A Comprehensive Framework for Industrial Sticker Information Recognition Using Advanced OCR and Object Detection Techniques

Gabriella Monteiro *, Leonardo Camelo, Gustavo Aquino, Rubens de A. Fernandes, Raimundo Gomes, André Printes, Israel Torné, Heitor Silva, Jozias Oliveira and Carlos Figueiredo

Embedded Systems Laboratory, State University of Amazonas, Manaus 69050-020, Brazil; leoyutocamelo@gmail.com (L.C.); gustavoaqui@gmail.com (G.A.); rubens.eng.elet@gmail.com (R.d.A.F.); rclaudio.gomes@gmail.com (R.G.); aprintes@uea.edu.br (A.P.); itorne@uea.edu.br (I.T.); heitor.lns@outlook.com (H.S.); jpoliveira@uea.edu.br (J.O.); cfigueiredo@uea.edu.br (C.F.)
* Correspondence: grm.eng19@uea.edu.br

Abstract: Recent advancements in Artificial Intelligence (AI), deep learning (DL), and computer vision have revolutionized various industrial processes through image classification and object detection. State-of-the-art Optical Character Recognition (OCR) and object detection (OD) technologies, such as YOLO and PaddleOCR, have emerged as powerful solutions for addressing challenges in recognizing textual and non-textual information on printed stickers. However, a well-established framework integrating these cutting-edge technologies for industrial applications still needs to be discovered. In this paper, we propose an innovative framework that combines advanced OCR and OD techniques to automate visual inspection processes in an industrial context. Our primary contribution is a comprehensive framework adept at detecting and recognizing textual and non-textual information on printed stickers within a company, harnessing the latest AI tools and technologies for sticker information recognition. Our experiments reveal an overall macro accuracy of 0.88 for sticker OCR across three distinct patterns. Furthermore, the proposed system goes beyond traditional Printed Character Recognition (PCR) by extracting supplementary information, such as barcodes and QR codes present in the image, significantly streamlining industrial workflows and minimizing manual labor demands.

Keywords: optical character recognition; sticker pattern; deep learning; object detection; YOLO; manufacture automation; PaddleOCR

1. Introduction

Recent advancements in Artificial Intelligence (AI) have paved the way for disruptive innovations, offering effective solutions across a wide range of industries [1]. Many of these innovations harness the power of deep learning (DL) and computer vision algorithms to enhance industrial processes [2]. DL algorithms are capable of learning the most relevant features for distinguishing patterns and can be employed for classification, regression, or other purposes [3–5]. Meanwhile, computer vision can be applied to tasks such as image classification and character recognition [3,6–8].

Object detection (OD) involves the identification and tracking of desired object instances in images, videos, or real-time applications [7]. A popular technology employed for OD in industrial applications is YOLO (You Only Look Once) [9]. YOLO facilitates real-time object detection by dividing the input image into a grid and predicting bounding boxes (BB) and class probabilities for each grid cell [9]. This process requires a single forward pass through the network, ensuring fast and efficient object detection [7].

Another widespread application of computer vision in industry is Optical Character Recognition (OCR) [10]. Contemporary OCR techniques are primarily rooted in deep learning and aim to recognize various characters or numerals in images [10]. These techniques have been increasingly effective, even for complex scripts such as Urdu [11]. OCR technologies can identify characters in different types of documents, such as printed stickers, and can be utilized for Printed Character Recognition (PCR) and Handwritten Character Recognition (HCR) [12]. Recent advancements in HCR have used deep learning for recognizing handwritten words in documents, significantly improving accuracy in this area [13]. PCR is comparatively simpler than HCR due to the limited number of available fonts in PCR as opposed to the diverse handwriting styles found in HCR [14]. However, it is often necessary to identify documents or printed stickers containing arbitrary characters, such as hexadecimal identifiers or passwords with special characters [14].

PaddleOCR (PP-OCR) is a versatile and powerful solution for PCR and HCR, effectively addressing these challenges [15]. This tool offers a comprehensive OCR system comprising three components: text detection, detection boxes, and text recognition [15]. Text detection identifies text segments, detection boxes correct text orientation, and text recognition classifies each character present in a text segment [15,16].

Industrial applications frequently face challenges related to non-textual and textual information recognition. These challenges can encompass variations in lighting conditions, low-resolution images, and complex image patterns, all of which can negatively impact the performance of OD and OCR systems. To tackle this issue, many industries rely on human visual inspection (VI) for information recognition. However, VI comes with its own set of problems, such as visual fatigue and inconsistent results.

Despite the advancements in sticker textual and non-textual information detection, a well-established framework is yet to be developed. OD integrated with OCR technologies can be harnessed to create a robust industrial application in this area. In this paper, we propose an innovative framework that combines OCR and OD to automate visual inspection processes in an industrial context. Our primary contribution is a comprehensive framework that detects and recognizes both textual and non-textual information on printed stickers within a company. Our system employs the latest AI tools and technologies for OCR, such as YOLOv8 and PaddleOCRv3, and it is capable of processing printed stickers beyond traditional PCR, extracting additional information such as barcodes and QR codes present in the image. We emphasize that the proposed framework was designed to operate without human intervention. The motivation behind this study was to automate a process currently performed manually, with visual inspection, by operators in the industry. Through introducing automation to the visual inspection of modem stickers, our aim is to enhance both the speed and accuracy of this procedure.

The remainder of this paper is structured as follows: Section 2 introduces the fundamental concepts of OD and OCR, with a particular emphasis on YOLO and PP-OCR. Section 3 outlines the relevant literature. Section 4 describes the databases, the OD approach, and the proposed framework. Section 5 delves into the training outcomes and performance evaluation. Lastly, Section 6 provides concluding remarks and highlights possible avenues for future research.

2. Background

This section aims to provide basic background on the YOLO and PP-OCR technologies, and the ground to understand our proposed framework, in order to enhance the comprehension of the present study.

