the text, the full text, and the total amount. Each generated receipt contains the store name, phone number, date and time, receipt number, 4 to 10 line items (some including item quantities), subtotal, value-added tax (VAT), and the total amount, as well as payment information—indicating either a cash payment with change or a card payment. With a probability of 0.35, a discount line is added to the receipt, and with a probability of 0.25, a service charge line is added. The decimal separator (a dot or a comma) is chosen randomly. The text is written in one of five monospaced fonts with a size of 24–30 pixels on an image 1000 pixels wide, and the height of the receipt ranges from 636 to 1348 pixels. The reference mask is an image of text prior to any degradation in quality.

Five categories of quality degradation were generated, with 40 images in each (10 for validation, 30 for testing):

C1—Ideal conditions: clear image without alterations;

C2—Uneven Illumination: The image is multiplied by a smooth illumination field. The field consists of a linear gradient, a circular shadow, and—with a probability of 0.6—a dark crease band; the darkest area has a brightness of 25–50% of the full value ($L_{\min} \in [0.25; 0.5]$), and Gaussian noise with $\sigma = 2$ is added;

C3—Faint Text: Ink brightness is 150–210, and paper brightness is $255 \cdot [0.85; 0.95]$;

C4—High-Frequency Noise: A random value drawn from a uniform distribution $U(0, A)$, where $A \in [80, 150]$, is subtracted from each pixel. This simulates paper texture, grain, and sensor noise under low-light conditions;

C5—Combined: Pale text (ink 150–200) with slight blurring ($\sigma = 0.7$), an illumination field with $L_{\min} \in [0.45; 0.7]$, and Gaussian noise with $\sigma \in [3; 8]$.

4.2. Real Receipts

Five receipts (in Czech), photographed using a smartphone, are taken from the ExpressExpense open collection [41]. Each photo is processed in two versions: the full frame, scaled to a width of $W = 1000$, and the receipt area, manually cropped using a rectangular frame (replacing the document detection stage) and scaled to $W = 1000$. The total amount printed was read from each image manually. The method parameters are taken from Table 2 without adjustment. Due to the small number, these images are used solely for illustrative purposes (Section 5.7), and no statistical conclusions were drawn.

Table 2. Hyperparameters selected via grid search on the validation set (criterion: mean pixel-level F-measure across all five categories (last column)). FM—F-measure.

Method	Selected Parameters	Validation FM (%)
Otsu	–	55.13
Niblack	$w = 95$, $k = -1.0$	65.57
Sauvola	$w = 35$, $k = 0.1$	74.49
Wolf	$w = 15$, $k = 0.5$	87.18
NICK	$w = 45$, $k = -0.1$	74.62
Su	$w = 9$, $N_{\min} = 9$	81.94
Gatos	$w = 45$, $k = 0.1$, glyph = 40	90.57
Bradley–Roth (tuned)	$w = 25$, $a = 8$	75.66
Gaussian$-C$ (OpenCV, tuned)	$w = 65$, $C = 15$	77.00
Mean$-C$ (OpenCV, tuned)	$w = 35$, $C = 15$	76.91
NA-Mean-C (tuned)	$w = 25$, $C_0 = 12$, $\kappa = 2.5$	93.48

The public SROIE (International Conference on Document Analysis and Recognition (ICDAR) 2019) dataset [19] was used: Scanned receipts with word-level transcriptions and annotations for key fields, including the total amount. From the SROIE dataset, 625 receipts were selected that include a total amount in their annotations. These were randomly divided into two parts; 100 receipts were used to refine the P-kw parser rules (keywords and OCR error handling). Of the remaining 525 receipts, 80 were selected for experiments with binarization and OCR. All 525 receipts were used to test the parsers against ground-truth text (Section 5.6). Each image is scaled to $W = 1000$ and evaluated under four conditions: R0—original scanned image; R1—field of illumination ($L_{\min} \in [0.3; 0.5]$) with noise $\sigma = 2$; R2—fading (ink contrast—multiplied by a factor in the range of 0.25 to 0.45; paper—multiplied by a factor in the range of 0.85 to 0.95); R3—fade combined with lightening ($L_{\min} \in [0.5; 0.7]$) and noise $\sigma \in [3; 6]$. The reference text for CER is a reference transcription in which word bounding boxes are grouped into visual lines. The reference total amount is the SROIE label. To limit OCR time, a subset of methods was evaluated in the SROIE dataset: Otsu, Sauvola, Wolf, Su, Gatos, Bradley–Roth tuned, Mean$-C$ (from OpenCV), Mean$-C$ with fixed settings ($w = 45$, $C = 12$), and NA-Mean$-C$, as well as Tesseract's internal binarization. All of them use parameters selected on a synthetic validation set, without re-tuning on SROIE.

4.3. Basic Methods and Protocol for Hyperparameter Tuning

The following methods are compared: Otsu [6], Niblack [11], Sauvola [12,20], Wolf [13], NICK [14], Su [21], Gatos [2], Bradley–Roth [8], Mean$-C$, and Gaussian$-C$ from OpenCV [9], an implementation based on integral-images of Mean$-C$ with fixed settings ($w = 45$, $C = 12$), and NA-Mean$-C$. For the Niblack, Wolf, NICK, Su, and Gatos methods, reference C++ implementations of the open-source Doxa library [42] were used. The Otsu, Mean$-C$ and Gaussian$-C$ methods utilize OpenCV 5.0.0 (Section 4.5). Bradley–Roth, Sauvola, and Mean$-C$ (with fixed settings), and NA-Mean-C methods use

our NumPy implementations. As an additional benchmark, Tesseract is also run directly on the grayscale image (in which case it applies its own Otsu binarization).

The hyperparameters for each method were selected via an exhaustive grid search on 50 validation images, based on the criterion of maximizing the mean pixel-wise F-measure across all five categories. For each method, a single global setting is used, corresponding to the deployment of one configuration. For the Gatos method, the grid search covered windows {25, 45, 65}, $k \in \{0.1, 0.2, 0.3\}$, and a glyph size of {20, 40, 60} pixels. For the Su method, the window is automatically determined based on an estimate of the stroke thickness or selected from the set {5, 9, 15, 21}; the minimum number of contrast pixels, $N_{\min}$, is selected from the set {3, 5, 9, 15} and does not exceed the window size. For the NA-Mean-C method, the grid covered the same windows w as for the other methods, with $C_0 \in \{4, 6, 8, 10, 12\}$ and $\kappa \in \{1.5, 2, 2.5, 3, 4\}$. In total, 480 configurations were tested. The selected values are presented in Table 2.

For comparison, two configurations are also presented, for which their parameters were set in advance and not tuned on the validation set. The first configuration is Mean-C with fixed settings of $w = 45$ and $C = 12$, and the second configuration is Bradley–Roth with $a = 15$—the value recommended by the authors of the method. The window used for it is the same as for Mean-C ($w = 45$), although the authors recommend a window width equal to $1/8$ of the image width. These two configurations show how much adjustment of the parameters changes the result. No parameters were tuned on the test data.

4.4. Metrics

The quality of the pixels on the synthetic data was measured using DIBCO metrics [26,27]: F-measure (FM), pseudo-F-measure (pFM), peak signal-to-noise ratio (PSNR), and DRD [28]. They were calculated using an implementation of Doxa that replicates the official DIBCO evaluation tool [42]. The quality of the OCR is determined based on the proportion of errors between the Tesseract 5 output (long short-term memory (LSTM) engine, page segmentation mode 4, English model) [43] and the reference text, following the compression of spaces, according to the expression

$$CER = \frac{S + D + I}{N} \cdot 100\%, \tag{17}$$

where the following variables are defined: S, D, and I—Levenshtein alignment substitutions, deletions and insertions; N—length of the reference.

CER can exceed 100% when OCR produces many extra characters. End-to-end quality is defined as the proportion of documents in which the extracted total amount matches the reference value exactly. Then, 95% Wilson confidence intervals are presented for the proportions. For pairwise comparisons of two methods on the same documents, McNemar's exact test (for sums) and the Wilcoxon signed-rank test (for CER) were used.

4.5. Platforms and Time-Measurement Protocol

Timing was measured on a single core of a cloud-based machine equipped with an Intel Xeon processor (2.8 GHz) running Ubuntu 24.04 with Python 3.13.17, NumPy 2.5.3, OpenCV 5.0.0 and Tesseract 5.3.4. The C++ code was compiled using GCC 13.3 with the -O3 flag. Each value is the mean ± standard deviation over 30 repetitions, following three warm-up runs. All libraries are limited to a single thread. Two input images were used: a 1000×300 fragment of a synthetic receipt of category C2 (a short receipt or a strip from a long one) and an SROIE receipt, scaled to 1000×1500. The durations of binarization (by module), OCR, and amount extraction were measured separately; the input and output of the image were not included in the measured time.

