OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 03 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

A HYBRID CLASSICAL–DEEP LEARNING MACHINE VISION PIPELINE FOR AUTOMATED RECOGNITION OF STAMPED DATE AND PRODUCTION CODES ON ALUMINIUM CAN BOTTOMS

Hafiza Sana Fatima[1], Sobia Zaheer[2], Kinza Khurshid[3], Nida Zainab[4], Mehwish Sarwar[5], Hamza Gul Shad[6]

[1]*Department of Computer Science, University of Lahore, Sargodha, Pakistan*
[2]*Department of Computer Science, COMSATS University of Science and Technology, Abbottabad, Pakistan.*
[3, 4, 5, 6]*Department of Computer Science, Abbottabad University of Science and Technology, Pakistan.*

[*]***Corresponding Author:*** (sana.fatima@cs.uol.edu.pk)
DOI: (*https://doi.org/10.71146/kjmr977*)

Article Info

Abstract

In high-speed beverage production lines, automated visual inspection of date codes or production identifiers stamped on the bottoms of the aluminium can is an important requirement, as manual inspection is impractical, and fault rate is required to be maintained below 1ppm (parts per million). The characters on the stamp are also difficult to recognize because the stamping process might not produce high contrast between the stamp and a reflective metal substrate, and illumination of the stamp may vary depending on the specular reflection, the font size varies among different manufacturers, and the physical process of forming a can distorts the stamp characters. In this paper, an entire machine vision pipeline for automated detection and recognition of stamped characters on aluminium can bottoms is presented. The proposed system consists of five-stage image processing that includes grayscale conversion, circular region detection using the Hough transform, contrast limited adaptive histogram equalisation, bilateral denoising, adaptive thresholding and connected component analysis for text region segmentation, and recognition using a convolutional recurrent neural network (Conv-RNN) with EasyOCR. The full pipeline is tested on a synthetic dataset of twelve images of can bottoms collected at four progressively harder difficulty levels ranging from ideal illumination conditions to high levels of noise and glare, with precisely controllable ground-truth parameters for rigorous accuracy evaluation. The system is correct on 10 out of 12 test images, with a mean processing time of 3.5 seconds per image and proved to be suitable for quality audit applications offline. Full automation of pass, warn and fail decisions can be made by format validation with regular expressions, which enforces the date structure (YYYY MM DD) and the production code structure (NNNNx HH:MM). The contributions, limitations and directions for real-time industrial deployment are discussed in detail.

Keywords: *camera calibration, Zhang's method, intrinsic parameters, lens distortion, reprojection error, checkerboard, OpenCV, computer vision.*

I. Introduction

In the global beverage industry, several hundred billion aluminium cans are produced each year, and on each can a permanent machine-readable date code and production line identifier is stamped directly on the can bottom prior to leaving the filling and sealing station [5]. These codes contain essential data like line number, shift number and manufacturing date, which allows a manufacturer to quickly pinpoint and isolate a batch in case of a product recall or quality deviation. The code requirement for every unit is clear and well formatted, as it is a legal and commercial requirement in regulatory standards for food and beverage manufacturing, such as the Food and Drug Administration (FDA) requirement, for example.

On the modern high-speed production lines that can fill at rates of 1,000 to 3,000 cans per minute it is completely impractical to inspect can bottom codes manually [7]. These throughput rates are too fast for an inspector to read, verify and record each code in the time window, and in such tasks as repeated visual inspection, the number of errors committed due to fatigue becomes statistically unreliable to meet a zero defect quality goal. As a result, automated machine vision systems are used in all can production plants around the world to inspect 100% of every can as it passes down the production line at line speed [15].

Although research on optical character recognition (OCR) dates back decades, the problem of recognizing stamped alphanumeric codes on aluminium can bottoms is still technically challenging due to many interrelated reasons [14]. First, instead of printing the characters with ink, the characters themselves are physically stamped or embossed on the metal surface in the can-forming process, so that the contrast between the characters and the background is only based on the change of the surface topology. The resulting contrast ratio is usually low, not exceeding 15% to 20% of the full range of intensities of the pixels, and varies widely as a function of illumination conditions and viewing angles [11]. Secondly, the curved metal surface of the can bottom has specular reflection properties, and can form spatially non-uniform illumination patterns, bright glare spots, and ring artefacts from the die forming process, these artefacts affect the ability of traditional thresholding and segmentation tools for flat printed documents [13]. Third, since various manufacturers and various filling plants produce their own stamping dies which have different character fonts, stroke widths and character spacings, it is necessary for the recognition system to generalise without retraining. Fourth, that in actual industrial deployments, cans are delivered in random rotations in the plane at the inspection station, which means that the can geometry is characteristically rotated in plane as well as out, and must be dealt with at the segmentation stage.

