International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-026-02513-9

ORIGINAL ARTICLE

Evaluating the suitability of handwritten text recognition for industrial operations: a practical case study on acceptance sampling inspection workflow

Luca Martuscelli[1] · **Italo Cesidio Fantozzi**[1] · **Francesco Mancusi**[2] · **Massimiliano Maria Schiraldi**[1]

Received: 9 June 2025 / Revised: 30 December 2025 / Accepted: 27 January 2026

Abstract
Handwritten Text Recognition (HTR) is an Artificial Intelligence (AI) technology designed to interpret and digitize handwritten text, enabling automated data extraction and reducing the reliance on manual processing. While HTR has been extensively developed in theory, leveraging advanced machine learning and pattern recognition models, its practical adoption remains uneven. Documented applications exist in fields such as historical document digitization, banking, and healthcare; however, its use in industrial and manufacturing contexts is still largely unexplored. This practical case study explores the use of HTR in an industrial setting by redesigning the operational execution of acceptance sampling activities in a pharmaceutical manufacturing company, in which handwritten records are traditionally used. The performance of the HTR software, which employs a patented hybrid stroke-based recognition method combining neural and statistical classifiers with structural matching, was tested and evaluated in terms of recognition accuracy and process efficiency to assess its potential to meet industrial standards. Preliminary findings indicate that while HTR can improve efficiency, limitations in recognition accuracy, and the need for human validation restrict full automation. Although promising, further advancements in recognition models and system adaptability are needed to ensure reliable and seamless integration into industrial workflows.

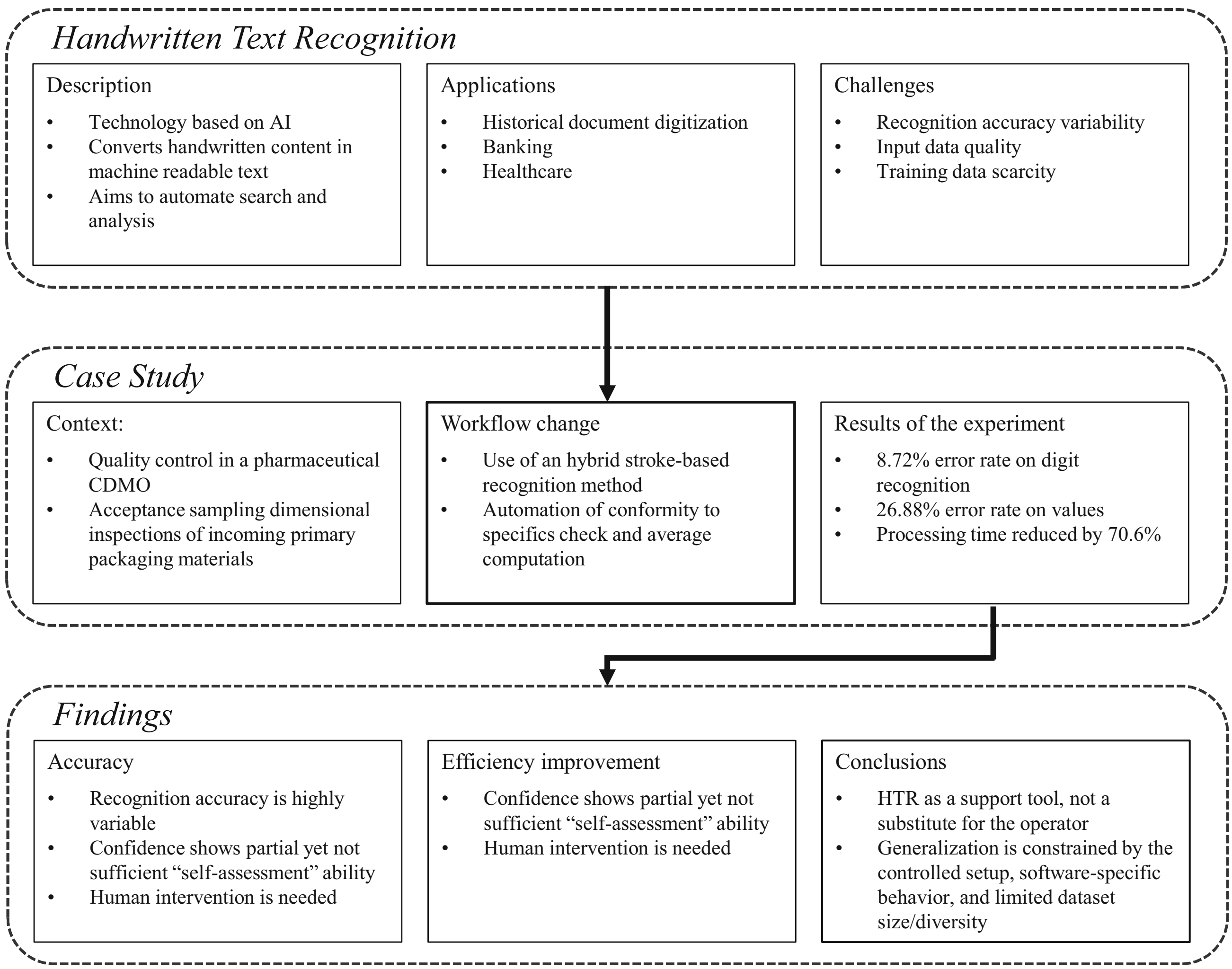
✉ Luca Martuscelli
luca.martuscelli@uniroma2.it

1 Department of Enterprise Engineering, University of Rome Tor Vergata, 00133 Rome, Italy

2 Department of Engineering, Industrial Systems, University of Basilicata, 85100 Potenza, Italy

Published online: 24 February 2026

Graphical Abstract



Keywords Handwritten text recognition · Artificial intelligence · Quality control · Handwritten data acquisition

1 Introduction

Handwritten Text Recognition (HTR) has seen significant progress in recent years, emerging as a crucial research focus within the broader fields of Artificial Intelligence (AI) and pattern recognition, due to the growing demand for digital transformation and the need for faster, more efficient data processing methods. Researchers have explored various techniques to facilitate the digitization of historical archives, medical prescriptions, administrative forms, and modern handwritten records, employing both spatial (offline) and temporal (online) recognition approaches [1]. While Optical Character Recognition (OCR) for printed text has reached high accuracy levels, HTR remains a more complex challenge due to handwriting variability, inconsistencies in letter formation, and document imperfections. Advances in machine learning, particularly deep learning architectures such as Convolutional Neural Networks (CNNs) and Transformers, have significantly improved recognition performance, leading to growing interest in HTR applications. However, its adoption in real-world scenarios is still constrained by limitations in accuracy and reliability, particularly in environments where handwritten data lack structure or follow informal notation.

HTR has been successfully applied in fields such as historical document digitization, medical record processing, and financial services, where it improves data accessibility and reduces manual transcription errors. Despite these advancements, its adoption in industrial and manufacturing environments remains largely unexplored. Given the increasing push toward Industry 4.0 and automation, HTR could play a crucial role in digitizing operational data, reducing human error in quality control, and improving overall process efficiency. However, its integration into industrial workflows presents unique challenges, including the variability of handwritten annotations, the need for high recognition accuracy and large training datasets. As a result, these systems are still considered unsatisfactory for many industrial uses [2] and documented studies on HTR applications in industry remain scarce. This gap highlights the need for investigations that evaluate digital tools application in industrial environments, focusing on their operational feasibility and impact rather than on algorithmic innovation alone, in line with recent industrial engineering case-study-based assessments of data-driven technologies [3].

At the same time, other types of AI-based computer vision tools are already employed to automate tasks in quality inspection, object detection, and process control. For instance [4] developed a hybrid Bayesian-optimized CNN–SVM (Support Vector Machine) model for real-time surface-roughness classification from in-process machined surface images, demonstrating near-complete automation of quality assessment. [5] applied computer-vision and support-vector-machine methods to the quality identification of traditional Chinese medicine slices, achieving a 95% classification accuracy and illustrating the role of AI in visual quality control. [6] integrated digital-twin simulation with computer vision to automate robotic pick-and-place operations, enabling adaptive coordination between physical and virtual production environments. [7] employed optical sensors combined with artificial neural networks for precision measurement and fault detection in metallurgical processes, improving accuracy and reducing manual inspection requirements. The demonstrated effectiveness of these solutions suggests that investigating the applicability of HTR within similar operational contexts is both relevant and timely.