For the NA-Mean-C, NICK, Su, and Gatos methods, the time was measured using the parameters presented in Table 2. For the remaining window-based measurement methods, the process was carried out using a uniform window size of $w = 45$ (Sauvola $k = 0.2$; Niblack $k = -0.2$; Wolf $k = 0.2$; Bradley–Roth $a = 15$; Mean-C and Gaussian-C $C = 12$). For implementations based on an integral image or a block filter, this does not affect the result; their cost per pixel is independent of w (Section 3.3), and the parameters k, a, and C do not alter the computational load. An exception is the Gaussian-C method, for which the cost of Gaussian smoothing increases with the window size. Therefore, for the setting $w = 65$, its time is longer than that presented in Sections 5.4 and 5.8.

For the comparison of accuracy and execution time (Section 5.4), each method is represented by its fastest measured implementation: Otsu, Mean-C, and Gaussian-C—OpenCV; Niblack, Sauvola, Wolf, NICK, Su, and Gatos—Doxa C++; Bradley–Roth, a streaming C++ implementation; NA-Mean-C is an OpenCV block filter with a noise estimator. The accuracy of Sauvola, Bradley–Roth, Mean-C with fixed settings, and NA-Mean-C was obtained using NumPy implementations. For Mean-C and NA-Mean-C, both implementations provide the same result up to the rounding of the local mean (Section 3.3). For the Sauvola and Bradley–Roth methods, the bitwise equivalence of the implementations was not verified.

In addition to the reference NumPy implementation, a single-threaded C++ implementation of Mean-C, Bradley–Roth, and Sauvola is provided (Section 3.4). It contains no platform-dependent instructions and compiles for both x86 and ARM (Raspberry Pi 5).

5. Results

5.1. Selected Hyperparameters

Table 2 presents the parameters selected for each method in the validation set. For Mean-C with a fixed shift, $C = 15$ and $w = 35$ were chosen; that is, the shift is close to the fixed setting ($C = 12$), while the window is smaller than in the fixed setting ($w = 45$). For NA-Mean-C, $\kappa = 2.5$ is selected. The lower bound $C_0 = 12$ coincides with the fixed-setting offset.

For the test images, the noise estimate calculated using Equation (8) is close to the noise level used to generate the data: $\hat{\sigma}_n = 0$ for C1 and C3 (no noise), 2.0–2.2 for C2 ($\sigma = 2$), 24.5–42.7 for C4 (uniform noise, $\sigma = A/\sqrt{12} = 23$–$43$), and 3.2–7.9 for C5 ($\sigma = 3$–$8$). Consequently, the adaptive shift C^* is equal to 12 in C1–C3, 61–107 in C4 and 12–20 in C5.

5.2. Synthetic Receipts: Five Categories

Table 3 presents the average values for all 150 test images. Tables 4 and 5 present the F-measure and the CER by category.

The results obtained confirm the following:

1. **The Mean-C with fixed shift is, on average, not better than other methods,** and in all categories, the fixed setting ($w = 45$, $C = 12$) provides FM of 71.97% and CER of 29.97%, while the OpenCV-based adaptive Mean-C method provides FM of 74.40% and CER of 29.04%. Wolf (FM 89.42%, CER 12.14%) and Gatos (FM 89.28%, CER 10.55%) perform noticeably better. Otsu has FM of 57.01% and CER of 37.38%: it does not perform well in C2, C4, and C5, but in C1 and C3, it ranks among the most accurate methods. Mean values excluding category C4 would overstate the quality of Mean−C.
2. **The absolute shift preserves the faded print (C3).** All Mean−C variants achieve 95.9–96.8% FM in C3. The Bradley–Roth method without tuning ($a = 15$) achieves only 48.21%, the tuned Bradley–Roth method ($a = 8$) achieves 89.35%, and the Sauvola method tuned achieves 84.87%. These results are consistent with the hypothesis in Section 3.1: A relative shift of a or $100k$ percent removes ink for which the Weber

contrast is less than this value. The Otsu in C3 is just as accurate (FM 98.99%) because the illumination in this case is uniform. The advantage of local methods is evident only under non-uniform illumination (C2 and C5).

3. **The fixed shift does not work in the presence of strong noise (C4), as predicted by inequality (6).** All methods based on local average with fixed or relative shift have, in this case, FM 15–24% and CER 93–100%, because the shift is much smaller than the noise level ($C \ll z\sigma_n$). Similarly, the Sauvola and NICK methods do not function. With noise-adaptive shifting, the same rule achieves 96.38% FM and a 2.66% CER (C4). This is better than in Gatos (FM 86.71%), which uses a Wiener filter for noise suppression.

Table 3. Synthetic test set, average across all five categories (150 images). FM—F-measure; pFM—pseudo-F-measure; PSNR—peak signal-to-noise ratio (dB); DRD—distance reciprocal distortion (DIBCO pixel-based metrics); CER—Tesseract character error rate; "Sum"—percentage of receipts for which the total amount was correctly extracted by the parser P-kw. ↑/↓: higher/lower is better.

Method	FM ↑	pFM ↑	PSNR ↑	DRD ↓	CER (%) ↓	Sum (%) ↑
Otsu	57.01	56.54	18.54	104.17	37.38	56.0
Niblack	64.93	64.37	17.83	29.19	24.10	50.7
Sauvola	71.73	72.51	19.24	38.65	29.75	68.0
Wolf	89.42	88.63	24.13	4.77	12.14	81.3
NICK	71.78	72.77	19.50	32.46	29.91	67.3
Su	79.02	76.70	19.91	38.05	36.66	50.7
Gatos	89.28	90.65	27.89	1.83	10.55	84.0
Bradley–Roth ($a = 15$, not tuned)	61.74	63.41	19.03	24.76	41.97	54.0
Bradley–Roth (tuned)	72.98	73.81	19.81	37.52	30.72	65.3
Gaussian$-C$ (OpenCV, tuned)	74.36	74.64	19.97	37.99	28.99	70.7
Mean$-C$ (OpenCV, tuned)	74.40	74.41	19.82	37.82	29.04	70.0
Mean$-C$ (fixed, $w = 45$, $C = 12$)	71.97	70.93	18.90	41.53	29.97	64.7
NA-Mean-C (tuned)	91.61	92.21	24.79	1.70	6.89	92.7
Tesseract internal binarization (grayscale image)	–	–	–	–	37.49	56.0

Table 4. Pixel-level F-measure (%) by category on the synthetic test set (30 images per category). The best value in each column is highlighted in bold.

Method	C1 Ideal Conditions	C2 Uneven Illumination	C3 Faint Text	C4 High-Frequency Noise	C5 Combined	All
Otsu	99.10	32.44	**98.99**	43.49	11.05	57.01
Niblack	97.13	63.64	96.89	31.40	35.60	64.93
Sauvola	92.88	92.09	84.87	15.61	73.18	71.73
Wolf	96.42	97.02	96.01	74.87	**82.78**	89.42
NICK	93.51	91.43	82.78	18.30	72.89	71.78
Su	93.95	94.77	91.29	81.17	33.92	79.02
Gatos	**99.56**	**98.34**	86.86	86.71	74.95	89.28
Bradley–Roth ($a = 15$, not tuned)	94.38	93.74	48.21	24.49	47.88	61.74
Bradley–Roth (tuned)	94.94	94.38	89.35	16.11	70.10	72.98
Gaussian$-C$ (OpenCV, tuned)	94.16	94.13	95.91	15.83	71.79	74.36
Mean$-C$ (OpenCV, tuned)	93.62	93.53	96.28	15.89	72.69	74.40
Mean$-C$ (fixed, $w = 45$, $C = 12$)	92.37	89.62	96.74	15.15	65.95	71.97
NA-Mean-C (tuned)	94.58	94.49	96.77	**96.38**	75.82	**91.61**

Table 5. Tesseract CER (%) by category on the synthetic test set. The CER for raw output using Tesseract's built-in binarization (last row, C1) represents the lower bound established by the OCR engine itself. The best value in each column is highlighted in bold.