2.1. You Only Look Once (YOLO)

One of the most widely utilized object detection models currently is YOLO, due to its simple architecture, low complexity, easy implementation, and speed [17]. The authors of YOLO transformed the object detection problem into a regression problem instead of classification. A convolutional neural network predicts the bounding boxes as well as their

position. Since this algorithm identifies objects and their positions with the help of bounding boxes by looking at the image only once, it is named You Only Look Once [18].

The YOLO algorithm has undergone numerous versions since its release in 2016, demonstrating consistent improvements to the algorithm's performance. YOLOv8 was released in 2023 and presents various pre-trained model versions [19,20]. For the development of this study, YOLOv8n architecture is employed due to its compact size and high performance [17]. One of the key reasons for choosing this YOLO version in our framework is the need to obtain a powerful, lightweight, and state-of-the-art model that can run efficiently on a CPU. We aim to implement the proposed framework in an industrial setting where hardware resources may be limited. Therefore, YOLOv8n was selected for its high accuracy and ability to run in real time on CPU-based systems.

2.2. PaddleOCR (PP-OCR)

PP-OCR is a framework that integrates the entire process of training, inference, and deployment of OCR models and also provides pre-trained models for use in industrial environments [21]. PP-OCR offers the PP-OCR solution [15], a universal OCR system for text detection and recognition, with various models suitable for industrial implementations, including universal, ultra-light, and multilingual segmentation and recognition models.

In addition to the input and output, it is composed of three modules: text detection, frame correction, and text recognition. The text detection module is composed of a text detection model that returns the areas where text is present in the image. The frame correction module receives the text boxes detected in the previous module and corrects the irregularly shaped boxes into a rectangular format, preparing them for text recognition. The direction of the text is also corrected if it is upside down or sideways. Finally, text recognition is performed on the detected and corrected boxes. Currently, there are three versions of PP-OCR proposed by PaddlePaddle, namely, PP-OCRv1 [15], PP-OCRv2 [22], and PP-OCRv3 [16], with the latter being the most recent.

We employed PP-OCRv3 in our framework, which utilizes the same structure as PP-OCRv1 but implements new algorithms and optimizations. Specifically, the detection model is optimized based on the same model used in PP-OCR, the Differentiable Binarization (DB), while the base recognition model is replaced by the Single Visual Model for Scene Text Recognition (SVTR), which was previously the Convolutional Recurrent Neural Network (CRNN).

3. Related Work

This section will discuss OD and OCR applications. We also will cover post-processing techniques used with OCR systems.

3.1. Sticker Information Extraction

Extracting information from product labels in industrial settings is a critical task that has garnered significant attention in recent years [23]. Numerous techniques and technologies have been devised to improve the efficiency and accuracy of information extraction, including OCR systems and various image and data processing methods [24].

One such study is by Subedi et al. [25], who developed a low-cost OCR system grounded in deep learning. Their system accepted code text segment images from real products as inputs. However, their approach did not incorporate OD and failed to address the challenges of extracting information from complex sticker patterns, such as stickers with non-textual data. Another approach by Gromova and Elangovan [26] involved developing an automated system for extracting pertinent information from medication container images. Their system utilized convolutional neural networks (CNNs) for sticker extraction and distortion correction, followed by OCR. Relevant information was obtained from the recognized text using natural language processing (NLP) techniques. Nonetheless, their system did not include OD and might not be suitable for extracting information from intricate real-world scenarios. Gregory et al. [27] proposed an automatic inventory

tracking system that leverages computer vision and DL techniques. Their system locates, processes, and decodes label information, continuously updating the inventory database. Their approach consists of locating the label using an OD algorithm, decoding the barcode, and performing OCR. However, their system may struggle with the complexity of industrial product labels.

In our proposed framework, we integrate OCR with OD based on Santiago Garcia [28]. This integration enables us to accurately extract all relevant data from product stickers, including text, barcodes, and QR codes. Furthermore, our framework is capable of handling multiple sticker patterns, aiming to become a robust solution for an industrial application.

3.2. PaddleOCR System and Post-Processing Information

PP-OCR is widely used in various applications, including document analysis, and passport card recognition [28,29]. In the context of industrial visual inspection, PP-OCR offers a robust solution for extracting textual information from product stickers.

The work of Dahlberg et al. [30] demonstrated the effectiveness of PP-OCR in an industrial environment for Printed Character Recognition. Their solution detects stickers from panoramic scans of industrial plants, comparing the performance of different OCR systems, such as Tesseract OCR [31] and PP-OCR, with the latter outperforming the former.

OCR systems can recognize text with high accuracy, but there is still room for improvement, particularly with post-OCR techniques. Post-processing methods can be classified into manual and semi-automatic approaches [32]. Manual post-processing involves reviewing and correcting OCR output, while semi-automatic post-processing combines human inspection and automated resources to enhance OCR accuracy [33]. Semi-automatic techniques can be further divided into isolated word and context-dependent methods.

Isolated word post-processing analyzes the word to which OCR has been applied, while context-dependent methods consider all words in a document and analyze surrounding words to correct errors [34]. In the context of post-OCR processing, the transformer-based architecture BERT has been employed in several works, such as Kaló and Sipos [35], which introduces a technique for correcting Tesseract OCR output.

Our proposed framework employs an isolated word approach for post-processing OCR, as the target label fields are rule-based and have no interdependence. This innovation contributes to post-processing OCR techniques.

4. Methodology

In this section, we will cover the databases utilized to train the object detector models and to evaluate the OCR system. Then, we explain the OD methodology for detecting stickers, QR codes, and barcodes. We covered the Post-OCR algorithm employed to enhance the performance of the OCR system output. In metrics, we explain how each AI component of our framework was evaluated. Finally, we cover how all these systems work together in the proposed OCR framework to detect textual and non-textual sticker information.

4.1. Databases

We employed a controlled environment to capture images of three distinct modem sticker patterns, A, B, and C, provided by a private company. These patterns are illustrated in Figure 1. Each of these sticker patterns is characterized by unique textual and non-textual data, creating a diverse landscape of information.