To explore the feasibility of integrating HTR into industrial processes, a practical case study was conducted in the context of quality control, specifically within the operational execution of acceptance sampling inspection. The study does not address the design or statistical properties of acceptance sampling plans (e.g., single or double sampling, OC curves, or AQL/LTPD), which are assumed as predefined and outside the scope of this work. This context was chosen as it traditionally relies heavily on manually recorded data, particularly to document inspection measurements and compliance decisions: often operators record data on forms, which are later processed to determine whether a batch meets predefined standards. The processing of data reported in this way can be time-consuming, limiting process efficiency. To assess whether HTR can enhance this workflow, this study evaluates an HTR-assisted system in a pharmaceutical manufacturing setting, analyzing its recognition accuracy, processing speed, and overall suitability under controlled experimental conditions. The utilized HTR system employs a patented hybrid stroke-based recognition method which combines neural and statistical classifiers with structural matching and works offline, producing interpreted values and confidence scores to assist operator validation. To structure the analysis, the following research questions are addressed:

- RQ1: How accurately can HTR software recognize and process handwritten numerical data in an industrial setting?
- RQ2: To what extent does the use of HTR software have the potential to improve the efficiency of data processing compared to manual methods?
- RQ3: What are the key technical limitations and challenges associated with implementing HTR in a structured industrial workflow?

The research emphasizes the increasing importance of HTR as a tool for digital transformation and underlines the need for ongoing improvements in AI-based recognition systems. By evaluating the suitability of HTR in an industrial environment, the paper explores a new application domain for the technology, and through the examination of the HTR-assisted workflow against the conventional manual process, it offers insights into the potential advantages and obstacles of incorporating this technology into operational processes. Moreover, it addresses the lack of research on HTR applications in structured industrial workflows, contributing to bridge an important gap in the field and paving the way for further research and similar case studies in manufacturing and operations.

2 Literature review

2.1 Handwritten text recognition

Handwritten Text Recognition refers to the mechanism that enables the automated interpretation and conversion of handwritten documents into editable, searchable, and analyzable digital text [8]. The primary goal of HTR is to replicate human reading abilities, allowing machines to efficiently read, modify, and engage with text [8]. Thus, its key advantage lies in the ability to automate and optimize handwritten data processing, from basic transcription to extracting and transmitting information to other systems or computing relevant parameters.

Within Computer Vision, which is a field of Artificial Intelligence that enables systems to extract significant information from graphical data [9], HTR is a part of the larger Optical Character Recognition field [10], which also encompasses systems designed for identifying printed text; in its turn, HTR can be categorized into online and offline systems, depending on how the handwritten input is captured and

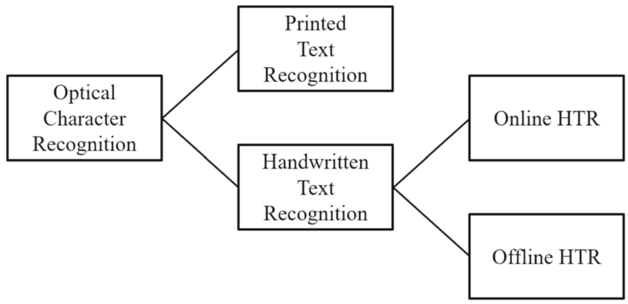


Fig. 1 Optical character recognition systems classification [10]

processed. While offline HTR systems process static images of handwritten text, typically obtained from scanned documents or photos, online systems recognize handwriting as it is being written using electronic devices like tablets, or digital pens, capturing not only the text but also dynamic information such as the movement, velocity, and stroke order of the writing [11]. Online systems are considered more complex and advanced, resulting in high recognition rates [10]. The recognition systems classification is shown in Fig. 1.

While there is no strict consensus on terminology, HTR is generally distinguished from Handwritten Character Recognition (HCR), with the first focusing on recognizing entire words and lines of text from scanned documents, while the latter refers to the identification of individual handwritten characters in various forms [12].

The offline HTR process typically follows a structured sequence of steps, as outlined by AlKendi et al. [1], which are in order:

- Image digitization: The conversion of handwritten text into a digital image using a scanner or camera to serve as input for the system.
- Pre-processing: A crucial initial stage that includes several subprocesses:
  - Binarization: Converts the digital image into a binary format (black-and-white pixels) to distinguish text from the background.
  - Noise removal: Eliminates unwanted artifacts from digitization that may distort textual information.
  - Edge detection: Identifies the boundaries of the text within the image.
  - Skew detection and correction: Aligns the text properly by correcting angular deviations.
  - Normalization: Standardizes the shape and size of the image while scaling features within a fixed range.
- Segmentation: Decomposes handwritten text into paragraphs, lines, words, and ultimately individual characters, forming the basis for recognition. This step is particularly complex for cursive handwriting due to Sayre's paradox, which states that a character cannot be recognized without first being segmented, but segmentation is impossible without prior recognition [13].
- Features extraction: Identifies key attributes within the image to reduce data dimensionality while preserving relevant information.
- Classification: Matches the extracted features with predefined character models, determining the most likely class for each symbol.
- Post-processing: Improves overall accuracy by applying error correction methods, such as dictionary-based validation and statistical techniques.

The resulting workflow is schematized in Fig. 2.

While these steps are common across most HTR systems, the underlying methodologies and technical implementations vary widely, leading to an ever-expanding array of models. As highlighted again by AlKendi et al. [1], recent research has introduced several deep learning architectures for HTR, including:

- Convolutional neural networks (CNNs)
- Convolutional recurrent neural networks (CRNNs)
- Sequence-to-sequence transformers
- Bidirectional long-short-term memory (BLSTM) networks

These models require extensive training on large and diverse handwriting datasets to learn to recognize different styles, stroke variations, and contextual dependencies. The quality and quantity of training data are crucial, as models trained on limited or biased datasets may struggle to generalize effectively to unseen handwriting, leading to performance inconsistencies.

Numerous studies have explored the use of these models across different languages and contexts. For instance, [14] developed a dual-CNN deep learning system for Arabic handwriting recognition, while [15] proposed an approach integrating transformer transducers and sequence-to-sequence transducers. Other notable contributions include the Attention-Based Fully Gated CNN with Connectionist Temporal Classification, tested on English and Russian datasets [16], and a depthwise convolutional model combined with a grid CNN [17]. Additionally, [18] applied a CRNN-based approach for the recognition of historical Greek manuscripts.

2.2 Applications

HTR technologies find application in all fields where handwritten documents are widespread and necessitate processing, whether through transcription, data extraction, or automated registration. Thus, its utilization is well documented

Fig. 2 Typical workflow of an HTR system [1]

in certain domains, such as historical research and financial transactions, and its potential is progressively extending into new areas, showcasing significant benefits in terms of accuracy, efficiency, and scalability.

One of the most significant applications of HTR is the digitization of historical manuscripts, which enables the preservation and accessibility of fragile or unique documents. This is particularly crucial for materials that are physically difficult to handle due to their age, condition, or complexity [19]. Advanced HTR platforms, such as Transkribus, leverage neural networks trained on various handwriting samples, achieving recognition accuracies of up to 98% [19]. These technologies are widely adopted by libraries, archives, and academic institutions. For instance, the Transcribe Bentham project has successfully used Transkribus to process the works of philosopher Jeremy Bentham, while the In Codice Ratio project has applied HTR to Vatican archival manuscripts, integrating convolutional neural networks and statistical language models to refine recognition, often in collaboration with paleographers [19].

HTR has been increasingly adopted in the banking sector, particularly for automating the processing of handwritten checks. Manual verification of checks is a time-consuming, costly, and error-prone process, requiring bank personnel to read and input handwritten details such as the payee's name, amount, and date [19, 20]. To streamline this workflow, HTR-based solutions have been developed to automatically extract and process handwritten fields. Some approaches leverage CNNs to optimize recognition through self-learning mechanisms [21], while others employ segmentation, feature extraction, and classification techniques such as Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVMs) [20]. Recent advancements have further enhanced the accuracy and efficiency of check processing, incorporating deep learning and image processing techniques to minimize human intervention: [22] demonstrated a deep learning-based system capable of extracting information from checks achieving 99.14% accuracy in handwritten numeric character recognition and 98.10% accuracy in signature verification through Scale Invariant Feature Transform (SIFT) and SVM classification. These advancements significantly reduce operational costs and improve security, making check processing more reliable and scalable.

Lastly, healthcare is increasingly utilizing HTR, as handwritten documents remain prevalent in prescriptions, medical records, and clinical reports. The interpretation of doctor handwriting is a critical challenge in this field, as unclear prescriptions can lead to medication errors with serious consequences [22, 23]. To address this issue, researchers have developed both online and offline HTR models: [24] proposes an online system that records handwriting in real time using smartpen technology, while [23] introduces an offline model based on neural networks and Python programming to process handwritten prescriptions. Beyond prescriptions, HTR is being used for converting patient records and medical reports into digital forms, improving data accessibility and administrative efficiency. [25] developed a deep learning system for recognizing handwritten clinical notes, demonstrating high accuracy in extracting medical data. These advancements support workflow automation in healthcare settings, easing administrative tasks and minimizing transcription errors.