Method	C1 Ideal Conditions	C2 Uneven Illumination	C3 Faint Text	C4 High-Frequency Noise	C5 Combined	All
Otsu	1.72	43.02	0.86	63.89	77.39	37.38
Niblack	1.76	31.63	0.89	35.12	51.10	24.10
Sauvola	2.34	2.58	12.92	99.65	31.25	29.75
Wolf	2.19	1.63	0.97	42.27	**13.65**	12.14
NICK	2.23	4.38	15.10	98.35	29.49	29.91
Su	8.37	8.19	9.73	84.26	72.73	36.66
Gatos	1.78	**1.60**	10.47	14.49	24.42	10.55
Bradley–Roth ($a = 15$, not tuned)	2.33	2.40	50.40	93.05	61.68	41.97
Bradley–Roth (tuned)	2.04	1.74	11.11	99.92	38.79	30.72
Gaussian$-C$ (OpenCV, tuned)	2.13	1.67	1.66	100.00	39.52	28.99
Mean$-C$ (OpenCV, tuned)	2.23	1.75	1.03	99.71	40.48	29.04
Mean$-C$ (fixed, $w = 45$, $C = 12$)	2.45	4.34	**0.70**	99.90	42.48	29.97
NA-Mean-C (tuned)	2.15	1.73	0.74	**2.66**	27.17	**6.89**
Tesseract internal binarization (grayscale image)	**1.52**	43.03	1.95	64.05	76.92	37.49

4. **Clean Images and Uneven Illumination (C1, C2).** Gatos provides the highest pixel-level accuracy (as does Otsu in C1). Methods based on the mean underperform by 4–9 FM points, because the mean within a window classifies the blurred edges of strokes as background. This has almost no effect on the CER: for C1, the CER is 2.15% for NA-Mean-C, 1.78% for Gatos, and 1.52% for Tesseract's own binarization. The last value is the lower bound of the CER for this OCR in such documents.
5. **Combined Degradation.** Combined degradation (C5) is the most complex category. The best result in this category is achieved by the Wolf method (FM 82.78%, CER 13.65%), followed by the NA-Mean-C (FM 75.82%, CER 27.17%) and Gatos (FM 74.95%) methods. The fixed setting underperforms the NA-Mean-C method by 9.9 FM points, primarily along the dark fold, where the noise is strong relative to local contrast.

Overall, NA-Mean-C achieves the highest FM (91.61%) and pFM (92.21%), the lowest DRD (1.70) and CER (6.89%), and the highest proportion of correctly extracted sums (92.7%, inflated for all methods on the synthetic dataset; see Section 6.3). The difference in CER between NA-Mean-C and Gatos or Wolf at the level of individual images is statistically insignificant (Wilcoxon test, $p = 0.48$ and $p = 1.0$), while the difference in the proportion of correctly extracted sums is significant (McNemar's test, $p < 0.001$ for both). If C4 is excluded, NA-Mean-C (average FM 90.4%, CER 8.0%) is comparable to Gatos (89.9%, 9.6%) and underperforms Wolf (93.1%, 4.6%).

Thus, the main advantage of the NA-Mean-C method is not that it is more accurate than the Wolf and Gatos methods, which account for local brightness variation. Rather, NA-Mean-C eliminates the primary drawback of Mean-C—failure in the presence of strong noise—while retaining the same simple thresholding rule. As a result, processing time increases: The NA-Mean-C method takes 29.0 ms to process a 1000×1500 frame, whereas the Mean-C method from OpenCV takes 4.0 ms. However, even 29.0 ms represents only about 2% of the time Tesseract spends recognizing the same image (Section 5.8).

5.3. Sensitivity to C, w, and κ

Figure 2 shows CER as a function of the offset C (Mean$-C$ with a fixed offset, $w = 35$), w ($C = 15$), and κ (NA-Mean-C) for 10 test images from each category. The curves represent both inequalities of Equation (6). In category C2, the noise level is $\sigma_n = 2$. If the shift is

smaller than the noise level, the background noise is treated as ink: The CER is 67.9% for $C = 2$ and 47.6% for $C = 4$. For $C \geq 6$, the CER does not exceed 4.9%. Thus, the shift must be at least $2.5\sigma_n = 5$, in accordance with the left-hand side of the inequality (6). For C3 (pale ink, no noise), the CER does not exceed 1.4% for all $C \leq 15$, and it increases to 8.6%, 48.5% and 100.0% at $C = 20$, 30, and 60 as the shift approaches the ink contrast. Strong noise and pale ink need the opposite. The CER in C4 decreases noticeably only at a very large shift (29.2% at $C = 60$), but at such a shift, the pale ink in C3 disappears: At $C = 30$, the CER in C3 increases to 48.5%, and at $C = 60$, it increases to 100%. In the combined category C5, the best result is achieved at an intermediate value (CER 43.4% at $C = 12$). On average in all categories, a fixed shift forms the lowest CER at $C = 10$–15 (29.7% at $C = 12$); therefore, $C = 12$ is a reasonable compromise. However, no single fixed value of C is suitable for both C3 and C4, and this is precisely what justifies an adaptive shift. For a window size w, there is a similar compromise. A small window affects the recognition of pale text (C3: CER 23.2% at $w = 9$ versus 0.5% at $w = 35$), because the ink occupies a large portion of the window, and the factor $1 - p$ in inequality (6) decreases. A large window worsens the result under non-uniform lighting, because the illumination within the window area is no longer constant (C2: CER 1.3% at $w = 35$ versus 5.2% at $w = 95$). For NA-Mean-C, a multiplier κ ranging from 2 to 3 forms an average CER of 9.1–13.0%. With a smaller κ, the shift is insufficient, and the noise in C4 is interpreted as ink (CER 94.7% at $\kappa = 1$). At higher κ, the shift is excessive, causing ink smearing in C4 and pale ink in C5 (mean CER 36.6% at $\kappa = 6$).

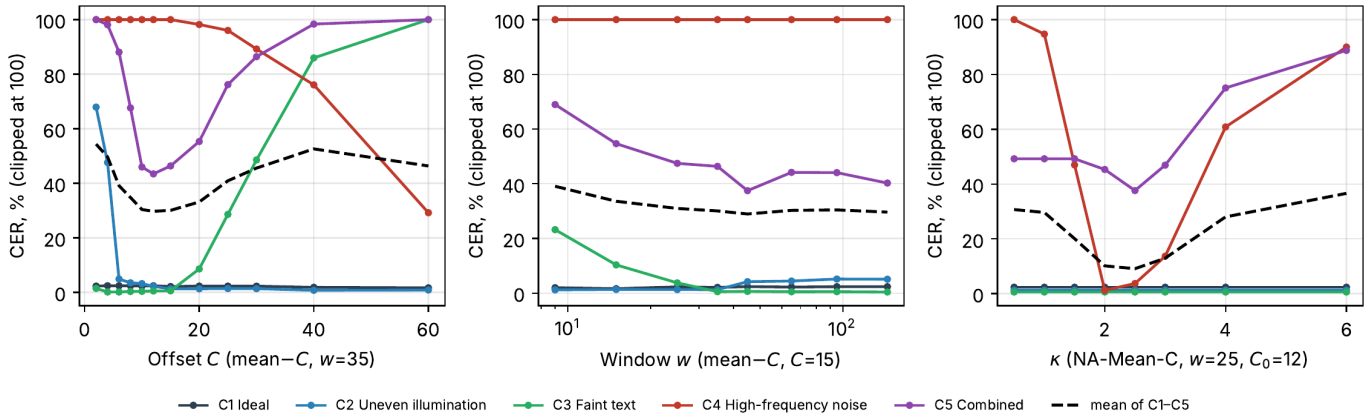


Figure 2. Dependence of Tesseract character error rate (CER) on method parameters. On the left—results of the Mean-C method as a function of the offset C (window size $w = 35$); in the center—results of the Mean-C method as a function of the window size w ($C = 15$); on the right—results of the NA-Mean-C method as a function of the multiplier κ ($w = 25$, $C_0 = 12$). Solid lines represent individual categories C1–C5; the dashed line represents the average across the five categories (10 test images were used for each category; the CER for each image is capped at 100%).

5.4. Compromise Between Accuracy and Cost

In Figure 3, the average CER for all five categories is compared with the binarization time of the fastest implementation of each method (single core, 1000 × 1500 frames; implementations—Section 4.5, values—Section 5.8). Since the realizations differ, the time axis is merely indicative. A method lies on the Pareto front if no other method is both faster and more accurate (Figure 3) [44]. Taking into account all categories, the Pareto front is formed by Otsu, Mean-C from OpenCV, Niblack, Wolf, and NA-Mean-C, in order of increasing time and decreasing CER. NA-Mean-C has the lowest CER (6.89% in 29.0 ms), with noise estimation accounting for 25.9 ms of this time. Gatos (10.55% in 48.8 ms) is both slower and less accurate than NA-Mean-C and therefore does not lie on the front. The

Mean-C with a fixed configuration has the same runtime as the Mean-C configured using OpenCV (4.0 ms), but a higher CER (29.97% vs. 29.04%); therefore, it is not located on the Pareto front.