Figure 1. Subset sticker patterns in databases. Sticker A comprises 43 images with 11 fields, 3 barcodes, and 2 QR codes. Sticker B consists of 40 images with 12 fields, 4 barcodes, and 1 QR code. Sticker C incorporates 22 images with 10 fields, 3 barcodes, and 1 QR code.

For the OD algorithms we created two databases, a sticker database and a non-textual database (for detecting barcodes and QR codes) and for validating our models. We used a third database to work as ground truth for the OCR output. We employed a hold-out technique as a validation strategy with a 70/30 training and validation proportion. For the first two databases, we mixed real sticker images and also non-sticker images to develop a system capable of detecting the desired objects with another sticker pattern. Additionally, we utilized MakeSenseAI for annotating the images due to its quick and specialized image annotation capabilities for object detection applications [36]. Next, the particular characteristics of the database used in this work will be detailed.

4.1.1. Stickers

The sticker database consists of modem images that we created with real images of three sticker patterns: A, B, and C, making some positioning and lighting modifications in the controlled environment. Additionally, we included some public sticker images available on the Internet. Although this OCR system is only validated with A, B, and C modem models, we designed the OCR detectors to be generalist and capable of detecting a new model without the need for retraining. At the end of this process, a database was obtained with 561 different modem images with 580 annotated stickers.

4.1.2. QR Code and Barcode

The non-textual (QR codes and barcodes) database consists of modem sticker images that have QR codes and barcodes captured in a controlled environment. In addition, to diversify the database, QR code and barcode images unrelated to modem stickers were collected from public sources on the Internet. The criteria were to choose images with different conditions, such as light, distance, and angulation. This step was crucial in order to simulate real-world scenarios, as the input for the QR code and barcode detection model is solely the modem sticker image. The process of annotating and labeling the QR code and barcode images was the same as that of the modem sticker images. The resulting database consisted of 963 images, with 555 QR code objects and 952 barcode objects.

4.1.3. OCR Evaluation

The validation database utilized in this study comprises two components: sticker images and the corresponding character recognition data. The sticker images used in the database have a size of 1920×1080 pixels, providing a standardized resolution for evaluation. The character recognition data include both textual and non-textual information extracted from the stickers. These stickers originate from three different modem models, and as a result, they exhibit variations in the content and layout of the information displayed.

This database serves as the ground truth for evaluating the quality control of the system, as it represents the true and reliable information that we aim to predict. To generate this database, a semi-automatic approach was adopted. Initially, a visual inspection system developed in this study was employed to generate initial answers for each label. Subse-

quently, the generated images were saved, and manual work was carried out to evaluate and correct the data, ensuring its accuracy.

Sticker A includes 43 images with 11 fields, 3 barcodes, and 2 QR codes. Sticker B consists of 40 images with 12 fields, 4 barcodes, and 1 QR code. Sticker C contains 22 images with 10 fields, 3 barcodes, and 1 QR code.

4.2. Sticker, QR Code and Barcode Detection

This section presents a comprehensive description of the process of detecting objects in modem images, specifically the modem label, QR codes, and barcodes. The process comprises several steps, including image input, object detection, and image processing. We illustrated this process in Figure 2.
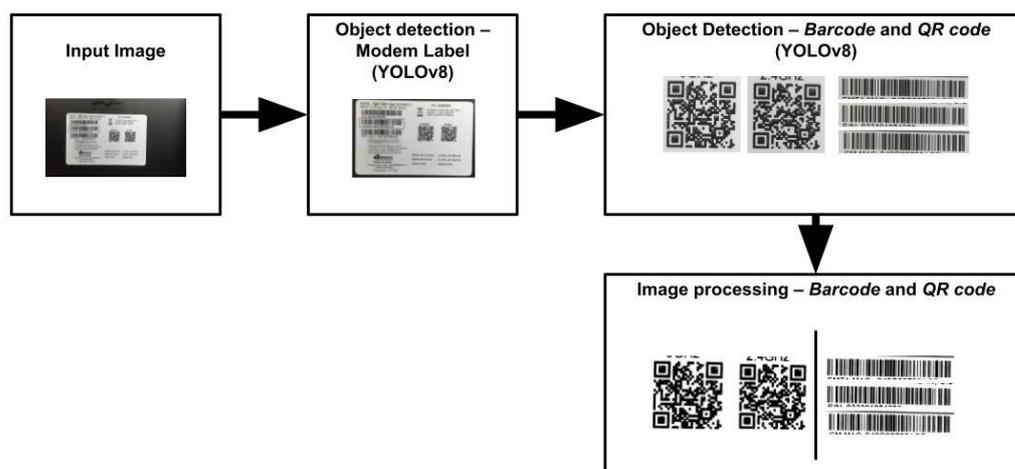
Figure 2. Modem label, QR code, and barcode detection overview.

This process is crucial to ensure the precise identification of both characters present in the image and QR codes and barcodes. It involves segmenting the image into specific areas, identifying characters, and reducing image noise, which significantly improves the accuracy of object detection. The object detection method adopted was YOLOv8. In this step, the input image passes thought the YOLOv8n detection model. This model provides the coordinates of the modem label (bounding boxes). This process is performed to isolate only the interesting portion of the image.

For this step, we applied image processing techniques such as smoothing, binarization, and region of interest using available functions from OpenCV library [37]. This helped improve OCR accuracy. For QR code and barcode object detection, we employed binarization, resizing, and morphological transformations. We used the Zbar library [38] to decode the barcodes and extract the information from the QR codes and the barcodes.

4.3. Sticker Character Recognition

The objective is to accurately identify the characters in the modem images with OCR, enabling further analysis and processing. The OCR module processes the image and identifies characters.