2.3 Challenges and limitations

Despite the significant advancements in HTR technologies in recent years, several technical challenges and limitations still hinder their effectiveness and reliability. The most critical issue concerns recognition accuracy, which is heavily variable, even within the same models and technologies, influenced by the irregularity of handwriting styles. Numerous factors contribute to this variability, such as historical period, geographic origin, and individual differences in handwriting. For instance, [26] describe how the irregular or degraded handwriting of 17th-century notaries negatively impacted HTR model performance, leading to high error rates, while [19] highlights that handwriting complexity, the presence of abbreviations, and non-standard symbols remain key obstacles for HTR systems. Even when fine-tuned for specific authors or writing styles, HTR systems can still exhibit notable error rates [26], and consequently human intervention remains essential, both for error correction and validation of transcriptions [19]. This dependence on manual review limits full automation, reducing efficiency gains and slowing transcription workflows.

Another significant challenge is given by image quality, which directly affects both training data and text recognition performance. Low-resolution images, poorly executed scans, or deteriorated documents pose major barriers to accurate character recognition. Poor image conditions severely degrade model accuracy [26], sometimes requiring custom-trained models or manual corrections to improve transcription results, and low-quality documents not only introduce recognition errors but also hamper the training process, preventing models from learning effectively [19].

A further limitation is the availability and diversity of training datasets, which are essential for developing robust and adaptable HTR models. Effective systems require large-scale datasets covering a broad range of handwriting styles [2]. However, such datasets are often difficult to access, for historical or rare manuscripts [19], or to produce [2]. The lack of representative training data restricts HTR models' ability to generalize across different handwriting styles, leading to inconsistent performance.

3 Case study

This section presents the conducted practical case study, focused on the application of HTR within the operational execution of acceptance sampling inspection, a crucial step in quality control aimed at ensuring that incoming materials meet regulatory and technical specifications. The study was carried out in a Contract Development and Manufacturing Organization (CDMO), a third-party pharmaceutical company that provides integrated services covering both drug development and manufacturing. Specializing in gene therapies and biologic drugs, the company operates under strict regulatory oversight from international agencies such as the FDA (Food and Drug Administration), EMA (European Medicines Agency), and AIFA (Agenzia Italiana del Farmaco), adhering to GMP (Good Manufacturing Practices) and ISO standards to guarantee product quality and safety. This study specifically evaluates HTR's potential to streamline the operational execution of dimensional inspections by automating the processing of handwritten inspection forms, aiming to improve efficiency and consistency in quality control workflows while reducing manual workload.

Acceptance sampling inspection is a fundamental quality control practice employed in industrial and manufacturing contexts to ensure product quality. The procedure entails a set of operations performed on a discrete quantity of products (e.g. a batch) to determine whether they should progress through the supply chain. For instance, as the case discussed in this research, acceptance sampling may be conducted on incoming materials before production to assess their suitability. However, aspects related to the design and statistical properties of acceptance sampling plans–such as single or double sampling schemes, OC curves, and AQL/LTPD considerations–are assumed as predefined and are considered here solely as part of the operational context rather than as an object of analysis. The goal of acceptance sampling is to minimize total quality-related costs, balancing inspection expenses (time, labor, equipment) and non-inspection costs (waste, returns, rework, etc.). Since acceptance sampling is usually conducted directly on the shop floor or in warehouses, where inspection data are collected manually by operators, it provides an ideal environment for introducing HTR with the aim of enhancing its operational execution and data processing efficiency.

The acceptance sampling inspection process examined in this study pertains to primary packaging materials, sourced from external suppliers. These materials, including bottles, blister packs, caps, and other components, are sampled upon arrival to evaluate their dimensional compliance with technical specifications and determine their suitability for production. When a batch of materials is received and identified in the incoming warehouse, a specific inspection method is selected from a set of predefined procedures. The selection criteria depend on the type of material, the supplier and the intended production process, ensuring a standardised approach to quality control, and the inspection method defines key parameters, such as the sample size and the specific dimensions to be measured. Once the method is determined, a printed inspection form is generated, which serves as the official record for measurements: the operator performing the dimensional analysis manually fills out this form while measuring the required parameters of each element of the sample with a digital caliper. Upon completion of the measurement phase, the process moves into the post-processing stage, where the recorded values are verified against technical specifications outlined in engineering drawings. Additionally, the operator is required to calculate the mean value of the measurements for each dimension within the sample using a basic calculator. The post-processing phase creates a significant burden for operators in the process due to its high reliance on manual operations, which introduce delays and an increased risk of human error. Comparing measurement data with technical specifications requires operators to frequently consult engineering drawings, making the process cumbersome and inefficient. Additionally, manual calculation of mean values is repetitive and error-prone, as it requires summing multiple measurements and dividing by the total count: any input mistake necessitates repeating the entire calculation, further prolonging execution time. Internal measurements indicate that this phase alone can take up to one hour per batch, limiting overall workflow efficiency and resource allocation.

Given these challenges, the implementation of an HTR software solution was considered as a means to streamline post-processing activities, focusing on reducing operator workload and improving efficiency. In the proposed new process, once all measurements are recorded on the inspection form, the document is scanned and uploaded into the software. By extracting handwritten values, the HTR system can automatically verify compliance with technical specifications and calculate mean values, reducing the operator's workload. However, the operator remains responsible for finalizing the process by reviewing and entering the software-generated data into the system, ensuring quality control standards are maintained. In this regard, in line with recent discussions on Industry 5.0 which emphasize resilient and human-centered industrial systems over full automation [27], the software is intended as a support tool rather than a replacement for human work, with the aim of improving efficiency while preserving manual oversight when necessary. The redesigned process was tested under controlled experimental conditions rather than in live production. Consequently, the study focuses on assessing the technical performance of the HTR tool within a stable and reproducible environment. Factors such as operator training, workflow integration, and organizational acceptance–which would influence actual industrial adoption–were intentionally excluded from the scope of this evaluation.

3.1 Software preparation and process design

The HTR system used in this study is based on a patented architecture than combines neural, Bayesian, and graph-based learning. Isolated characters are classified by a multi-expert neural network ensemble–20 feed-forward backpropagation networks trained on geometric moments and pixel statistics. Outputs are fused via a Bayesian Network that infers posterior probabilities through supervised learning, implementing a probabilistic weighted voting scheme. For cursive fragments, the system reconstructs the writing order using a semi-Eulerian graph and modified Fleury's algorithm to unfold strokes. Each stroke is described by a chain code, and similarity is computed via a weighted edit distance (WED) across multiple scales. Recognition integrates these stroke comparisons with a Dynamic Lexicon derived from known words and their expected stroke counts. The architecture blends pattern recognition, probabilistic inference, and lexical constraint modeling, providing robust, language-flexible handwriting recognition from static images without deep CNN-style end-to-end training.

The software underwent several preliminary setup stages to ensure its capability to process the forms used in this case study. First, the form templates—blank forms—were uploaded into the software, and the specific fields that contain the handwritten values to be extracted were marked. This step allowed the system to correctly identify the relevant data when scanning completed forms while ignoring unnecessary content. The software achieves this by recognizing reference points, such as printed headings or table frames, which enable it to locate the reading areas even when minor discrepancies arise between the original template and the scanned forms, such as variations in zoom, alignment or minor scanning imperfections. The subsequent stage involved integrating the software with the technical specifications of all the materials being inspected. Each form is intended for an inspection method, but the same inspection method is used for different materials, or product key. Thus, the system was set up to enable the choice, upon form submission, of both a form and a product key. In this way it can automatically evaluate read measurements against established specifications, flagging any non-conforming value. This eliminates the need for operators to manually consult technical drawings during inspections, streamlining the workflow and minimizing cognitive load, thereby enhancing focus during the verification process. Finally, a training phase was conducted to enhance the software's ability to accurately interpret handwritten text within the specific document structure used in this process. To do so, the software provider was supplied with a dataset of manually completed forms containing fictitious data, sufficient to allow the system to learn to recognize human handwriting within the specific document structure used in this process.

The newly designed inspection process integrates the software to streamline workflow execution, improve accuracy, and facilitate faster data validation. Thus, it follows a structured sequence of steps, which are:

1. Scanning the completed form–The software processes scanned documents as input, requiring a high-quality image to ensure optimal recognition, so the first phase is to create the digital image.
2. Uploading the digital file–The user uploads the scanned document and selects the corresponding template, to ensure the correct reading areas are identified, and product key, to individuate the correct specifications.
3. Verification of form registration–The system detects the field boundaries and displays them through the user interface. The operator visually checks the correct identification of reading areas and confirms it before proceeding. A detail of the user interface is shown in Fig. 3.
4. Reading phase–The operator initiates the recognition process, after which the software generates an annotated version of the scanned document. Each designated cell is marked with two key values: the extracted handwritten value and a confidence score (ranging from 0 to 1), indicating the system's certainty in its interpretation. This score is critical in identifying potential misreadings. Details of the output can be seen in Fig. 4.
5. Correction phase–Since an entirely error-free reading is unrealistic, an operator reviews and corrects any misinterpreted values. While modifying a single value is quick, a high number of errors can impact efficiency, making it crucial to maintain a low error rate.
6. Final validation and automatic calculations–Once reading is verified and confirmed by the operator, the software flags any non-conforming values and automatically calculates the mean measurement, reducing the need for manual computations.