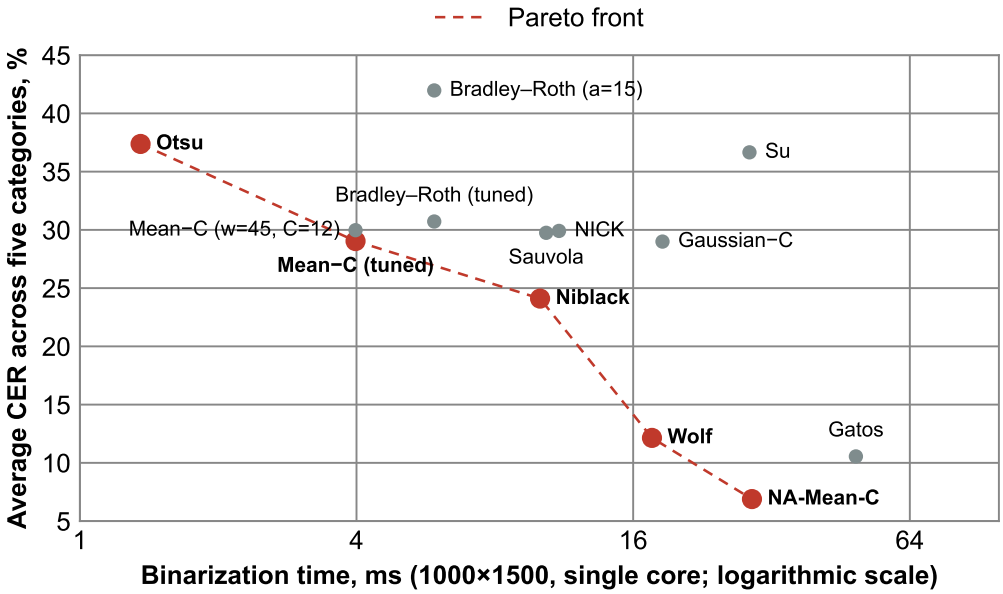


Figure 3. Average character error rate (CER) for categories C1–C5 as a function of frame binarization time (1000 × 1500 frame, single core, fastest implementation; logarithmic scale). Pareto front is shown in red (the Mean-C and Bradley–Roth configurations have identical execution times within the framework of the method; the implementations and parameters used for timing measurements are described in Section 4.5).

5.5. Real Receipts: SROIE

Table 6 presents the CER and the accuracy of extracting the total amount from 80 SROIE test receipts under four conditions (320 documents per method). The parameters were selected using synthetic data and were not re-tuned. Two groups are distinguished. Methods based on local mean (tuned Bradley–Roth, Mean-C with tuned or fixed parameters, NA-Mean-C) and Sauvola achieve a sum extraction accuracy of 53.1–57.5% under all conditions. Gatos, Wolf, Su, Otsu, and Tesseract's internal binarization achieve an accuracy of 40.3–48.8%. NA-Mean-C (55.6%) is more accurate than Gatos (48.8%, McNemar $p = 0.013$), Wolf (46.6%, $p = 0.001$), Su ($p < 0.001$), Otsu ($p < 0.001$) and Tesseract's internal binarization (44.7%, $p = 0.001$). There are no significant differences compared to Sauvola, Bradley–Roth, or Mean-C with a fixed offset ($p \geq 0.34$). This is expected, as the SROIE conditions contain little sensor noise ($\sigma \leq 6$). Global methods fail under combined degradation (R3: Otsu 16.2%, Tesseract internal binarization 18.8%), while average-based methods maintain an accuracy of 38.8–51.2%. The differences in CER between the best methods are approximately one percentage point (43.4–44.9%). Absolute CER values are high (around 40%, even for original scanned images) because, for receipts, the page-level CER is determined by discrepancies in reading order and column layout between Tesseract and the reference transcription. Therefore, CER is used only to compare methods rather than as an absolute measure of quality.

Table 6. Results on SROIE test receipts (conditions R0–R3—Section 4.2). CER—average CER (%); Sum—share of receipts with an accurately extracted total amount (%, parser P-kw). The best value in the column is highlighted in bold. The half-width of the 95% confidence interval for a proportion is approximately ±10 percentage points for a single condition and ±5 percentage points for "All".

Method	CER R0	CER R1	CER R2	CER R3	CER All	Sum R0	Sum R1	Sum R2	Sum R3	Sum All
Otsu	40.9	59.6	40.8	74.5	54.0	55	38	52	16	40
Sauvola	40.2	40.4	46.0	**47.0**	**43.4**	61	58	49	45	53
Wolf	49.1	48.9	49.0	57.1	51.0	54	48	54	31	47
Su	42.3	47.3	41.6	62.4	48.4	50	46	48	29	43
Gatos	39.7	**39.6**	46.7	48.0	43.5	50	55	45	45	49
Bradley–Roth (tuned)	39.9	41.0	46.3	49.1	44.1	65	59	52	49	56
Mean−C (OpenCV, tuned)	41.0	42.1	43.7	48.1	43.7	65	61	52	**51**	**58**
Mean−C (fixed, $w = 45$, $C = 12$)	44.2	41.7	41.9	48.8	44.2	**66**	62	55	39	56
NA-Mean-C (tuned)	44.5	44.2	43.4	47.6	44.9	56	**64**	51	**51**	56
Tesseract internal binarization (grayscale image)	**38.7**	58.0	**39.1**	73.1	52.2	60	36	**64**	19	45

5.6. Separation of Binarization, OCR, and Parsing

A single "success/failure" indicator does not show where the gain originates—from the threshold or from the parser. Therefore, the authors evaluate the parsers on a ground-truth text, which is free from binarization and OCR errors (Table 7), and subsequently on the OCR output corresponding to each binarization method (Table 8).

Table 7. Proportion of receipts with the total amount correctly extracted (%) on the reference text (95% Wilson confidence interval in parentheses).

Dataset	P-freq	P-max	P-kw
SROIE, 525 receipts	42.9 [38.7; 47.1]	45.5 [41.3; 49.8]	83.0 [79.6; 86.0]
Synthetic, 150 receipts	74.7 [67.2; 81.0]	35.3 [28.1; 43.3]	100.0 [97.5; 100.0]

In the SROIE benchmark text, the frequency-based rule P-freq extracts the correct total amount for only 42.9% of the receipts. Of the 300 existing errors, 147 are caused by the threshold of 10.0 (a lower amount), 19 errors occur because the regular expression fails to find the amount (e.g., RM9.00), and 134 errors are caused by the mode selecting a different recurring amount. On synthetic receipts, the rule makes errors in 27 out of 50 receipts with a discount and 14 out of 27 receipts with a service charge. The P-kw keyword rule achieves 83.0% on the SROIE benchmark text.

Table 8. Proportion of SROIE receipts with the total amount correctly extracted (%) for each combination of binarization method and parser (320 documents and conditions R0–R3 combined).

Binarization	P-freq	P-max	P-kw
Otsu	19.1	23.4	40.3
Sauvola	24.7	28.1	53.1
Wolf	17.8	25.3	46.6
Su	17.8	22.2	43.1
Gatos	22.5	28.7	48.8
Bradley–Roth (tuned)	23.1	31.6	56.2
Mean−C (OpenCV, tuned)	26.6	29.1	57.5
Mean−C (fixed, $w = 45$, $C = 12$)	25.9	28.1	55.6
NA-Mean-C (tuned)	25.3	31.6	55.6
Tesseract internal binarization (grayscale image)	23.8	26.9	44.7

The separation of modules shows that, for actual receipts, the extraction rule carries more weight than the choice of threshold. Both parsers are applied to the same recognized

text. Regardless of the binarization method, P-kw retrieves the correct sum 1.9–2.6 times more frequently than P-freq. For example, the NA-Mean-C method, it retrieves the correct sum in 55.6% of receipts versus 25.3%. The difference between the best and worst binarization using P-kw is 17.2 points. The upper limit, determined by the parser itself (P-kw in the reference text), is 83.0%; the remainder of the gap, approximately 25 points, is attributable to OCR errors in the total sum line.

5.7. Photographed Receipts (Qualitative Assessment)

Figure 4 shows fragments of synthetic, SROIE, and photographed input images, together with the results of their binarization. Table 9 presents the total amount printed and its value for each of the five photographed receipts (shown in Figure 5), as extracted by each pipeline. The binary parsing status does not reveal whether the extracted value is correct. The photographs form complete scenes; after scaling the entire frame to a width of $W = 1000$, the characters are only about 10 pixels high. Therefore, the authors cropped the receipt area (using a manual bounding box, which replaces the document detection stage) prior to scaling, and Table 9 presents this specific version. With the Tesseract model in English, no pipeline recovers more than three out of five (Czech) grand totals. Niblack and NA-Mean-C correctly extract the sums from receipts (a), (c), and (d), whereas all other methods—including every other variant based on the mean—extract them only from receipt (a). Without cropping, no method recovers more than one sum. Visually, the mean-based methods in (c)–(e) preserve more faded strokes than Gatos and Sauvola (Figure 5), but the OCR engine does not convert them into correct text. The method parameters were obtained from Table 2 without adjustment. Due to the limited amount of data, these images are used only to illustrate how the methods work.