The framework chosen for this step was PP-OCR. The direction classification model was not required in this project since the direction of characters was always correct. The multilingual model of PP-OCR was utilized for text detection and recognition, which takes an image as input and returns a list of detected segments with the recognized text and coordinates inside the image. This approach significantly improved the accuracy of character recognition in images, making it possible to identify characters with precision and efficiency. Figure 3 illustrates this process.
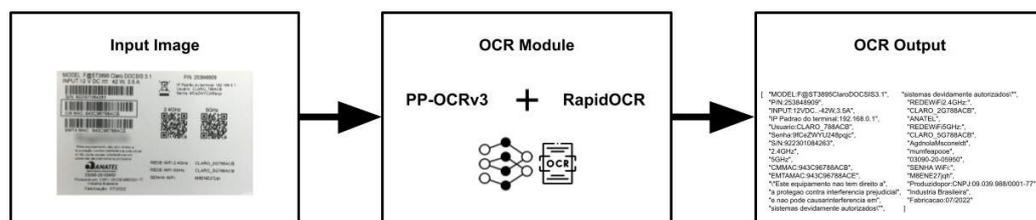
Figure 3. OCR process overview.

We employed PP-OCRv3 [16] in conjunction with the RapidOCR [39] library to optimize the inference time for the PP-OCR.

4.4. Post-OCR

Our post-processing approach aimed to extract information from the OCR output and transform it into a structured and organized format. To achieve this, we adopted a key value format for the fields that needed to be returned, as demonstrated in Table 1. This allowed us to effectively organize the data and facilitate further analysis. Moreover, we also implemented error correction methods to address any inaccuracies that may have arisen during the OCR process.

Table 1. Post-OCR output in key value format for Sticker C subset.

Key	Format	Value
P/N	DDDDDDDD	253951842
IP	DDD$.$DDD$.$D$.$D	192.168.0.1
Usuário	$CLARO_$BBBBBB	CLARO_FDBBD5
Senha	FFFFFFFFFF	ghykmcUG827zxVA
S/N	DDDDDDDDDDDD	929240013008
CM MAC	HHHHHHHHHHHH	B0FC88FDBBD5
EMTA MAC	HHHHHHHHHHHH	B0FC88FDBBD8
REDE Wi-Fi	$CLARO$BBBBBB	CLARO_FDBBD5
SENHA Wi-Fi	FFFFFFFFFFFFFFF	jNyYUUnJtg
MODEL	$F@ST3895 Claro DOCSIS 3.1$	F@ST3896 Claro DOCSIS 3.1

It is crucial to note that the keys employed in this experiment were specifically adapted to correspond to the needs of the company we were serving. However, this approach is inherently generalizable. If the same method were to be adopted in a different context, it would likely necessitate the use of unique keys and values tailored to that specific situation.

The input for the Post-OCR is the result of the previous step, which is a list containing each of the text segments detected and converted into characters as well as the position of the segment inside the image, given by another list containing the coordinates (x, y) of the bounding boxes (BB).

For modem stickers, the fields of interest are already specified by the producer company, and the number of characters of each field is fix. Furthermore, depending on the field, the value only receives specific types of characters, such as numerical, alphanumeric, and hexadecimal. This information was used to model the key value processing step.

To process and retrieve the relevant information from the OCR, a template was implemented in the key value JSON structure. The keys in the template represent the fields in the sticker that should be returned at the end of the visual inspection process, while the values are a set of rules that model the structure of the values of each field. Since different sticker models have different fields, a template is used for each label type. The template format used for the Sticker C pattern is shown in Table 1.

The rules used for the values are based on representations and literals. Representations are alphabet characters indicating the types of characters that can appear in the position of the value. For example, "D" represents decimal numbers, and "H" represents hexadecimal

numbers. The literal texts are indicated by the symbol "$" and signify that all text contained between the literal symbols will be reproduced as it is. The key processing is performed first. This step aims to detect and extract the text segments that contain the keys of the predetermined fields and divide them into key and value. These keys are those specified in the template shown in Table 1 for the Sticker C pattern. For that, three rules were created: the colon rule, the space rule, and the size rule.

The colon rule is applied when an item in the results list contains at least one colon character. The method tries to find a match between the OCR key value and the template value, using the colon character's position as a hint to find the corresponding key. The space rule is applied when the colon rule does not result in a key match. In this case, the method tries to find a key match by splitting the item value into words and testing each word as a possible key. Finally, the size rule is applied when none of the previous rules result in a key match. This rule tries to find a key by comparing the size of the OCR key value with the size of the template key values.

After the key processing step, the value processing is performed. This step aims to fix mistakes that could have happened during the OCR processing. To achieve this, we identified common errors made by the OCR and compiled a list of possible errors based on their representation. Characters that are commonly mistaken for one another, such as "O" and "0", "1", "I", and "/", are corrected by replacing them in the values based on their respective representations. The corrections involve replacing the positions containing literal characters. At the end, a dictionary with textual information is returned. This stage ensures that the values are accurate, and any errors are corrected, resulting in reliable information.

4.5. Evaluation Metrics

In this section, we will cover the metrics that were employed to evaluate the OD results obtained in this study. Additionally, we will introduce the OCR metrics that were utilized to interpret the results.

4.5.1. Object Detection

To evaluate object detection models, the main ways of measuring performance are based on accuracy. The most commonly used metric for evaluating the accuracy of object detection models is mean Average Precision (mAP) [40–42].

A True Positive (TP) is the correct detection of an annotated object (ground truth). A False Positive (FP) is the incorrect detection of a non-existent object or a wrong detection of an existing object. A False Negative (FN) is an undetected object. In addition, a True Negative (TN) is also used for machine learning models. For machine learning models, a TN occurs when the model correctly predicts that an instance does not belong to a certain class. In other words, the model correctly classified a negative sample; that is, a sample that does not belong to the class of interest. As there is an infinite number of bounding boxes that should not be detected within a given image in object detection, a TN is not employed [42].

It is also important to define what is considered a "correct detection" and an "incorrect detection" along with the above concepts. For this, the Intersection over Union (IoU) is used. The IoU is a measure based on the Jaccard similarity, a similarity coefficient for two datasets [43]. For object detection, the IoU measures the overlapping area between the predicted bounding box and the real bounding box or ground truth, divided by the union area between them, given by Equation (1).