Fig. 3 Detail of user interface with identification of reading areas

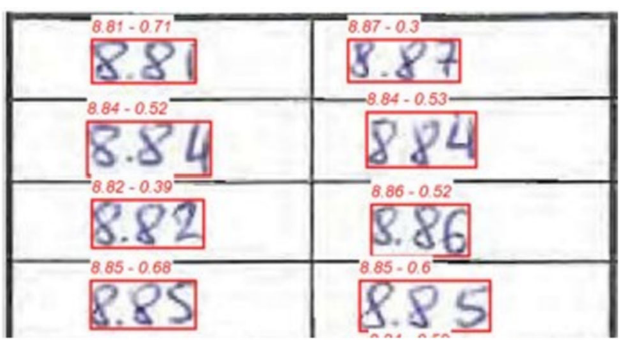


Fig. 4 Detail of the software output

This structured process ensures a balanced integration of automation and human oversight, enhancing quality control operations while maintaining accuracy and reliability.

3.2 Methodology

To assess the impact of HTR software on the inspection process of the study, a test phase was conducted. The evaluation focused on the software's accuracy, processing time, and error-handling efficiency. The tests were performed using three different inspection form types, i.e. three inspection methods, each corresponding to a distinct primary packaging material. For each material, two different product keys were considered, resulting in a total of six product keys being tested. The three inspection form types used were selected by the process owner as representative of the dimensional inspection procedures routinely applied to incoming materials. These forms share an identical structure and differ only in the number of recorded measurements, reflecting minor variations in the corresponding inspection methods. Since all forms used in practice follow the same layout and data recording conventions, the selection does not introduce bias and can be considered representative of the overall process. Table 1 summarizes the materials included in the study.

Table 1 Materials and product key included in the study

Form type	Material	Product keys
17	Ring	A, B
18	Cap	C, D
19	Bottle	E, F

Table 2 Forms in the test set with relative types and product keys

Form number	Form type	Product key
1	17	A
2	19	F
3	18	C
4	17	A
5	18	D
6	19	F
7	19	E
8	18	C
9	17	A
10	17	A
11	18	D
12	19	F
13	17	B
14	17	B
15	19	E
16	19	F
17	18	D
18	18	C
19	17	B
20	18	D

Each form is structured to assess a sample of 50 units. Forms 17 and 18, used for inspecting rings and caps, necessitate four measurements per unit, leading to a total of 200 recorded values per form. In contrast, Form 19—for bottle inspection—necessitates five measurements per unit, resulting in 250 recorded values per form. In Table 2 the complete description of the test set can be found.

All forms in the test set were completed by a limited number of operators using fictitious data, ensuring consistency in evaluation. Although writer-specific effects were not analyzed in this study, individual handwriting styles may have influenced recognition performance: this aspect represents an important consideration for interpreting the results and for future investigations. The software was tested following the exact operational procedure outlined in the previous section. To quantify performance in terms of time efficiency, the processing workflow was divided into three primary phases:

1. Scanning and uploading–Including entering the corresponding form template and product key and verifying correct registration.
2. Automated reading–Performed solely by the software, assessing its ability to extract values accurately.
3. Correction and validation–Encompassing manual review, correction of extracted values, and the final validation of flagged measurements.

By segmenting the evaluation into these phases, the study aimed to determine the extent of efficiency gains while identifying potential challenges related to software accuracy and human intervention requirements.

4 Results

The following section presents the results obtained from the evaluation of the HTR software performance in processing modules from the test set, focusing on data recognition accuracy and processing time. The assessment aims not only to determine the extent to which the software can enhance efficiency in the inspection process in the case study, but also to provide, with a broader point of view, insights into the maturity and feasibility of HTR technology in industrial applications.

4.1 Data recognition

The primary and most critical analysis assesses the software's ability to recognize handwritten text, as this represents the core function of HTR technology. In the examined case study, numerical values consisting of three or four digits are recorded, and a recognition error has been defined as any instance in which one or more digits within a value are misinterpreted. This determination has been made since a mistake in even a single one digit requires manual correction, so the accurate recognition of the remaining digits within the same cell becomes irrelevant from an operational standpoint. Despite this, a complementary evaluation of digit recognition accuracy has also been conducted. Although this aspect is not directly impactful for process efficiency, it provides valuable insights into the maturity of HTR technology, helping to understand its fundamental capabilities and areas for improvement. Furthermore, confidence values have been analyzed to assess the software's ability to recognize instances where it may have misread a value, providing insights into its self-awareness and capacity for error detection.

Table 3 Percentage of errors of the test set forms

Form number	N. of values	N. of errors	Percentage of errors (%)
1	200	68	34.00
2	250	70	28.80
3	200	126	63.00
4	200	136	68.00
5	200	57	28.50
6	250	62	24.80
7	250	69	27.60
8	200	26	13.00
9	200	61	30.50
10	200	47	23.50
11	200	27	13.50
12	250	70	28.00
13	200	67	33.50
14	200	72	36.00
15	250	72	28.80
16	250	57	22.80
17	200	57	28.50
18	200	66	33.00
19	200	68	34.00
20	200	35	17.50

4.1.1 Values

As anticipated, the number of numerical values per form varies depending on the form type: Form types 17 and 18 contain 200 values each, whereas Form type 19 contains 250 values. However, as the recognition rate can be expressed in percentage form, it is possible to analyze the results collectively, ensuring a comprehensive evaluation of the software's performance across different form structures. Table 3 shows errors number and percentage for each form within the test set, while the main statistics are reported in Table 4 and represented through a boxplot in Fig. 5.

Two modules (Test 3 and Test 4) exhibit error rates particularly high and can be identified as outliers according to the definition of threshold as the third quartile plus 1.5 times the interquartile range (IQR). These outliers were analyzed separately and excluded from the overall distribution, determining the statistics to change in the ones shown in Table 5.

The brief analysis of the outliers revealed that the anomalies were primarily due to specific handwriting characteristics of the operators. Certain digits were consistently written in a non-standard way that made recognition difficult for the

Table 4 Percentage of errors statistics

Maximum	Mininum	Mean	St. Dev	Median	Q1	Q3
68.000%	13.000%	30.825%	13.509%	28.500%	24.480%	33.632%

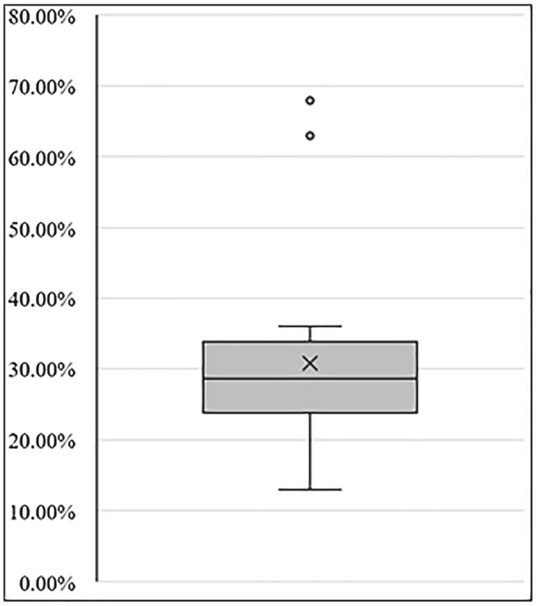


Fig. 5 Boxplot of percentage of errors

Table 6 Percentage of errors on digits of the test set forms

Form number	N. of digits	Percentage of errors (%)
1	700	10.57
2	900	9.67
3	700	21.29
4	700	26.29
5	700	8.14
6	900	7.58
7	950	8.00
8	700	4.43
9	700	10.20
10	700	8.31
11	700	3.71
12	900	9.33
13	700	11.14
14	700	12.86
15	950	8.53
16	900	7.68
17	700	8.86
18	700	10.60
19	700	12.63
20	700	4.72

system and led to systematic misinterpretation. This observation suggests that recognition performance may heavily vary according to individual handwriting styles, highlighting the potential influence of writer-specific factors on HTR accuracy.