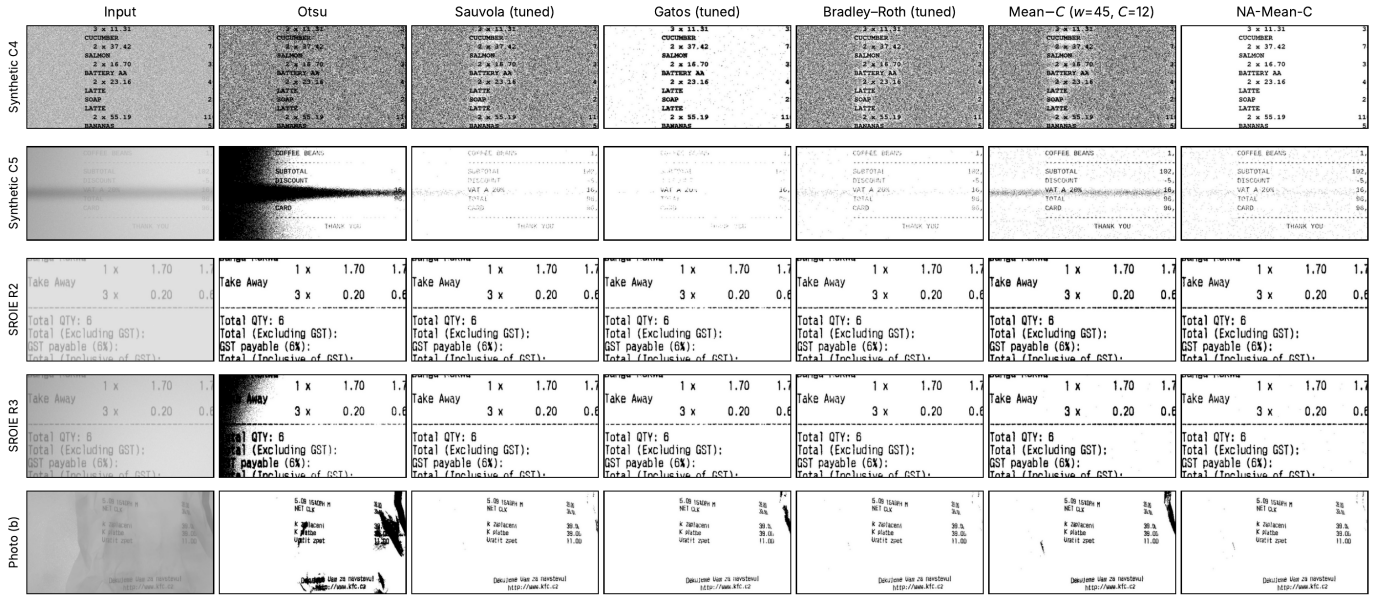


Figure 4. Fragments of input images and binarization results (method parameters are from Table 2; for Mean-C, fixed settings of $w = 45$ and $C = 12$).

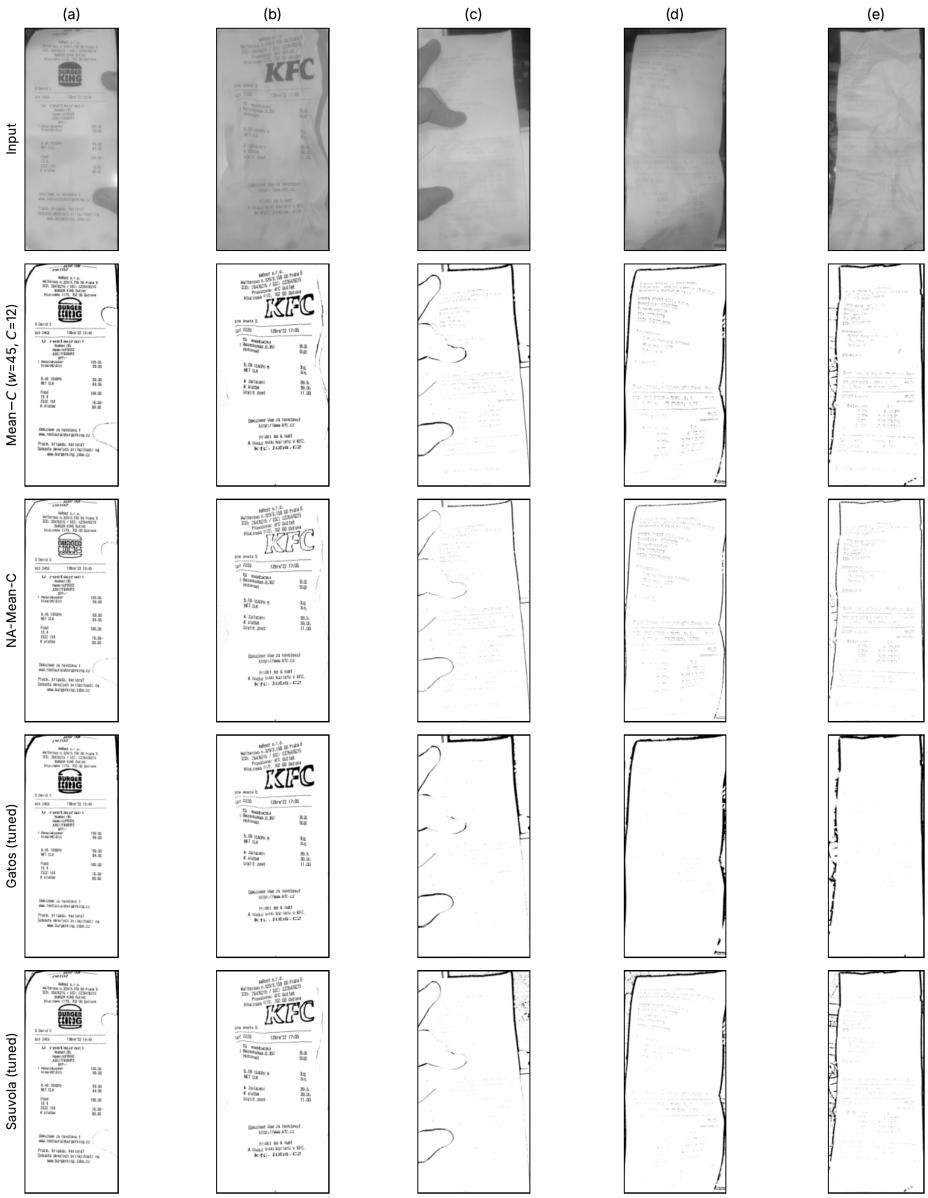


Figure 5. Photographed receipts (**a**–**e**) with the document region cropped, and the results of their binarization (for Mean-C: fixed settings $w = 45$, $C = 12$; for the other methods: parameters from Table 2).

5.8. Processing Time

Table 10 presents the processing time for receipt images using various methods and modules. The binarization time should be separated from the OCR time. For a 1000 × 1500 receipt on a single x86 core, the Mean−C method from OpenCV executes in 4.0 ms. The C++ implementation takes 5.3 ms for the Mean−C method, 5.9 ms for the Bradley–Roth method, and 12.5 ms for the Sauvola method. The 2.4-fold difference between Sauvola and Mean−C (2.1-fold relative to Bradley–Roth), given an identical volume of memory accesses, reflects the additional arithmetic operations in Section 3.4. The reference NumPy implementation requires 107 ms, primarily due to windowed sums with extended indexing (Step 3, 72 ms) and cumulative sums (Step 1). This fact is a feature of the implementation, not of the algorithm. Tesseract OCR processes the same image in 1357 ms; therefore, binarization accounts for less than 1% of the end-to-end processing time for the fixed-shift local-mean implementations (Mean−C in OpenCV and in C++, Bradley–Roth in C++), about 1.4% for

Gaussian−*C*, and about 2% for NA-Mean-C; only the NumPy reference implementations take noticeably longer.

Table 9. Results of extracting the total amount from the photographed receipts (a)–(e): the printed amount and the values obtained via OCR parsers P-freq/P-kw (– denotes amount not found; values matching the printed amount are in bold).