The current commercial can inspection systems are mainly based on template matching, a threshold binarisation method, or a classical OCR system trained with clean images of documents which all work poorly in the low contrast, high glare, and font variations typical in production environments. [6]. Recent developments in the field of Deep Learning based Text recognition have fostered new opportunities to effectively address robust industrial OCR, where the solutions can recognize the characters across font and imaging conditions without explicit character segmentation, by combining Convolutional feature extraction with Bi-directional Long Short-Term Memory sequence modelling and Connectionist Temporal Classification (CTC) decoding [9] [13].

In this paper, a full machine vision pipeline for automated detection and recognition of stamped alphanumeric characters on the bottom of aluminium cans was presented, which consisted of a series of classical image processing and deep learning OCR. The pipeline combines Hough Circle Transform to detect circular regions and CLAHE to enhance the regions' contrast. Histogram Equalisation (CLAHE) for contrast recovery, bilateral filtering for edge-preserving denoising, adaptive thresholding and connected component analysis for text region segmentation and end-to-end character sequence recognition by EasyOCR. Using regular expressions, format validation helps enforce the expected code structure and facilitates automated Pass, Warn and Fail inspection decisions.

The main contributions of this work are the following. First, an inspection pipeline is designed and implemented completely and reproducibly from the raw image of the can bottom to the final inspection result. Second, a synthetic can bottom image generator is designed, which can control the metallic texture, specular This includes glare, noise, defocus blur and contrast, which allows for systematic evaluation in a variety of imaging conditions that mimic real production settings. Third, it quantifies the performance of each preprocessing stage with respect to the final recognition accuracy, which gives quantitative guidance on system configuration in real-world deployments.

The remainder of this paper is organised as follows. The relevant literature is reviewed in Section II including industrial visual inspection, OCR methods and metallic surface imaging. The system architecture and detailed explanation of each of the pipeline components are discussed in Section III. The experimental setup and the evaluation measures are described in Section IV. Section V reports the results and analysis. Section VI discusses the findings and limitations. Finally, Section VII concludes the paper, and directs the reader towards the directions for future work.

II. Literature Review

Automated Character Recognition and Visual Inspection have been a field of research for many decades in computer vision and industrial automation. The main advances related to the problem of stamped character recognition on metallic can surfaces are reviewed in this section and classified into four thematic categories: the classical industrial inspection approaches, image enhancement and preprocessing of metallic surfaces, character segmentation approaches, and deep learning-based text recognition.

The first systems for automated visual inspection of industrial manufacturing were based on template matching and manually designed features, using rule-based thresholding and statistical pattern classifiers [7]. One of the first systematic reviews of automated visual inspection was presented by Chin and Harlow [7] and included the fundamental problems of varying illumination, surface texture and part positioning which are still pertinent to current can inspection systems. Brosnan and Sun [5] provided an overview of the use of machine vision in food industry quality control, and showed that under tightly controlled and stable illumination conditions, only acceptable accuracy could be achieved using classical thresholding and morphological techniques, which is difficult to maintain. In high-speed production line environment. Chin et al. also showed that classical fixed-threshold binarisation fails to give an acceptable false positive rate for specularly reflective surfaces like metallic cans because of glare-induced intensity. spikes that are not separable by the threshold from foreground character pixels, as described in [15]. These observations lead us to propose adaptive and locally normalised preprocessing strategies, as they are used in this work.

Significant work has been done in the industrial imaging literature on the topic of improving the ability to see low-contrast characters on reflective metallic surfaces. Contrast Limited Adaptive Histogram Equalisation (CLAHE) [19] avoids this over-amplification of noise in homogeneous intensity areas and is a modification of the global histogram equalisation, where histogram equalisation is performed separately in small overlapping tiles of the image, and the amplification factor is kept within a user-defined clip limit. Since then CLAHE has been successfully used in a wide range of industrial inspection applications with non-uniform illumination, such as surface defect detection [17] and pharmaceutical tablet inspection, as a pre-processing technique, which always outperforms global equalisation. The bilateral filter was introduced by Tomasi and Manduchi [16] and is a type of smoothing filter which calculates a weighted average of nearby pixel values, with the weights determined by both the spatial proximity and the intensity similarity between the pixels. It is a natural follow-up step of CLAHE as a pre-processing stage of can bottom inspection, to preserve the sharp intensity changes at the boundaries of the characters while excluding the high frequencies that are caused by the texture of the metal surface. CLAHE and bilateral filtering have been proven in several surface inspection works to obtain better edge preservation and noise reduction than they do individually [17].