4.1.2 Digits

As anticipated, an additional analysis was conducted at the individual digit level to further evaluate the system's recognition capabilities and provide a more detailed and accurate measure of its performance. Since this approach accounts for correctly identified digits even within misread values, it is expected that the percentage of recognition errors will be lower than when considering full numerical values, done in the previous section. Table 6 shows errors percentage on digit reading for each form within the test set, where the total number of digits in each form depends on the form type and the associated product key. The main statistics are reported in Table 7 and represented through a boxplot in Fig. 6.

Computing the threshold as per the value analysis, forms 3 (21.17%) and 4 (26.29%) can be considered outliers. These are the same tests that were identified as outliers when analyzing errors on full numerical values. By excluding these outliers from the calculations, the refined statistics are presented in Table 8.

Lastly, an additional analysis was conducted to determine whether digit recognition accuracy is influenced by the digit's position within the numerical value. This decision was prompted by empirical observations, indicating a higher frequency of misread digits in the final position, whereas digits in the initial positions were typically recognized correctly. To investigate this pattern, values consisting of three and four

Table 5 Percentage of errors corrected statistics

Maximum	Mininum	Mean	St. Dev	Median
36.000%	13.000%	26.882%	6.984%	28.250%

Table 7 Percentage of errors on digits statistics

Maximum	Mininum	Mean	St. Dev	Median	Q1	Q3
26.290%	3.710%	10.227%	5.162%	9.095%	7.920%	10.735%

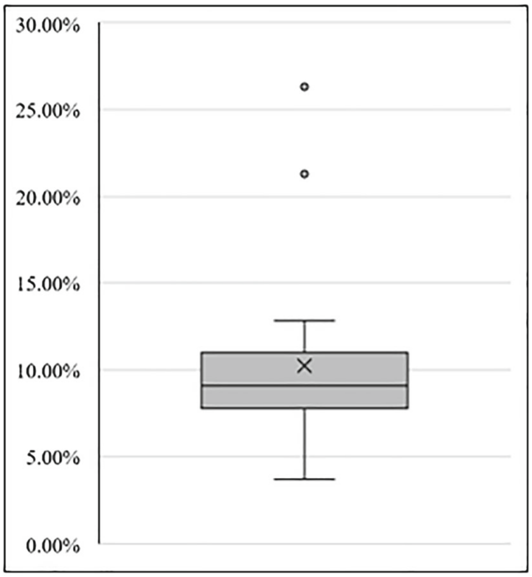


Fig. 6 Boxplot of percentage of errors on digits

Table 8 Percentage of errors on digits corrected statistics

Maximum	Mininum	Mean	St. Dev	Median
12.860%	3.710%	8.720%	2.491%	8.695%

digits were analyzed separately, starting with three-digit values, whose percentages are shown in Table 9 and summarized in Table 10.

An analysis of variance (ANOVA) was performed to confirm the influence of digit position on error rate, suggested by resulting statistics. The resulting p-value (1.554×10^{-10}) strongly indicates that the assumption of invariance—where any observed differences in recognition performance would be attributed solely to random variation within the sample —can be rejected, as it is lower than the significancy threshold ($\alpha = 0.05$).

Same analysis has been conducted on 4-digits values and are coherently reported in Tables 11 and 12.

Again, an ANOVA was conducted and resulted in a p-value of 2.168×10^{-18}, which allows to reject the invariance hypothesis.

4.1.3 Confidence

A key factor in assessing the feasibility of the tool for the practical application is the confidence score assigned to each recognized value. This metric, given as a value between 0 and 1, quantifies the system's certainty in its interpretation, and if well correlated with recognition performance, can help flag potential errors, directing operator intervention to uncertain readings. Table 13 shows, for each form of the data set, the percentage of errors on the values, already analyzed, and the mean confidence within the form.

As an initial verification, the statistical correlation between error percentage and average confidence score was analyzed. If the system accurately evaluates its own performance, trials with higher error rates should correspond to lower average confidence scores, resulting in a negative correlation. The Pearson correlation coefficient was calculated with the following results:

- Including outliers (Test 3 and Test 4, previously discussed):
  - $r = -0.591$
  - $r^2 = 0.350$
- Excluding outliers:
  - $r = -0.738$
  - $r^2 = 0.544$

These results indicate a moderate negative correlation. The scatter plot illustrating this correlation is shown in Fig. 7.

To further investigate the relationship between confidence scores and recognition accuracy, all values processed by the software across the entire test set were analyzed, without differentiation by module, product, or individual trial. The test Set consists of 20 total forms—14 with 200 values and 6 with 250 values—for a total of 4300 values, of which:

- Correctly recognized values: 2988
- Misread values: 1312

This results in an overall error rate of 30.51%, confirming previous findings. The correctly recognized values have an average confidence score of 0.459 with a standard deviation of 0.162, while the misread values have an average confidence score of 0.358 with a standard deviation of 0.109. The distributions of confidence scores for correctly and incorrectly recognized values are shown in Fig. 8.

An ANOVA was conducted to statistically assess the rejectability of the invariance hypothesis, which in this case states that the confidence scores for correctly and incorrectly

Table 9 Percentage of errors on digits in different positions for 3-digits values

Form number	N. of 3-digits values	Percentage of errors on 1st digit (%)	Percentage of errors on 2nd digit (%)	Percentage of errors on 3rd digit (%)
1	100	0.00	5.00	24.00
2	100	2.00	7.00	16.00
3	100	0.00	25.00	48.00
4	100	2.00	27.00	61.00
5	100	0.00	4.00	23.00
6	100	0.00	8.00	18.00
7	50	0.00	18.00	16.00
8	100	0.00	1.00	9.00
9	100	0.00	18.00	26.00
10	100	0.00	9.00	20.00
11	100	0.00	2.00	13.00
12	100	0.00	4.00	12.00
13	100	3.00	8.00	23.00
14	100	0.00	7.00	34.00
15	50	0.00	16.00	22.00
16	100	3.00	8.00	17.00
17	100	0.00	11.00	18.00
18	100	0.00	12.00	26.00
19	100	9.00	22.00	24.00
20	100	0.00	4.00	14.00

Table 10 Statistics of percentage of errors on digits in different positions for 3-digits values

	Percentage of errors on 1st digit (%)	Percentage of errors on 2nd digit (%)	Percentage of errors on 3rd digit (%)
Mean	0.95	10.80	23.20
St. Dev	2.11	7.49	12.04
Mininum	0.00	1.00	9.00
Median	0.00	8.00	21.00
Maximum	9.00	27.00	61.00

recognized values follow the same distribution, meaning the software does not accurately evaluate its performance, and the observed trend is random. The returned *p*-value is of 2.79 $\times 10^{-90}$, which is significantly below the significance threshold, allowing us to reject the invariance hypothesis.

4.2 Processing time

Processing time is a key factor in assessing the feasibility of integrating HTR into the inspection process, particularly to evaluate the potential impact in the operation object of the case study of the analyzed software.

The evaluation of processing time was conducted by measuring the duration required to complete each step of the workflow, following the methodology outlined in Sect. 3.2. The first phase of processing, scanning and uploading, is the simplest and least significant in terms of performance analysis, as it occurs prior to the actual utilization of the software. The second step, reading phase, is performed by the software and gives indications about the time required to perform the recognition, which is the main function of the technology. Lastly, the correction phase is done by the operator, and clearly requires a time proportional to the accuracy of the reading.

The analysis considers both 200-value forms and 250-value forms separately, as the number of values directly affects processing time, with larger datasets requiring more computational resources and potentially influencing recognition speed and efficiency.

Table 11 Percentage of errors on digits in different positions for 4-digits values

Form number	N. of 3-digits values	Percentage of errors on 1st digit (%)	Percentage of errors on 2nd digit (%)	Percentage of errors on 3rd digit (%)	Percentage of errors on 4th digit (%)
1	100	0.00	0.00	9.00	36.00
2	150	0.67	5.33	14.00	21.33
3	100	0.00	1.00	28.00	47.00
4	100	0.00	3.00	26.00	65.00
5	100	0.00	0.00	3.00	27.00
6	150	0.00	1.33	12.00	16.67
7	200	0.50	1.00	8.50	19.50
8	100	0.00	0.00	4.00	17.00
9	100	0.00	0.00	7.00	24.00
10	100	0.00	1.00	8.00	22.00
11	100	0.00	0.00	1.00	10.00
12	150	0.00	4.00	18.67	22.67
13	100	0.00	3.00	8.00	33.00
14	100	0.00	6.00	14.00	29.00
15	200	1.50	1.50	6.00	22.00
16	150	0.00	4.67	11.33	12.67
17	100	0.00	0.00	7.00	26.00
18	100	0.00	0.00	5.00	33.00
19	100	0.00	3.00	6.00	27.00
20	100	0.00	0.00	4.00	12.00

Table 12 Statistics of percentage of errors on digits in different positions for 4-digits values

	Percentage of errors on 1st digit (%)	Percentage of errors on 2nd digit (%)	Percentage of errors on 3rd digit (%)	Percentage of errors on 4th digit (%)
Mean	0.13	1.74	10.03	26.14
St. Dev	0.36	1.94	7.00	12.41
Mininum	0.00	0.00	1.00	10.00
Median	0.00	1.00	8.00	23.34
Maximum	1.50	6.00	28.00	65.00

4.2.1 200-values forms

This section focuses on Form types 17 and 18, which contain 200 values per form. Table 14 reports the times required for each step in the processing of each form in the test set, while Table 15 summarizes them showing the main statistics. Most interesting are the values about the total processing time, given by the sum of the three stages. For each form, data are showed in Table 16, and general statistics are visible in Table 17.