Method	(a)	(b)	(c)	(d)	(e)	Correct
Printed sum	93.00	39.00	99.00	99.00	99.00	–
Otsu	**93.00**/**93.00**	11.00/11.00	–/–	–/–	–/–	1/1
Niblack	**93.00**/**93.00**	–/5.08	**99.00**/**99.00**	**99.00**/**99.00**	–/–	3/3
Sauvola	**93.00**/**93.00**	–/–	–/–	–/–	–/–	1/1
Wolf	**93.00**/**93.00**	50.00/50.00	–/–	–/–	–/–	1/1
NICK	**93.00**/**93.00**	–/–	–/–	–/–	–/–	1/1
Su	**93.00**/**93.00**	11.00/11.00	–/–	–/–	–/–	1/1
Gatos	**93.00**/**93.00**	11.00/11.00	–/–	–/–	–/–	1/1
Bradley–Roth ($a = 15$, not tuned)	**93.00**/**93.00**	–/5.09	–/–	–/–	–/–	1/1
Bradley–Roth (tuned)	**93.00**/**93.00**	39.06/39.06	–/–	–/–	–/–	1/1
Gaussian−*C* (OpenCV, tuned)	**93.00**/**93.00**	–/6.09	–/–	–/–	–/–	1/1
Mean−*C* (OpenCV, tuned)	**93.00**/**93.00**	–/5.08	–/–	–/–	–/–	1/1
Mean−*C* (fixed, $w = 45$, $C = 12$)	**93.00**/**93.00**	–/9.36	–/–	–/–	–/–	1/1
NA-Mean-C (tuned)	**93.00**/**93.00**	39.06/39.06	**99.00**/**99.00**	**99.00**/**99.00**	–/–	3/3
Tesseract internal binarization (grayscale image)	**93.00**/**93.00**	11.00/11.00	–/–	–/–	–/–	1/1

Table 10. Processing time for receipt images using various methods and modules (ms, mean ± standard deviation over 30 runs, single core). x86: Intel Xeon 2.8 GHz. The value for Raspberry Pi 5 is obtained using the same software (Section 4.5). Method parameters during measurement—Section 4.5.

Method/module	x86, 1000×300	x86, 1000×1500	Raspberry Pi 5, 1000×300	Raspberry Pi 5, 1000 × 1500
Otsu (OpenCV)	0.214 ± 0.030	1.36 ± 0.08	0.856 ± 0.120	5.44 ± 0.32
Mean−*C* (OpenCV)	0.691 ± 0.032	3.98 ± 1.01	2.76 ± 0.13	15.92 ± 4.04
Gaussian−*C* (OpenCV)	3.69 ± 0.49	18.53 ± 1.61	14.76 ± 1.96	74.12 ± 6.44
Mean−*C* (NumPy, reference code)	24.48 ± 5.08	107 ± 8	61.20 ± 12.70	268 ± 20
NA-Mean-C (OpenCV block filter + estimator)	6.14 ± 0.70	29.02 ± 0.23	15.35 ± 1.75	72.55 ± 0.58
NA-Mean-C (NumPy)	39.59 ± 5.59	159 ± 15	98.98 ± 13.98	398 ± 38
including noise estimation (Equation (8))	4.87 ± 0.29	25.87 ± 0.77	12.18 ± 0.73	64.68 ± 1.93
Bradley–Roth (NumPy)	29.17 ± 7.60	111 ± 7	72.93 ± 19.00	278 ± 18
Sauvola (NumPy, two integral images)	56.09 ± 4.85	251 ± 26	140 ± 12	628 ± 65
Mean−*C* (stream-based C++)	0.944 ± 0.159	5.32 ± 1.74	2.36 ± 0.40	13.30 ± 4.35
Bradley–Roth (stream-based C++)	1.09 ± 0.22	5.91 ± 1.85	2.73 ± 0.55	14.78 ± 4.63
Sauvola (stream-based C++)	2.90 ± 0.83	12.53 ± 2.06	7.25 ± 2.08	31.33 ± 5.15
Sauvola (Doxa C++)	2.22 ± 0.92	10.35 ± 1.49	5.55 ± 2.30	25.88 ± 3.73
Niblack (Doxa C++)	1.69 ± 0.29	10.04 ± 2.62	4.23 ± 0.73	25.10 ± 6.55
Wolf (Doxa C++)	4.87 ± 1.00	17.58 ± 2.70	12.18 ± 2.50	43.95 ± 6.75
Su (Doxa C++)	5.40 ± 1.08	28.66 ± 5.67	13.50 ± 2.70	71.65 ± 14.18
Gatos (Doxa C++)	10.84 ± 3.47	48.85 ± 9.10	27.10 ± 8.68	122 ± 23
Tesseract OCR (at the Mean-C output)	209 ± 20	1357 ± 128	679 ± 65	4410 ± 416
Parser P-freq	0.005 ± 0.014	0.049 ± 0.013	0.013 ± 0.035	0.123 ± 0.033
Parser P-kw	1.20 ± 0.85	16.64 ± 3.46	3.00 ± 2.13	41.60 ± 8.65
Reference NumPy implementation of Mean-C, step-by-step (Figure 1; 1000 × 300 and 1000 × 1500):				
Type casting (input)	1.25 ± 0.24	2.27 ± 1.62	3.13 ± 0.60	5.68 ± 4.05
Step 1: Integral image	7.02 ± 0.80	31.51 ± 5.37	17.55 ± 2.00	78.78 ± 13.43
Step 2: Window bounds and area	1.85 ± 0.28	6.32 ± 1.32	4.63 ± 0.70	15.80 ± 3.30
Step 3: Window sum	14.08 ± 1.95	71.59 ± 10.33	35.20 ± 4.88	179 ± 26
Step 4: Local mean	1.24 ± 0.24	10.60 ± 1.83	3.10 ± 0.60	26.50 ± 4.58
Decision: threshold and comparison	1.82 ± 0.34	11.47 ± 1.10	4.55 ± 0.85	28.68 ± 2.75

6. Discussion

6.1. Confirmation of Data

The experiments confirm three statements. First, absolute shifts preserve faded ink, which relative shifts with standard parameters remove. This follows from Equation (6) and is evident in category C3 and in the photographed receipts (Figure 5). Secondly, a fixed small offset fails whenever the noise level exceeds approximately C/z; this explains the failure of every method that employs a fixed or relative offset in category C4. Third, selecting the shift based on a robust noise estimate eliminates this failure without altering the rule for individual pixels, and it forms the best average across all five categories in our benchmark. On real-world SROIE receipts, methods based on local averaging and Sauvola extracted the total amount more frequently than Gatos, Wolf, Su, Otsu, and Tesseract's internal binarization. NA-Mean-C did not perform better than Mean-C with a fixed offset, which is consistent with the low sensor noise level under these conditions ($\sigma \leq 6$).

Experiments do not support the claim that Mean$-C$, with or without adaptive shift, generally outperforms methods based on variance or background estimation. Gatos and Wolf perform best on clean images and images with uneven illumination (C1, C2), while Wolf performs better in cases of combined degradation (C5). Otsu performs as well as local methods under uniform illumination. In the five photographed receipts, no pipeline recovers more than three of the five printed amounts. Average-based binarization methods noticeably better preserve faint strokes (Figure 5), but Tesseract fails to recognize characters of such size and contrast. In this case, end-to-end accuracy is limited by the OCR engine, not by the threshold.

6.2. Implications for Streaming Implementation

Based on the analysis of resources (Section 3.4), two practical conclusions can be drawn. In the streaming implementation, memory usage is determined by line buffers that are shared across all local-mean-based methods; consequently, omitting the brightness-squared statistics saves only about 6% of the on-chip memory. Greater savings (2.3-fold for precise bit-widths, 3.0-fold for CPU word sizes) are achieved only when the integral representation of the entire frame is stored, as in the NumPy implementation. A real hardware advantage of Mean$-C$, which Bradley–Roth also has, lies in arithmetic: The division-free comparison derived from Equation (13) requires only adders and multiplication by constants, whereas Sauvola requires a squaring unit, a general multiplier, and a square root operation (or additional multipliers). NA-Mean-C adds a 3×3 filter using existing line buffers and one median per frame. The time differences in Table 10 are highly dependent on the implementation (NumPy, OpenCV, C++), and the results obtained via cloud computing cannot be interpreted as the performance of an embedded device.

6.3. Limitations

Data. Synthetic categories serve as models of actual degradations, and the total-sum parser shares a label vocabulary with the generator; consequently, the parser's results on synthetic text are optimistic. SROIE images are scanned images; conditions R1–R3 introduce synthetic lighting, fading, and noise to real receipts, but they do not replicate all the effects associated with mobile capture (such as perspective, blur, and specular reflections). Only five receipt photos were available and were used solely as an illustration.