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{1}$$

When comparing IoU for a certain threshold t, it is possible to classify detection as correct or incorrect. If IoU > t, the detection is considered correct, and for IoU < t, it is incorrect. In the literature, the most used threshold values t are 0.5 and the mean of the range from 0.5 to 0.95, varying by 0.05.

The evaluation methods used for object detection models are based on precision and recall. Precision measures the model's ability to identify only relevant objects. It is the percentage of correct predictions and is given by Equation (2) [42].

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{All detections}} \tag{2}$$

Recall measures the model's ability to find all relevant cases (all true bounding boxes). It is the percentage of correct predictions among all true objects and is given by Equation (3) [42].

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{All ground truth}} \tag{3}$$

An object detector is considered good when precision and recall values remain high even with an increase in the threshold t, while precision remains high for different recall values. AP is the average precision for each recall level. For each class, precision is calculated at various threshold t levels. AP is calculated as the area under the P x R curve obtained by plotting precision against recall and can be obtained using Equation (4) [42].

$$AP = \frac{1}{11} \sum_{r=i \in 0, 0.1, \ldots, 0.9, 1} \max(P(R_i)) \tag{4}$$

Finally, mAP is the average of AP for all object detector classes, given by Equation (5) [42].

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5}$$

where AP_i is the AP for the i-th class, and N is the total number of classes. This work uses mAP@0.5 as a metric, with mAP with a threshold equal to 0.5, and mAP@0.5:0.95, which is the mean of mAP for the threshold range from 0.5 to 0.95, with a step of 0.05.

4.5.2. OCR

This work will use three metrics to evaluate the performance of the proposed visual inspection. The first is the sticker-wise accuracy rate, and the other two are character error rate (CER) and field error rate (FER).

The sticker-wise accuracy rate is given by the proportion of stickers that returned all their fields without any error, compared to the ground truth, and also decoded all the barcodes correctly.

The error rate (ER) is a metric that quantifies the minimum number of insertions (I), deletions (D), and substitutions (S) of characters, words, or fields, needed to transform the true text into the OCR output, given by Equation (6).

$$ER = \frac{I + S + D}{N} \tag{6}$$

where N is the total number of characters/words in the true text [44].

4.6. Proposed Framework

The proposed process for text label extraction is based on four main steps. The first step is image acquisition, which involves capturing the image of the label using a camera attached to a controlled environment. The second step is detection and image processing, where the label image is extracted from the captured image and processed to reduce noise and correct the image angle. The third step involves information extraction from the label, which is divided into two parts: the image decoding of barcodes and QR codes, and Optical Character Recognition (OCR). Finally, the fourth step involves post-processing of the OCR output. This step involves processing the text information obtained from OCR

in a structured manner that can be easily used as structured data. Figure 4 illustrates an overview of the framework employed in this study.
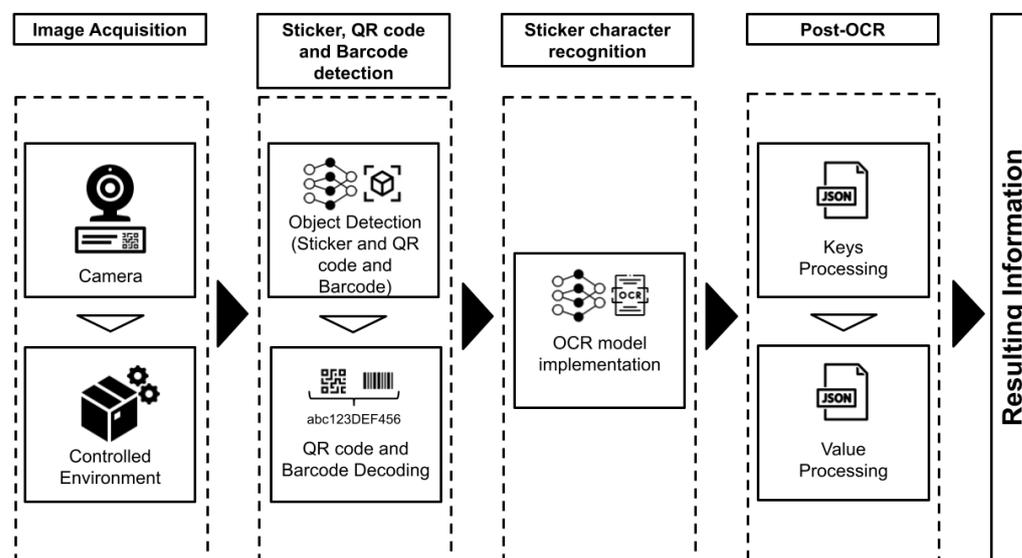

Figure 4. Proposed framework architecture.

5. Results

In this section, we will discuss the outcomes related to sticker detection. Next, we will proceed to exhibit the results obtained from QR code and barcode detection. Finally, we will also present the results of the OCR system integrated with our pipeline.

5.1. Sticker Detection

In the Results section, the label detection model demonstrates outstanding performance, as shown in Table 2. The sticker OD model achieves a remarkable mean Average Precision (mAP) of 0.99 at an IoU (Intersection over Union) threshold of 0.5, highlighting the model's accuracy in detecting stickers. This level of accuracy is crucial, given that the system operates as a cascade, with sticker detection being the first level.

Table 2. Result of the performance evaluation of the label detection model.

Classes	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Modem Sticker	0.972	0.966	0.992	0.930

Despite the apparent simplicity of the task for modern OD systems, the dataset in this study contains various sticker patterns. The primary focus of the research, however, is on modem stickers. The model's performance, as evident in the high precision and recall scores of 0.972 and 0.966, respectively, reflects its ability to accurately identify and distinguish modem stickers from other patterns in the dataset. The mAP score of 0.930 at IoU thresholds ranging from 0.5 to 0.95 further emphasizes the model's robustness in handling different levels of overlap between predicted and ground truth bounding boxes.

5.2. Barcode and QR Code Detection

The results for barcode and QR code detection using the validation dataset are illustrated in Table 3. Accurate and precise detection of barcodes and QR codes on stickers is essential for their subsequent decoding and information extraction.