Computing the threshold again as 1.5 times IQR above Q3 and 1.5 times IQR below Q1, a single outlier can be identified, i.e. form 4, which processing time is highly influenced by the reading time. Form 4 was also identified as one of the two outliers in recognition accuracy. Removing it from the statistics computation, new values are obtained and reported in Table 18.

This metric is essential as it facilitates a direct comparison between the software-assisted process and the current manual method. The discussion section will address this comparison.

4.2.2 250-values forms

This section focuses on Form 19, which contains 250 values per form. Since these forms require a greater number of recorded values than Forms 17 and 18, processing time is expected to be proportionally higher. Table 19 reports the time required for each step of processing each form in the test

Table 13 Percentage of errors and mean confidence of the forms in the test set

Form number	Percentage of errors (%)	Mean confidence
1	34.00	0.4780
2	28.80	0.4190
3	63.00	0.3912
4	68.00	0.3597
5	28.50	0.5175
6	24.80	0.4428
7	27.60	0.3801
8	13.00	0.5463
9	30.50	0.3456
10	23.50	0.4273
11	13.50	0.5331
12	28.00	0.4070
13	33.50	0.4340
14	36.00	0.3580
15	28.80	0.4184
16	22.80	0.4400
17	28.50	0.4298
18	33.00	0.3416
19	34,00	0.3792
20	17.50	0.5363

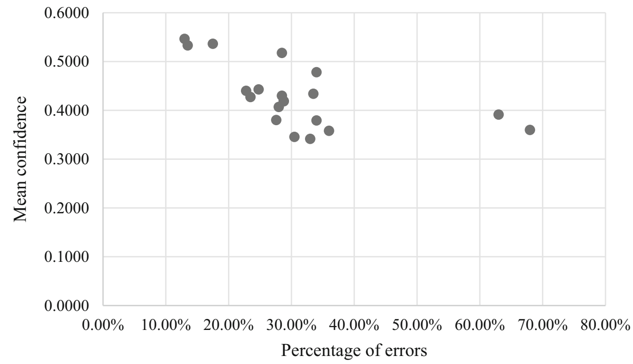


Fig. 7 Scatterplot of mean confidence against percentage of errors of the forms in the test set

set, while Table 20 provides a statistical summary of these values.

As in the previous section, the most relevant metric is the total processing time; it is shown in Table 21 and summarized in Table 22.

Given that there are only six observations for the 250-value forms, it is challenging to draw definitive conclusions about outliers applying statistical methods such as IQR-based thresholding with high reliability. However, the available data still provides useful insights into processing time trends.

5 Discussion

This section discusses and interprets the key results presented in the previous section, drawing conclusions about the strengths and limitations of HTR in the investigated practical case study; additionally, it provides insights for the broader objective of evaluating HTR's feasibility in industrial settings. The discussion is structured around the research questions formulated in the introduction, aiming to provide a comprehensive answer to each in light of the results obtained.

5.1 Recognition accuracy

Answering RQ1, which investigates how accurately the HTR software can recognize and process handwritten numerical

Fig. 8 Distribution of confidence scores for correctly and incorrectly recognized values

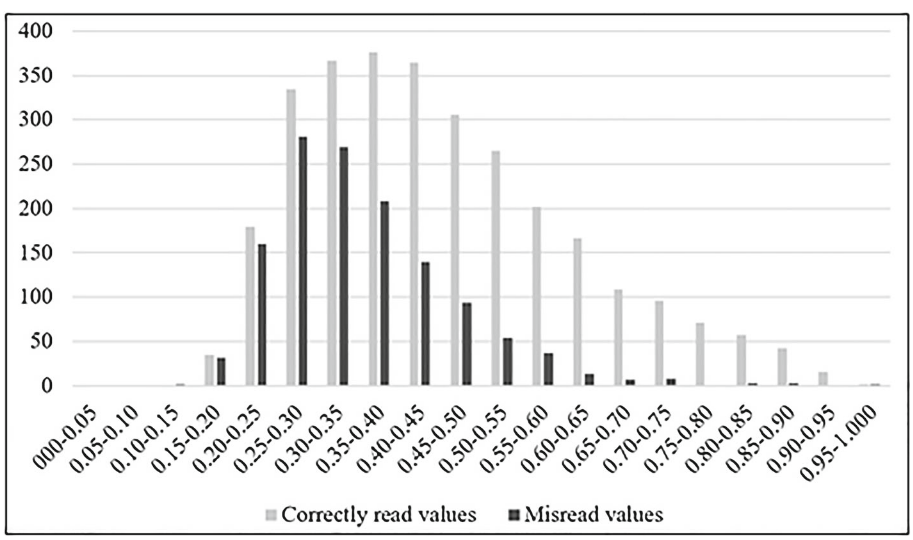


Table 14 Processing times for 200-values forms of the test set

Form number	Scanning and uploading [s]	Automated reading [s]	Correction and validation [s]
1	150	270	500
3	150	270	670
4	150	910	570
5	160	200	380
8	150	170	300
9	150	225	370
10	150	380	340
11	150	130	240
13	115	426	335
14	155	124	396
17	135	315	310
18	125	265	320
19	136	255	390
20	125	155	285

Table 15 Statistics of processing times for 200-values forms of the test set

	Scanning and uploading [s]	Automated reading [s]	Correction and validation [s]
Max	160	910	670
Min	115	124	240
Mean	143	293	386
Median	150	260	355
St. Dev	13	191	114

Table 16 Total processing time for 200-values forms in the test set

Form number	Total [s]	Total
1	920	15 min 20 s
3	1090	18 min 10 s
4	1630	27 min 10 s
5	740	12 min 20 s
8	620	10 min 20 s
9	745	12 min 25 s
10	870	14 min 30 s
11	520	8 min 40 s
13	876	14 min 36 s
14	675	11 min 15 s
17	760	12 min 40 s
18	710	11 min 50 s
19	781	13 min 01 s
20	565	9 min 25 s

Table 17 Statistics of total processing time for 200-values forms in the test set

	Total [s]	Total
Max	1630	27 min 10 s
Min	520	8 min 40 s
Mean	822	13 min 42 s
Median	753	12 min 33 s
Q1	684	11 min 24 s
Q2	875	14 min 35 s
St. Dev	266	4 min 26 s

Table 18 - Corrected statistics of total processing time for 200-values forms in the test set

	Max	Min	Mean	Median	St. Dev
Total [s]	1090	520	759	745	148
Total	18 min 10 s	8 min 40 s	12 min 39 s	12 min 25 s	2 min 28 s

Table 19 Processing times for 250-values forms of the test set

Form number	Scanning and uploading [s]	Automated reading [s]	Correction and validation [s]
2	150	280	550
6	165	215	460
7	150	240	510
12	150	340	420
15	125	380	440
16	125	230	390

Table 20 Statistics of processing times for 250-values forms of the test set

	Scanning and uploading [s]	Automated reading [s]	Correction and validation [s]
Max	165	380	550
Min	125	215	390
Mean	144	281	462
Median	150	260	450
St. Dev	15	60	54

Table 21 Total processing time for 250-values forms in the test set

Form number	Total [s]	Total
2	980	16 min 20 s
6	840	14 min 00 s
7	900	15 min 00 s
12	910	15 min 10 s
15	945	15 min 45 s
16	745	12 min 25 s

Table 22 Statistics of total processing time for 250-values forms in the test set

	Total [s]	Total
Max	980	16 min 20 s
Min	745	12 min 25 s
Mean	887	14 min 47 s
Median	905	15 min 05 s
St. Dev	76	1 min 16 s

data, the system demonstrated a good level of accuracy at the digit level, with a mean error rate of 8.72% and a standard deviation of 2.49%, consistently with similar research. For instance, [28] developed a convolutional-network model for handwritten digits extracted from digitized administrative documents, achieving 85% accuracy under real operational conditions. Likewise, [29] reported a progressive enhancement of a CRNN architecture for handwritten data on industrial acquisition sheets, improving its recognition rate from 0.693 to 0.857 after dataset augmentation. In this context, the results obtained in our study indicate that the system's digit recognition performance lies within a comparable range, confirming its technological soundness and suggesting that the core recognition mechanism is robust and aligned with the state of the art. Moreover, an analysis of variance suggested that the software performs better in recognizing digits in the first position. In this regard, it is necessary to note that the software has access to the technical specifications of the materials, which may influence its performance: in most cases, within the specified range, the first digit remains unchanged, potentially contributing to a higher accuracy in recognizing the leading digits. The observed behavior necessitates further examination to determine whether the software is interpreting these digits or simply relying on specifications to infer them.