Unevaluated Benchmarks. This study does not evaluate DIBCO/H-DIBCO (handwritten and historical documents, a different domain) or the CORD and KORIE receipt datasets. The evaluation accepts any folder of images with reference annotations. Learning-based binarization was not retrained on our data and is considered solely based on the literature.

Hardware Implementation. Hardware implementation has not been synthesized, and power or energy consumption has not been measured; resource estimates are analytical and verified using a bit-accurate software model.

Method Assumptions. The window must be larger than the widest stroke and smaller than the scale of lighting variations, and with large fonts, the center within the window is defined by the ink, and the inner parts of the symbols become empty. The noise estimator assumes that the edges occupy less than half of the pixels; the strong texture of the paper is counted as noise and increases C^*, which may result in the removal of faint ink. The shift C^* is global—that is, a single shift applies to the entire image—so spatially non-uniform noise (for example, along a dark fold in category C5) is not modeled. When $\kappa\hat{\sigma}_n \geq (1-p)\Delta_{\min}$, no shift works, and noise reduction or another method is needed. For Niblack ($w = 95$, $k = -1.0$) and Wolf ($w = 15$, $k = 0.5$), the selected values lie at the edge of the grid, so their optimum may lie outside of it.

Extraction. P-kw is a rule-based baseline method developed using 100 receipts from the SROIE dataset. It does not account for multi-currency receipts or languages other than English and Malay (for example, the Czech labels "K platbě" and "Celkem" on photographed receipts) and does not replace models that incorporate page layout [36,37].

7. Conclusions and Future Work

The article analyzes the local Mean-C threshold, $T = m - C$, for the OCR of low-contrast thermal receipts. The thresholding form itself is the classic Mean-C rule, implemented in OpenCV. The contribution of this work consists of three results. First, a condition is derived under which a constant shift C separates the ink from the background: C must be greater than the noise level but less than the ink contrast. Secondly, the NA-Mean-C method is proposed, which determines C based on an estimate of the image noise; as a result, it does not fail under high-noise conditions—situations where the standard Mean-C method (with a fixed C) would incorrectly classify the noisy background as ink. Third, the methods are compared under identical conditions: the parameters for all methods are tuned on a separate validation set, and all five synthetic categories are accounted for in all average values.

1. The image formation model establishes an allowable shift range $z\sigma_n \leq C < (1-p)\Delta_{\min}$. The model explains why an absolute shift preserves a faded imprint, which relative shifts with standard parameters would remove, and why a fixed small shift fails when the noise level exceeds approximately C/z.
2. The NA-Mean-C method is proposed—the Mean-C method with a shift determined by a robust estimate of the noise level: $C^* = \max(C_0, \kappa\hat{\sigma}_n)$, $\kappa = 2.5$, $C_0 = 12$. The thresholding rule for each pixel remains unchanged, so the structure involving a single integral image (or a single set of line buffers) is preserved; additionally, a 3×3 filter and a single median calculation per frame are required. In category C4 (high-frequency noise), NA-Mean-C achieves an F-measure of 96.4% and a CER of 2.7%, while Mean-C with a tuned fixed offset achieves 15.9% and 99.7%. In categories C1–C3, the NA-Mean-C shift is equal to $C_0 = 12$, and both methods provide similar results. In the combined category C5, the advantage of NA-Mean-C is moderate (CER 27.2% vs. 40.5%), while the Wolf method outperforms both (CER 13.7%). Across all five categories, NA-Mean-C achieves an average F-measure of 91.61% and an average CER of 6.89%; for Mean-C with a tuned fixed shift, these values are 74.40% and 29.04%, while with fixed settings of $w = 45$ and $C = 12$, they are 71.97% and 29.97%. The proportion of synthetic receipts for which the P-kw parser accurately extracted the total amount is 92.7% for NA-Mean-C, 70.0% for Mean-C with a configured fixed offset and 64.7% for Mean-C with a fixed configuration. The absolute values of this proportion on synthetic data are

inflated for all methods because the parser and the receipt generator use the same label vocabulary (Section 6.3); therefore, they are suitable only for comparing the methods against one another. NA-Mean-C works slower than Mean-C: 29.0 ms versus 4.0 ms for Mean-C (using OpenCV) on a 1000×1500 frame, with noise estimation accounting for 25.9 ms. The total time for NA-Mean-C is approximately 2% of the recognition time using Tesseract on the same image (Section 5.8).

3. On 80 public SROIE receipts across four conditions (320 documents), with parameters tuned solely on synthetic data, methods based on local averages (tuned Bradley–Roth, Mean−C with either tuned or fixed parameters, and NA-Mean-C) and the Sauvola method successfully extracted the total amount in 53.1–57.5% of the documents, whereas Gatos, Wolf, Su, Otsu, and Tesseract's internal binarization methods achieved this for 40.3–48.8% of the documents. NA-Mean-C (55.6%) is significantly more accurate than Gatos (48.8%), Wolf (46.6%), Su, Otsu, and Tesseract's internal binarization (44.7%) (McNemar's test, $p \leq 0.013$), but it does not differ significantly from Sauvola, Bradley–Roth, and Mean-C with a fixed offset ($p \geq 0.34$). The difference in CER between the best methods is approximately one percentage point (43.4–44.9%). The SROIE conditions contain little sensor noise ($\sigma \leq 6$); therefore, the advantage of NA-Mean-C observed in category C4 is absent in this dataset.
4. In the SROIE benchmark dataset, the frequency-based rule P-freq identifies the total amount in 42.9% of the receipts; of its 300 errors, 147 are caused by the threshold value of 10.0, and 134 are caused by the selection of a different recurring amount (the mode heuristic). The keyword-based rule P-kw achieves 83.0%. Separate evaluation of binarization, OCR, and extraction is necessary to correctly identify the source of the end-to-end gain.
5. In a streaming implementation, Mean-C saves approximately 6% of the on-chip memory compared to Sauvola (374 vs. 396 kbit for $W = 1000$, $w = 45$). Its real advantage, shared with Bradley–Roth, is arithmetic that does not involve general multiplication, division, or the calculation of square roots. NA-Mean-C adds only a 3×3 filter, operating on existing line buffers, and a single median per frame.

Future work includes synthesizing the data flow scheme on an FPGA and measuring resource usage and power consumption; evaluating the method on benchmarks of photographed receipts (CORD, KORIE) and on DIBCO; estimating the noise level locally, rather than globally, for images with spatially varying noise; and replacing the rule-based extractor with a model that takes page layout into account [36,37].

Author Contributions: Conceptualization, R.L. and O.L.; methodology, R.L. and O.L.; software, S.B. and R.L.; validation, R.L., V.K. and P.K.; formal analysis, R.L., V.K. and O.L.; investigation, V.K. and P.K.; resources, R.L. and O.L.; data curation, S.B. and V.K.; writing—original draft preparation, R.L., P.K. and S.B.; writing—review and editing, S.B., A.S. and R.L.; visualization, P.K. and S.B.; supervision, A.R. and A.S.; project administration, A.R. and A.S.; funding acquisition, S.B. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors are grateful to all colleagues and institutions that contributed to the research and made it possible to publish its results.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASIC	Application-specific integrated circuit
BRAM	Block random access memory
CER	Character error rate
CPU	Central processing unit
DIBCO	Document Image Binarization Contest
DRD	Distance reciprocal distortion
DSP	Digital signal processing
FM	F-measure
FPGA	Field-programmable gate array
ICDAR	International Conference on Document Analysis and Recognition
LSTM	Long short-term memory
LUT	Look-up table
NA-Mean-C	Noise-adaptive Mean-C
OCR	Optical character recognition
pFM	Pseudo-F-measure
PSNR	Peak signal-to-noise ratio
SROIE	Scanned Receipt OCR and Information Extraction
VAT	Value-added tax