Table 3. Results for the barcode and QR code detection model.

Classes	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	0.982	0.968	0.984	0.861
QR code	0.987	0.972	0.988	0.855
Barcode	0.977	0.964	0.979	0.868

The detection model demonstrates impressive performance, as evidenced by the high precision and recall scores across both QR codes and barcodes. For QR codes, the model achieves a precision of 0.987 and a recall of 0.972, while for barcodes, the precision and recall scores are 0.977 and 0.964, respectively. The high mean Average Precision (mAP) scores of 0.988 and 0.979 at an IoU threshold of 0.5 for QR codes and barcodes, respectively, indicate the model's strong detection capabilities. This value is most significant than the mAP scores at IoU thresholds ranging from 0.5 to 0.95 (0.855 for QR codes and 0.868 for barcodes).

The overall performance of the model, as shown by the results, ensures the reliable detection of barcodes and QR codes on stickers, facilitating their successful decoding and information extraction.

5.3. Sticker Textual Recognition

The performance of the OCR system was evaluated using three different subsets of sticker patterns, namely, A, B, and C. Table 4 presents the overall performance metrics for each subset, including Character Error Rate (CER), Field Error Rate (FER), and Sticker Accuracy. Additionally, the table also presents a macro of the average results for each metric used.

Table 4. The performance of the OCR system for Subsets A, B, and C is shown. The values in parentheses indicate the standard deviation. The "Number of Images" column represents the total quantity of images used in each subset.

Subset	Number of Images	CER (%)	FER (%)	Sticker Accuracy
A	43	0.21 (0.42)	2.16 (3.87)	0.76
B	40	0.04 (0.16)	0.62 (2.19)	0.92
C	22	0.31 (1.42)	0.34 (1.56)	0.95
All	105	0.17	1.04	0.88

The metrics displayed in the table are interdependent; a misclassified character will affect the CER, which in turn will influence the FER and Sticker Accuracy. It is important to note that a sticker is considered accurate only when there are no errors in any of its fields or characters.

Despite the Sticker Accuracy appearing somewhat unsatisfactory for Subset A, the performance should be evaluated within the context of the stringent requirement for complete correctness in fields and characters.

For Subset A, we have more characters than other sticker patterns (as the field "IP padrão do terminal: 192.168.0.1"), which justifies the performance of 0.76 accuracy. Meanwhile, Subset B demonstrates a marked improvement in Sticker Accuracy, with a value of 0.92. As for Subset C, it achieved the most favorable outcome, reaching a value of 0.95.

Overall, the OCR system's performance varies depending on the sticker subset, with the most accurate results observed in Subset C. The overall accuracy for all sticker patterns was 0.88.

For a more detailed analysis, Tables 5–7 display the CER and FER values for each field of the A, B, and C sticker patterns, respectively.

Table 5. Performance evaluation results of the system for sticker Subset A.

Field	CER (%)	FER (%)
CM MAC	0	0
EMTA MAC	0	0
IP address	0	0
MODEL	0	0
P/N	0	0
NETWORK Wi-Fi 2.4 GHz	0	0
NETWORK Wi-Fi 5 GHz	0	0
S/N	0	0
PASSWORD Wi-Fi	1.19	9.52
Password	1.11	14.29
User	0	0

The system achieved error-free performance for most fields, including CM MAC, EMTA MAC, IP address, MODEL, P/N, NETWORK WiFi 2.4GHz, NETWORK WiFi 5GHz, and S/N. However, it encountered some errors in the PASSWORD WiFi and Password fields, with CER values of 1.19% and 1.11%, respectively, and FER values of 9.52% and 14.29%, respectively.

Overall, these results indicate that the system performed effectively on the A sticker database, with only minor errors. The low CER and FER values demonstrate a high level of accuracy and reliability in recognizing fields within the A sticker database.

Table 6. Result of the performance evaluation of the system for the fields of the B sticker database.

Field	CER (%)	FER (%)
IP	0	0
MAC	0.21	2.50
Model	0	0
PN	0	0
PON/ID	0	0
Password	0	0
S/N	0.33	5.00
SAP	0	0
SSID 2.4 GHZ	0	0
SSID 5 GHZ	0	0
Senha 5 GHZ	0	0
User	0	0

Table 7. Result of the performance evaluation of the system for the fields of the C sticker database.

Field	CER (%)	FER (%)
CM MAC	0	0
EMTA MAC	0	0
IP	0	0
MODEL	0.31	0.34
P/N	0	0
REDE Wi-Fi	0	0
S/N	0	0
SENHA Wi-Fi	0	0
Password	0	0
User	0	0

Most fields in Table 6 exhibit a CER and FER of 0, indicating no recognition errors occurred for those fields. However, the MAC and S/N fields experienced slightly higher CER and FER values, with 0.21% and 0.33% for CER, and 2.50% and 5.00% for FER, respectively.

These results suggest that while the majority of fields were accurately recognized, the system encountered minor difficulties with the MAC and S/N fields in the B sticker database.

In comparison to the previous two tables, Table 7 shows that the system performed well for the fields of the C sticker database, with the majority of fields having CER and FER values of 0. The only field with a slightly higher CER and FER was MODEL, with values of 0.31% and 0.34%, respectively.

These results demonstrate that the system was able to accurately recognize fields across all three sticker databases, with only minor discrepancies in a few fields. This highlights the effectiveness of the proposed system in extracting information from diverse product stickers within an industrial setting.

While the method has demonstrated good performance in Subsets A, B, and C, it has encountered difficulties in certain fields, as evidenced by Tables 5–7 and the high standard deviation relative to the mean values. For the A model, a high error rate was observed in the PASSWORD Wi-Fi and Password fields, which negatively impacted the accuracy per label, despite low average CER and FER values. Upon closer examination of the fields' formats, it was found that they comprised a mix of alphanumeric characters, with alphabetic characters that can be both uppercase and lowercase, without any discernible structural or lexical pattern. The following errors were identified:

(a)   Substitution of "q" by "g";
(b)   Removal of "W" when it appears more than once in a row;
(c)   Substitution of "j" by "i";
(d)   Substitution of "t" by "f";
(e)   Removal of "W" when followed by another "W";
(f)   Substitution of "vv" by "w";
(g)   Substitution of "w" by "W";
(h)   Substitution of "z" by "Z";
(i)   Removal of "v" when followed by another "v".