From an operational perspective, the recognition of entire numerical values provides a more meaningful indicator of usability. Even a single misread digit renders a value incorrect and requires manual verification or correction by the operator. Consequently, while the digit recognition accuracy can be deemed satisfactory, the observed error rate on complete values ($\approx 27\%$) still demands substantial human intervention, which significantly limits the potential for full automation in industrial workflows. This limitation becomes clearer when compared with similar systems deployed in industrial environments, where automation standards are considerably higher. [30] developed an OCR system for tracking printed coil identifiers in a Brazilian steel plant, achieving over 98% accuracy and demonstrating that even minor identification errors could cause costly production stoppages or misdeliveries. [31] introduced an AI-based OCR system for on-site text recognition on iron plates, reaching F1 accuracies above 92% after advanced data augmentation and model integration. Although these OCR solutions rely on printed rather than handwritten characters, they represent the industrial benchmark for automated reading tasks.

Hence, while the HTR system used in the research performs well given the inherent complexity of handwriting, its value-level accuracy remains below industrial expectations for automation-oriented digit recognition. Further progress will require not only model refinement but also domain adaptation strategies and post-processing checks to bridge the gap between technical feasibility and industrial reliability.

About the ability to detect uncertain readings, the correlation between confidence scores and recognition errors was moderate but significant, with a Pearson correlation coefficient of -0.738 when outliers were excluded. This indicates that the software possesses a partial capability to self-assess its uncertainty. Confirmation comes from a variance analysis that allows to say that the distribution of confidence scores for correctly recognized values differs significantly from the one of misread values. Despite this, the substantial overlap between these distributions–where the difference between their means is roughly equal to one standard deviation–reveals that confidence variability remains too high to clearly distinguish accurate from erroneous readings. This limited precision prevents the confidence metric from serving as a reliable indicator for automated decision-making. In fact, the concept of confidence-driven adaptation represents an interesting direction for future development. The system could leverage confidence thresholds to automatically flag low-confidence readings for human verification while processing high-confidence values autonomously, thereby optimizing operator workload and overall efficiency. To enable such adaptive workflows, however, confidence estimation would first need to become significantly more precise and stable. Refining the calibration of confidence scores–through improved probabilistic modeling or uncertainty quantification–would be a necessary step toward making confidence-based selective automation feasible in industrial environments.

5.2 Processing time reduction

Subsequently, processing times have been analyzed, allowing for the discussion of RQ2, about the efficiency improvements in data processing. The manual post-processing phase, as reported by the operating team, requires an average of 60 min per module for a 200-value form. For the same kind of modules, the software-assisted process presents a processing time characterized by an average of 12 min and 39 s and a standard deviation of 2 min and 28 s. To make a clear comparison ensuring a conservative estimate, a processing time of 17 min and 36 s (mean plus two standard deviations) was adopted, covering approximately 95% of cases under the assumption of a normal distribution. With this adjustment, software utilization results in a 70.6% reduction in post-processing time. This provides essential insights into whether the introduction of HTR can yield tangible benefits in terms of efficiency gains and labor optimization in industrial environments, even if this type of analysis remains highly specific to the process under study, depending on the unique characteristics and constraints of its workflow. Moreover, an important consideration is the role of operator involvement. While in the manual process the operator is fully engaged in post-processing activities, in the software-assisted process part of the task is automated, particularly the reading phase, which requires an average of 4 min excluding the outlier. Consequently, operator saturation, defined as the ratio between operator working time and total processing time, decreases from 100 to 77.48%. Although this marks an improvement, it underscores a key limitation of HTR systems: they are neither fully autonomous nor entirely reliable, necessitating continuous human supervision to monitor and correct potential recognition errors.

5.3 Technical limitations and strategies

Finally, RQ3 concerns the key limitations and challenges associated with implementing HTR in a structured industrial workflow. The findings regarding RQ1 and RQ2 already provide a foundation for addressing this question, as they highlight both the potential benefits and the critical constraints of the technology. While the significant reduction in processing time suggests that HTR can be a valuable tool for enhancing efficiency, the high error rate–aggravated by the significant variability–limits its applicability for full automation. The necessity for manual corrections means that these kinds of tools are best suited as a support rather than a replacement for human work, which is an interesting starting point but casts doubts about the maturity of the technology for the industrial context. The persistence of recognition errors highlights the need for targeted strategies to improve model robustness. In practice, several approaches could be adopted to mitigate such issues. Data augmentation has proven particularly effective in enhancing generalization under limited data conditions [2], either through traditional transformations or synthetic handwriting generation using GANs and diffusion models. Architectural refinements, such as the integration of attention mechanisms or optimized activation functions, have also been shown to improve feature extraction and recognition accuracy [29]. Moreover, combining preprocessing strategies tailored to the document type with advanced augmentation techniques, such as tiling-corruption or morphological variations, can further increase resilience to noise and handwriting variability [32]. Although such strategies were beyond the scope of the present study, they represent realistic pathways for improving recognition accuracy in future industrial deployments.

The confidence evaluation mechanism, which could mitigate the impact of the high error rate, remains too imprecise to support adaptive workflows based on the automatic flagging of uncertain readings. While confidence-driven intervention could theoretically enhance efficiency by directing operator effort toward low-certainty outputs, achieving this would require substantial improvements in confidence calibration and reliability. It would be essential to transform this metric from a purely descriptive indicator into a truly functional component.

Finally, it is important to recognize that, as anticipated, the results were obtained under controlled experimental conditions, which isolate the technical performance of the HTR tool from the variability of real operational contexts. In practice, the successful industrial adoption of such systems would depend not only on recognition accuracy but also on human and organizational factors, including operator training, workflow integration, data governance, and overall acceptance of technological change. These aspects were beyond the scope of the present evaluation but represent critical directions for future implementation studies.

6 Conclusion

The presented practical case study explored the application of a Handwritten Text Recognition software within the operational execution of an industrial acceptance sampling inspection process, assessing its potential to improve efficiency and reduce manual workload. Within the specific conditions of this study, the tested tool demonstrated potential to reduce processing time, though recognition accuracy remains a limiting factor that would likely affect similar systems in comparable industrial tasks. The results are inherently linked to the characteristics of the specific HTR system employed–as is typical for empirical studies of this kind–and performance outcomes are strongly influenced by the software's underlying architecture. Consequently, while the observed tendencies may resemble those of other HTR tools, the quantitative results should not be interpreted as representative of the technology as a whole.

This research provides a preliminary yet meaningful contribution by systematically evaluating how HTR could support structured industrial workflows, an area where empirical research is still scarce. The findings illustrate a methodological approach for evaluating such technologies under controlled conditions and highlight the practical challenges that would need to be addressed before large-scale deployment. Beyond the specific results, the work offers a proof of concept that HTR can be effectively integrated into industrial quality control workflows to support, rather than replace, human inspection activities. In this sense, the study bridges a gap between laboratory-level AI development and practical manufacturing applications, showing how evaluation under controlled but realistic conditions can inform future implementation strategies.

The main limitation lies in its controlled experimental nature: the process was not implemented in an operational setting, and therefore human and organizational factors–such as operator interaction, workflow integration, and acceptance of new technologies–were not examined. Addressing these dimensions will be essential to evaluate the technology under real industrial conditions. Moreover, since the forms were completed by a small number of operators, the results may reflect writer-specific patterns that influenced recognition performance–as is noticed in the discussion section–but this aspect was excluded from the analysis. Lastly, despite the application of statistical adjustments, the test dataset was limited in both size and diversity, being derived from a controlled environment and a small number of operators. The limited dataset and the software-specific performance thus constitute clear constraints on the generalizability of the study. Nevertheless, the results still offer meaningful insights into the significant potential of HTR as a supportive tool for industrial applications, particularly in structured workflows where manual data entry and validation act as bottlenecks. In this sense, the study provides a valuable step toward understanding how HTR can contribute to improving efficiency while maintaining human oversight within industrial quality control processes.