References

1. Merenda, M.; Porcaro, C.; Iero, D. Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors* **2020**, *20*, 2533. https://doi.org/10.3390/s20092533.
2. Gatos, B.; Pratikakis, I.; Perantonis, S.J. Adaptive Degraded Document Image Binarization. *Pattern Recognit.* **2006**, *39*, 317–327. https://doi.org/10.1016/j.patcog.2005.09.010.
3. Kieu, V.C.; Visani, M.; Journet, N.; Domenger, J.P.; Mullot, R. A character degradation model for grayscale ancient document images. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 685–688. Available online: https://ieeexplore.ieee.org/document/6460227 (accessed on 17 November 2026).
4. Saddami, K.; Munadi, K.; Away, Y.; Arnia, F. Effective and Fast Binarization Method for Combined Degradation on Ancient Documents. *Heliyon* **2019**, *5*, e02613. https://doi.org/10.1016/j.heliyon.2019.e02613.
5. Chaki, N.; Shaikh, S.H.; Saeed, K. A Comprehensive Survey on Image Binarization Techniques. In *Studies in Computational Intelligence*; Springer: New Delhi, India, 2014; pp. 5–15. https://doi.org/10.1007/978-81-322-1907-1_2.
6. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. https://doi.org/10.1109/tsmc.1979.4310076.
7. Wellner, P.D. *Adaptive Thresholding for the DigitalDesk*; Technical Report EPC-1993-110; Rank Xerox Research Centre: Cambridge, UK, 1993.
8. Bradley, D.; Roth, G. Adaptive Thresholding Using the Integral Image. *J. Graph. Tools* **2007**, *12*, 13–21. https://doi.org/10.1080/2151237x.2007.10129236.
9. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *25*, 120–125.
10. Fisher, R.; Perkins, S.; Walker, A.; Wolfart, E. Adaptive Thresholding. In *Hypermedia Image Processing Reference (HIPR2)*; University of Edinburgh: Edinburgh, UK, 2003. Available online: https://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm (accessed on 23 October 2026).
11. Niblack, W. *An Introduction to Digital Image Processing*; Strandberg Publishing Company: Birkeroed, Denmark, 1986; pp. 115–116.
12. Lazzara, G.; Géraud, T. Efficient Multiscale Sauvola's Binarization. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2014**, *17*, 105–123. https://doi.org/10.1007/s10032-013-0209-0.
13. Wolf, C.; Jolion, J.-M. Extraction and Recognition of Artificial Text in Multimedia Documents. *Form. Pattern Anal. Appl.* **2004**, *6*, 309–326. https://doi.org/10.1007/s10044-003-0197-7. Erratum in *Pattern Anal. Appl.* **2004**, *7*, 224. https://doi.org/10.1007/s10044-004-0216-3.

14. Khurshid, K.; Siddiqi, I.; Faure, C.; Vincent, N. Comparison of Niblack Inspired Binarization Methods for Ancient Documents. *SPIE Proc.* **2009**, *7247*, 72470U. https://doi.org/10.1117/12.805827.
15. Shafait, F.; Keysers, D.; Breuel, T.M. Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images. *SPIE Proc.* **2008**, *6815*, 681510. https://doi.org/10.1117/12.767755.
16. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. I-511–I-518. https://doi.org/10.1109/cvpr.2001.990517.
17. Crow, F.C. Summed-Area Tables for Texture Mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*; ACM:New York, NY, USA, 1984; pp. 207–212. https://doi.org/10.1145/800031.808600.
18. Immerkær, J. Fast Noise Variance Estimation. *Comput. Vis. Image Underst.* **1996**, *64*, 300–302. https://doi.org/10.1006/cviu.1996.0060.
19. Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; Jawahar, C.V. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 1516–1520. https://doi.org/10.1109/icdar.2019.00244.
20. Sauvola, J.; Pietikäinen, M. Adaptive Document Image Binarization. *Pattern Recognit.* **2000**, *33*, 225–236. https://doi.org/10.1016/s0031-3203(99)00055-2.
21. Su, B.; Lu, S.; Tan, C.L. Binarization of Historical Document Images Using the Local Maximum and Minimum. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*; ACM: New York, NY, USA, 2010; pp. 159–166. https://doi.org/10.1145/1815330.1815351.
22. Su, B.; Lu, S.; Tan, C.L. Robust Document Image Binarization Technique for Degraded Document Images. *IEEE Trans. Image Process.* **2013**, *22*, 1408–1417. https://doi.org/10.1109/tip.2012.2231089.
23. Lu, S.; Su, B.; Tan, C.L. Document Image Binarization Using Background Estimation and Stroke Edges. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2010**, *13*, 303–314. https://doi.org/10.1007/s10032-010-0130-8.
24. Howe, N.R. Document Binarization with Automatic Parameter Tuning. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2013**, *16*, 247–258. https://doi.org/10.1007/s10032-012-0192-x. Erratum in *Int. J. Doc. Anal. Recognit. (IJDAR)* **2013**, *16*, 259. https://doi.org/10.1007/s10032-013-0200-9.
25. Sulaiman, A.; Omar, K.; Nasrudin, M.F. An Efficient Phase-Based Binarization Method for Degraded Historical Documents. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2019**, *9*, 2207–2215. https://doi.org/10.18517/ijaseit.9.6.7774.
26. Ntirogiannis, K.; Gatos, B.; Pratikakis, I. Performance Evaluation Methodology for Historical Document Image Binarization. *IEEE Trans. Image Process.* **2013**, *22*, 595–609. https://doi.org/10.1109/tip.2012.2219550.
27. Pratikakis, I.; Zagoris, K.; Barlas, G.; Gatos, B. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 1395–1403. https://doi.org/10.1109/icdar.2017.228.
28. Lu, H.; Kot, A.C.; Shi, Y.Q. Distance-Reciprocal Distortion Measure for Binary Document Images. *IEEE Signal Process. Lett.* **2004**, *11*, 228–231. https://doi.org/10.1109/lsp.2003.821748.
29. Tensmeyer, C.; Martinez, T. Document Image Binarization with Fully Convolutional Neural Networks. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 99–104. https://doi.org/10.1109/icdar.2017.25.
30. Calvo-Zaragoza, J.; Gallego, A.-J. A Selectional Auto-Encoder Approach for Document Image Binarization. *Pattern Recognit.* **2019**, *86*, 37–47. https://doi.org/10.1016/j.patcog.2018.08.011.
31. Vo, Q.N.; Kim, S.H.; Yang, H.J.; Lee, G. Binarization of Degraded Document Images Based on Hierarchical Deep Supervised Network. *Pattern Recognit.* **2018**, *74*, 568–586. https://doi.org/10.1016/j.patcog.2017.08.025.
32. Tarbă, N.; Boiangiu, C.-A.; Voncilă, M.-L. Document Image Binarization Using Various Machine Learning Models and Ensembles Trained on Classic Local and Global Binarization Algorithms and Image Statistics. *MAKE* **2026**, *8*, 149. https://doi.org/10.3390/make8060149.
33. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. https://doi.org/10.48550/ARXIV.1704.04861.
34. Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; Lee, H. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In Proceedings of the Workshop on Document Intelligence at NeurIPS, Vancouver, BC, Canada, 8–14 December 2019. Available online: https://openreview.net/forum?id=SJl3z659UH (accessed on 17 November 2026).
35. Kasem, M.S.; Mahmoud, M.; Senussi, M.F.; Abdalla, M.; Kang, H.S. KORIE: A Multi-Task Benchmark for Detection, OCR, and Information Extraction on Korean Retail Receipts. *Mathematics* **2026**, *14*, 187. https://doi.org/10.3390/math14010187.

36. Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M. LayoutLM: Pre-Training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, 2020; pp. 1192–1200. https://doi.org/10.1145/3394486.3403172.
37. Appalaraju, S.; Jasani, B.; Kota, B.U.; Xie, Y.; Manmatha, R. DocFormer: End-to-End Transformer for Document Understanding. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 973–983. https://doi.org/10.1109/iccv48922.2021.00103.
38. Thompson, K. Programming Techniques: Regular Expression Search Algorithm. *Commun. ACM* **1968**, *11*, 419–422. https://doi.org/10.1145/363347.363387.
39. Zhai, X.; Bensaali, F.; Sotudeh, R. Real-time Optical Character Recognition on Field Programmable Gate Array for Automatic Number Plate Recognition System. *IET Circuits Devices Syst.* **2013**, *7*, 337–344. https://doi.org/10.1049/iet-cds.2012.0339.
40. Li, L.; Sau, C.; Fanni, T.; Li, J.; Viitanen, T.; Christophe, F.; Palumbo, F.; Raffo, L.; Huttunen, H.; Takala, J.; et al. An Integrated Hardware/Software Design Methodology for Signal Processing Systems. *J. Syst. Archit.* **2019**, *93*, 1–19. https://doi.org/10.1016/j.sysarc.2018.12.010.
41. ExpressExpense. Free Receipt Images OCR Machine Learning Dataset. Available online: https://expressexpense.com/blog/free-receipt-images-ocr-machine-learning-dataset/ (accessed on 5 May 2026).
42. Petty, B. Δoxa Binarization Framework (ΔBF): Image Binarization Algorithms and DIBCO-Compatible Performance Metrics. Source Code. 2026. Available online: https://github.com/brandonmpetty/Doxa (accessed on 23 October 2026).
43. Smith, R. An Overview of the Tesseract OCR Engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 23–26 September 2007; Volume 2, pp. 629–633. https://doi.org/10.1109/icdar.2007.4376991.
44. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; John Wiley & Sons: Chichester, UK, 2001.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.