Upon analyzing the aforementioned errors, it became evident that the primary errors stemmed from confusion between similar alphabetical characters (e.g., "j" and "i"), discrepancies between lowercase and uppercase characters with similar appearances (e.g., "z" and "Z"), and issues with character suppression and substitution when characters such as "w" and "v" appeared consecutively. The errors in the 5657 model, although fewer, were also due to confusion between similar characters.

Figure 5 illustrates these errors, highlighting the impact of factors such as font, character arrangement, and OCR system performance on the final result. For example, consecutive "v" and "w" characters can cause recognition confusion, the physical position of characters and fields within the label can lead to misrecognition when characters are too close or overlapping, and similar characters can create confusion for the OCR model.

Senha: rWWcAVKK972yaut
Ground truth: rWWcAVKK972yaut
Prediction: rWcAVKK972yaut

Senha: rrzZEWUN784pjfy
Ground truth: rrzZEWUN784pjfy
Prediction: rrzZEWUN784pjty

Figure 5. Examples of modem label errors.

6. Conclusions

In this paper, we have presented a groundbreaking framework that amalgamates the capabilities of OCR and OD to automate visual inspection processes within industrial environments. This comprehensive framework adeptly detects and recognizes both textual and non-textual elements on printed stickers used in a company, harnessing cutting-edge AI tools and technologies for sticker information recognition. By transcending traditional Printed Character Recognition (PCR), our system extracts supplementary information, such as barcodes and QR codes, from images.

The experimental results substantiate the effectiveness of our proposed framework over three different sticker patterns, achieving a CER of 0.21 for non-textual recognition on Subset A, 0.04 on Subset B, and 0.31 on Subset C with an overall Sticker Accuracy of 0.88, based on the macro accuracy across all subsets. For OD, sticker recognition attained a mean Average Precision (mAP) of 0.99 at a 0.5 Intersection Over Union (IoU) and 0.98 for barcodes and QR codes using the same metric. Integrating modern OCR with PP-OCRv3 and modern OD with YOLOv8 within our framework enables the seamless recognition and interpretation of diverse information types on printed stickers, fostering more efficient visual inspection processes in industrial applications.

However, our framework exhibits certain limitations. It may encounter difficulties recognizing characters in low-resolution images or when characters are distorted due to suboptimal printing quality. Furthermore, our system may struggle with text featuring unconventional fonts. Certain character sequences also pose challenges, as exemplified in Figure 5. These limitations present opportunities for future research to further refine and enhance the proposed framework.

In future work, we aim to extend the capabilities of our framework to handle different types of labels, including those with unconventional fonts or low-quality printing. Additionally, we will explore the integration of additional technologies, such as natural language processing, to improve the extraction and interpretation of information from complex labels. Furthermore, investigating the scalability of our framework to handle large-scale industrial production processes will be a crucial area of future research.

In conclusion, our trailblazing framework provides a reliable and efficient solution for automating visual inspection processes in the industrial sector by fusing OCR and OD technologies. We have incorporated the latest and most robust techniques within this framework. By accurately detecting and recognizing both textual and non-textual information on printed stickers, our system significantly streamlines industrial workflows and minimizes manual labor demands. The successful implementation of our framework sets the stage for future progress in automating visual inspection processes and underscores the potential of AI-driven solutions in industrial settings.