Future research should focus on enhancing transcription accuracy through more advanced models. Employing a more extensive and heterogeneous dataset would therefore be essential to validate the observed trends and to provide stronger evidence of the technology's scalability and robustness across use cases. Additionally, practical implementation trials in real production settings would be essential to validate the effectiveness of the technology in live operations and address the challenges related to user adoption and process integration. Future implementation studies should also include an economic assessment of costs and benefits, which was beyond the scope of the present technical evaluation but would be essential to fully determine the industrial viability of HTR solutions.

Funding Open access funding provided by Università degli Studi di Roma Tor Vergata within the CRUI-CARE Agreement. Next Generation EU, E83C22003240001

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Consent for publication The manuscript is not currently submitted to another journal for simultaneous consideration. The submitted work is original and has not been published elsewhere. Results are presented honestly and without fabrication, falsification or inappropriate data manipulation. No data, text, or theories by others are presented as if they were the author's own: proper acknowledgements to other works are given.

References

1. AlKendi, W., Gechter, F., Heyberger, L., Guyeux, C.: Advancements and challenges in handwritten text recognition: a comprehensive survey. J. Imaging **10**(1), 18 (2024). https://doi.org/10.3390/jimaging10010018
2. De Sousa Neto, A.F., Bezerra, B.L.D., De Moura, G.C.D., Toselli, A.H.: Data augmentation for offline handwritten text recognition: a systematic literature review. SN Comput. Sci. **5**(2), 258 (2024). https://doi.org/10.1007/s42979-023-02583-6
3. Fantozzi, I.C., Delfini, L., Martuscelli, L., Antonelli, D., Schiraldi, M.M.: Enhancing production efficiency with an optimized PBOM structure: a clustering-based methodology. Prod. Eng. **20**(1), 14 (2026). https://doi.org/10.1007/s11740-025-01380-9
4. Arif, A., Rao, P.G., Prasad, K.: A novel hybrid Bayesian-optimized CNN–SVM deep learning model for real-time surface roughness classification and prediction based on in-process machined surface image analysis. Int J Interactive Des Manuf (IJIDeM) **19**(10), 6941–6958 (2025). https://doi.org/10.1007/s12008-025-02261-2
5. Gao, J., Li, J., Duan, P.: The application of machine learning and artificial intelligence technology in the production quality management of traditional Chinese medicine decoction pieces. Int J Interactive Des Manuf (IJIDeM) **18**(1), 239–251 (2024). https://doi.org/10.1007/s12008-023-01448-9
6. Jakhotiya, Y., Rahul, M.R., Chiddarwar, S.S.: Integrating digital twin and computer vision system for efficient pick-and-place operation using tecnomatix process simulate. Int J Interactive Des Manuf

(IJIDeM) **18**(10), 7429–7443 (2024). https://doi.org/10.1007/s12008-023-01679-w
7. Machacek, Z., et al.: Use of optical sensors and artificial neural networks for precision measurement and fault detection in metallurgical environments. J. Intell. Manuf. (2025). https://doi.org/10.1007/s10845-025-02601-9
8. Khandokar, I., Hasan, M., Ernawan, F., Islam, S., Kabir, M.N.: Handwritten character recognition using convolutional neural network. J. Phys. Conference Series.1918 (4) 042152 (2021) https://doi.org/10.1088/1742-6596/1918/4/042152.
9. Odeh, A., Odeh, M., Odeh, H., Odeh, N.: Hand-written text recognition methods: review study. Revue d'Intelligence Artificielle **36**(2), 333–339 (2022). https://doi.org/10.18280/ria.360219
10. Karthick, K., Premkumar, M., Manikandan, R., Cristin, R.: Study on diverse automatic identification techniques. Int. J. Eng. Technol. **7**(4), 2895–2898 (2018). https://doi.org/10.14419/ijet.v7i4.15311
11. Memon, J., Sami, M., Khan, R.A., Uddin, M.: Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR). IEEE Access **8**, 142642–142668 (2020). https://doi.org/10.1109/ACCESS.2020.3012542
12. Ritonga, M., et al.: Optimized convolutional neural network deep learning for Arabian handwritten text recognition. Bull. Electr. Eng. Inform. **14**(2), 1497–1506 (2025). https://doi.org/10.11591/eei.v14i2.7696
13. Vinciarelli, A.: Offline cursive handwriting: from word to text recognition. IDIAP Research Report, (2003)
14. Balaha, H.M., Ali, H.A., Saraya, M., Badawy, M.: A new Arabic handwritten character recognition deep learning system (AHCR-DLS). Neural Comput. Appl. **33**(11), 6325–6367 (2021). https://doi.org/10.1007/s00521-020-05397-2
15. Momeni, S., Babaali, B.: Arabic offline handwritten text recognition with transformers. 21 (2022) https://doi.org/10.21203/rs.3.rs-2300065/v1.
16. Abdallah, A., Hamada, M., Nurseitov, D.: Attention-based fully gated CNN-BGRU for Russian handwritten text. J. Imaging **6**(12), 141 (2020). https://doi.org/10.3390/jimaging6120141
17. Kumari, L., Singh, S., Rathore, V., Sharma A.: A Lexicon and depth-wise separable convolution based handwritten text recognition system. (2022) *arXiv*. https://doi.org/10.48550/ARXIV.2207.04651.
18. Tsochatzidis, L., Symeonidis, S., Papazoglou, A., Pratikakis, I.: HTR for Greek historical handwritten documents. J. Imaging **7**(12), 260 (2021). https://doi.org/10.3390/jimaging7120260
19. Terras, M.: The role of the library when computers can read: critically adopting handwritten text recognition (HTR) technologies to support research. The Rise of AI (2022)
20. Ghosh, R., Panda, C., Kumar, P.: Handwritten text recognition in bank cheques. In 2018 Conference on Information and Communication Technology (CICT), Jabalpur, India: IEEE, ott. (2018) 1–6. https://doi.org/10.1109/INFOCOMTECH.2018.8722420.
21. Mhalgi, S., Ganu, K., Marne, P., Phadke, R., Shekapure, S.: Handwritten character recognition using neural networks forBanking applications. Int. J. Comput. Appl. **176**(27), 1–7 (2020). https://doi.org/10.5120/ijca2020920296
22. Agrawal, P., et al.: Automated bank cheque verification using image processing and deep learning methods. Multimed. Tools Appl. **80**(4), 5319–5350 (2021). https://doi.org/10.1007/s11042-020-09818-1
23. Shaw, U., Mamgai, R., Malhotra, I.: Medical handwritten prescription recognition and information retrieval using neural network. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India: IEEE, ott. (2021) 46–50. https://doi.org/10.1109/ISPCC53510.2021.9609390.
24. Tabassum, S., et al.: An online cursive handwritten medical words recognition system for busy doctors in developing countries for ensuring efficient healthcare service delivery. Sci. Rep. **12**(1), 3601 (2022). https://doi.org/10.1038/s41598-022-07571-z
25. Dinh *et al.*, M.N.: A vietnamese handwritten text recognition pipeline for tetanus medical records. In presentato al Forty-Fourth International Conference on Information Systems, (2023)
26. Couture, B., Verret, F., Gohier, M., Deslandres, D.: The challenges of HTR model training: feedback from the project Donner le gout de l'archive a l'ere numerique. J Data Min Dig Humanities (2023). https://doi.org/10.46298/jdmdh.10542
27. Gianassi, M., Leoni, L., Fantozzi, I.C., De Carlo, F., Tucci, M.: Mixed-model and multi-model assembly lines: a systematic literature review on resource management. J. Manuf. Syst. **82**, 632–657 (2025). https://doi.org/10.1016/j.jmsy.2025.07.005
28. Mykolaivna Boliubash, N., Nykyforovych Yevtushenko, O.: Automating digitized document processing with handwritten digits in the public sector using convolutional neural networks. Interdiscip. J. Inf. Knowl. Manag. **20**, 019 (2025). https://doi.org/10.28945/5527
29. Liu *et al*., W.: Handwritten text recognition of imagery acquisition sheet in tobacco industry using an improved CRNN network. In Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition, Xiamen China: ACM, set. (2022) 649–656. https://doi.org/10.1145/3573942.3574075.
30. Caldeira, T., Ciarelli, P.M., Neto, G.A.: Industrial optical character recognition system in printing quality control of hot-rolled coils identification. J. Control Autom. Electr. Syst. **31**(1), 108–118 (2020). https://doi.org/10.1007/s40313-019-00551-1
31. Tang, Q., Lee, Y., Jung, H.: The industrial application of artificial intelligence-based optical character recognition in modern manufacturing innovations. Sustainability **16**(5), 2161 (2024). https://doi.org/10.3390/su16052161
32. Omidi, Z., BabaAli, B.: On usage of an end-to-end deep neural architecture for handwritten digit string recognition. Signal Image Video Process. **18**(4), 3009–3020 (2024). https://doi.org/10.1007/s11760-023-02966-5