References

1. Peres, R.S.; Jia, X.; Lee, J.; Sun, K.; Colombo, A.W.; Barata, J. Industrial Artificial Intelligence in Industry 4.0—Systematic Review, Challenges and Outlook. *IEEE Access* **2020**, *8*, 220121–220139. [CrossRef]
2. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
3. Chauhan, R.; Ghanshala, K.K.; Joshi, R. Convolutional Neural Network (CNN) for Image Detection and Recognition. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; pp. 278–282. [CrossRef]
4. Aquino, G.; Costa, M.G.F.; Costa Filho, C.F.F. Explaining One-Dimensional Convolutional Models in Human Activity Recognition and Biometric Identification Tasks. *Sensors* **2022**, *22*, 5644. [CrossRef]
5. Zhang, L.; Zhou, W.; Zhang, X.; Lou, X. An End-to-end Computer Vision System Architecture. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; pp. 2338–2342. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef]
7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Hsu, M.M.; Wu, M.H.; Cheng, Y.C.; Lin, C.Y. An Efficient Industrial Product Serial Number Recognition Framework. In Proceedings of the 2022 IEEE International Conference on Consumer Electronics, Taipei, Taiwan, 6–8 July 2022; pp. 263–264. [CrossRef]
9. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
10. Srivastava, S.; Verma, A.; Sharma, S. Optical Character Recognition Techniques: A Review. In Proceedings of the 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 19–20 February 2022; pp. 1–6. [CrossRef]
11. Jain, M.; Mathew, M.; Jawahar, C. Unconstrained OCR for Urdu using deep CNN-RNN hybrid networks. In Proceedings of the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 26–29 November 2017; pp. 747–752.
12. Plamondon, R.; Srihari, S. Online and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 63–84. [CrossRef]
13. Daraee, F.; Mozaffari, S.; Razavi, S.M. Handwritten keyword spotting using deep neural networks and certainty prediction. *Comput. Electr. Eng.* **2021**, *92*, 107111. [CrossRef]
14. Vinjit, B.M.; Bhojak, M.K.; Kumar, S.; Chalak, G. A Review on Handwritten Character Recognition Methods and Techniques. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 1224–1228. [CrossRef]
15. Du, Y.; Li, C.; Guo, R.; Yin, X.; Liu, W.; Zhou, J.; Bai, Y.; Yu, Z.; Yang, Y.; Dang, Q.; et al. PP-OCR: A Practical Ultra Lightweight OCR System. *arXiv* **2020**, arXiv:2009.09941.
16. Li, C.; Liu, W.; Guo, R.; Yin, X.; Jiang, K.; Du, Y.; Du, Y.; Zhu, L.; Lai, B.; Hu, X.; et al. PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. *arXiv* **2022**, arXiv:2206.03001.
17. Ravi, N.; El-Sharkawy, M. Real-Time Embedded Implementation of Improved Object Detector for Resource-Constrained Devices. *J. Low Power Electron. Appl.* **2022**, *12*, 21. [CrossRef]
18. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef] [PubMed]
19. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
20. Kim, J.H.; Kim, N.; Won, C.S. High-Speed Drone Detection Based On Yolo-V8. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–2. [CrossRef]
21. Li, C.; Liu, W.L.; Guo, R.; Yin, X. *Dive in to OCR*; Baidu, PaddlePaddle, 2022. Available online: https://paddleocr.bj.bcebos.com/ebook/Dive_into_OCR.pdf (accessed on 9 May 2023).
22. Du, Y.; Li, C.; Guo, R.; Cui, C.; Liu, W.; Zhou, J.; Lu, B.; Yang, Y.; Liu, Q.; Hu, X.; et al. PP-OCRv2:Bag of tricks for ultra lightweight OCR system. *arXiv* **2021**, arXiv:2109.03144.
23. Song, S.; Shi, X.; Song, G.; Huq, F.A. Linking digitalization and human capital to shape supply chain integration in omni-channel retailing. *Ind. Manag. Data Syst.* **2021**, *121*, 2298–2317. [CrossRef]
24. Antonio, J.; Putra, A.R.; Abdurrohman, H.; Tsalasa, M.S. A Survey on Scanned Receipts OCR and Information Extraction. February 2022. Available online: https://www.researchgate.net/publication/358405430_A_Survey_on_Scanned_Receipts_OCR_and_Information_Extraction (accessed on 8 May 2023).
25. Subedi, B.; Yunusov, J.; Gaybulayev, A.; Kim, T.H. Development of a low-cost industrial OCR system with an end-to-end deep learning technology. *IEMEK J. Embed. Syst. Appl.* **2020**, *15*, 51–60.
26. Gromova, K.; Elangovan, V. Automatic Extraction of Medication Information from Cylindrically Distorted Pill Bottle Labels. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 852–864. [CrossRef]
27. Gregory, S.; Singh, U.; Gray, J.; Hobbs, J. A computer vision pipeline for automatic large-scale inventory tracking. In Proceedings of the 2021 ACM Southeast Conference, Virtual Event, 15–17 April 2021; pp. 100–107.

28. Santiago Garcia, E. Country-Independent MRTD Layout Extraction and Its Applications. Master's Thesis, University of Twente, Enschede, The Netherlands, 2022.

29. Li, S.; Ma, X.; Pan, S.; Hu, J.; Shi, L.; Wang, Q. VTLayout: Fusion of Visual and Text Features for Document Layout Analysis. In Proceedings of the PRICAI 2021: Trends in Artificial Intelligence: 18th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2021, Hanoi, Vietnam, 8–12 November 2021; Proceedings, Part I 18; Springer: Berlin/Heidelberg, Germany, 2021; pp. 308–322.

30. Dahlberg, E.; Lehtonen, T.; Yllikäinen, M. Tag Recognition from Panoramic Scans of Industrial Facilities. 2022. Available online: https://www.utupub.fi/handle/10024/154486 (accessed on 10 May 2023).

31. Smith, R. An Overview of the Tesseract OCR Engine. In Proceedings of the ICDAR '07: Ninth International Conference on Document Analysis and Recognition, Washington, DC, USA, 23–26 September 2007; pp. 629–633.

32. Nguyen, T.T.H.; Jatowt, A.; Coustaty, M.; Doucet, A. *Survey of Post-OCR Processing Approaches*; Association for Computing Machinery: New York, NY, USA, 2021; Volume 54. [CrossRef]

33. Mei, J.; Islam, A.; Moh'd, A.; Wu, Y.; Milios, E. Statistical learning for OCR error correction. *Inf. Process. Manag.* **2018**, *54*, 874–887. [CrossRef]

34. Khosrobeigi, Z.; Veisi, H.; Ahmadi, H.R.; Shabanian, H. A rule-based post-processing approach to improve Persian OCR performance. *Sci. Iran.* **2020**, *27*, 3019–3033. [CrossRef]

35. Kaló, Á.Z.; Sipos, M.L. Key-Value Pair Searhing System via Tesseract OCR and Post Processing. In Proceedings of the 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia, 21–23 January 2021; pp. 000461–000464.

36. Desai, M.; Mewada, H. A novel approach for yoga pose estimation based on in-depth analysis of human body joint detection accuracy. *PeerJ Comput. Sci.* **2023**, *9*, e1152. [CrossRef]

37. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000** . Available online: https://opencv.org/ (accessed on 9 May 2023).

38. ZBar Development Team. ZBar: Barcode Reader Software. Available online: https://zbar.sourceforge.net/index.html (accessed on 9 May 2023).

39. RapidAI. RapidOCR: Open-Source Optical Character Recognition. Available online: https://github.com/RapidAI/RapidOCR (accessed on 9 May 2023).

40. Zhu, H.; Wei, H.; Li, B.; Yuan, X.; Kehtarnavaz, N. A review of video object detection: Datasets, metrics and methods. *Appl. Sci.* **2020**, *10*, 7834. [CrossRef]

41. Padilla, R.; Passos, W.L.; Dias, T.L.; Netto, S.L.; Da Silva, E.A. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]

42. Padilla, R.; Netto, S.L.; Da Silva, E.A. A survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 1–3 July 2020; pp. 237–242.

43. Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.

44. Nguyen, T.T.H.; Jatowt, A.; Coustaty, M.; Doucet, A. Survey of post-OCR processing approaches. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [CrossRef]