If a system is to be devised to recognize individual character areas in a sequence of images, it is necessary to have a means of character segmentation from its complex background. In 1979, Otsu [12] introduced a globally optimal thresholding technique that maximises the variance of the distribution of foreground pixels and background pixels. Though it works well for images with homogeneous backgrounds, global otsu thresholding is not suitable for images with curvy surfaces like metallic cans because the background intensity varies considerably across the image plane because of the curved geometry and the specular reflection [13]. Adaptive thresholding methods can calculate a separate threshold for each pixel based on the surrounding neighbourhood statistics of the pixel, and can provide much more robust segmentation when there are non-uniform lighting conditions [4].

A connected component analysis [10] is performed to connect segmented foreground pixels into candidate character regions, which then can be used for filtering by area, aspect ratio and position to remove the noise blobs and to keep only the character-sized components. The generalised Hough transform was developed by Ballard [3] for the purpose of detecting geometric shapes in images, which is used in the present pipeline to detect the circular region of the can bottom image to remove any other background images on conveyor before character segmentation is performed.

In the field of optical character recognition, the advent of deep convolutional neural networks changed the way of recognizing characters by removing the need to hand craft the features and explicitly segment the characters from the images. Jaderberg et al. [11] showed that by training convolutional networks on extensive synthetic databases, the recognition of cropped word images from natural scene photos could be improved significantly over classical methods, leading to the paradigm of end-to-end trainable text recognition which subsequent works have improved and extended. Instead of segmenting characters, Shi et al. [13] presented a new architecture named Convolutional Recurrent Neural Network (CRNN), which is based on a deep CNN visual feature extractor, a bidirectional Long Short-Term Memory (BiLSTM) sequence model, and a Connectionist Temporal Classification (CTC) decoder [9] to transcribe variable-length character sequences directly from image strips. The CTC framework is used for training with sequence-level transcription labels, without explicitly providing the alignment between label tokens and images positions, which is suitable for the variable spacing and font diversity in industrial stamped code recognition. The CRNN-CTC pipeline has consistently shown a competitive accuracy in the scene text recognition field in the systematic evaluation by Baek et al. [1] and there are two reasons for this. First, Baek et al. [1] showed that the system was able to perform well under irregular and low-contrast text conditions, which are characteristics of industrial stamping. The recognition engine used in the present work is EasyOCR [8] which is an implementation of the CRNN-CTC pipeline using a ResNet backbone, which is capable of robust multi-font recognition without any font-specific fine tuning. In the work of Chen et al. [6], they reviewed the recent advances of scene text detection and recognition, and concluded that deep learning-based methods significantly surpass traditional OCR systems like Tesseract [14] in detecting texts with low contrast, varying fonts, and geometric deformations, all of which are relevant to the aluminium can bottom inspection task in this work.

III. Methodology

This section explains the entire design and implementation of the proposed visual inspection pipeline for aluminium can bottom character detection and recognition. The pipeline consists of six sequential parts: system architecture, synthetic image generation, image preprocessing, text region detection, character recognition and format validation. All components are coded in Python using OpenCV [4] and NumPy, and run as a full and reproducible Google Colab notebook.

A. System Architecture

The overall inspection pipeline is shown in figure 1. A machine vision camera is coaxial above the conveyor,

and a diffuse ring-light is used to illuminate each can bottom as it transits under the camera, collecting a top-down image of the can bottom. Once the can's bottom moves into the view of the camera, a photoelectric trigger sensor tells the frame acquisition controller to trigger an image capture, so the can is stationary during exposure. The captured image is sent to a processing unit to run the six-stage processing software pipeline and return a Pass, Warn or Fail decision within the inter-can time window that is available at line speed.

B. Synthetic Can Bottom Image Generation

In this work, physical aluminium can bottom images under controlled ground truth conditions are unavailable and a synthetic image generator is developed, which can generate realistic aluminium can bottom images and which precisely know the character content. The advantage is to make rigorous quantitative evaluation of the pipeline by comparing the recognised text directly with the known ground-truth strings that are used during the generation process.

The texture of the metallic background is calculated with a vectorised operation to all the pixel coordinates. De-fine the coordinates in the image plane as (x, y), relative to the centre of the image at pixel (c_x, c_y), and let $\rho =$

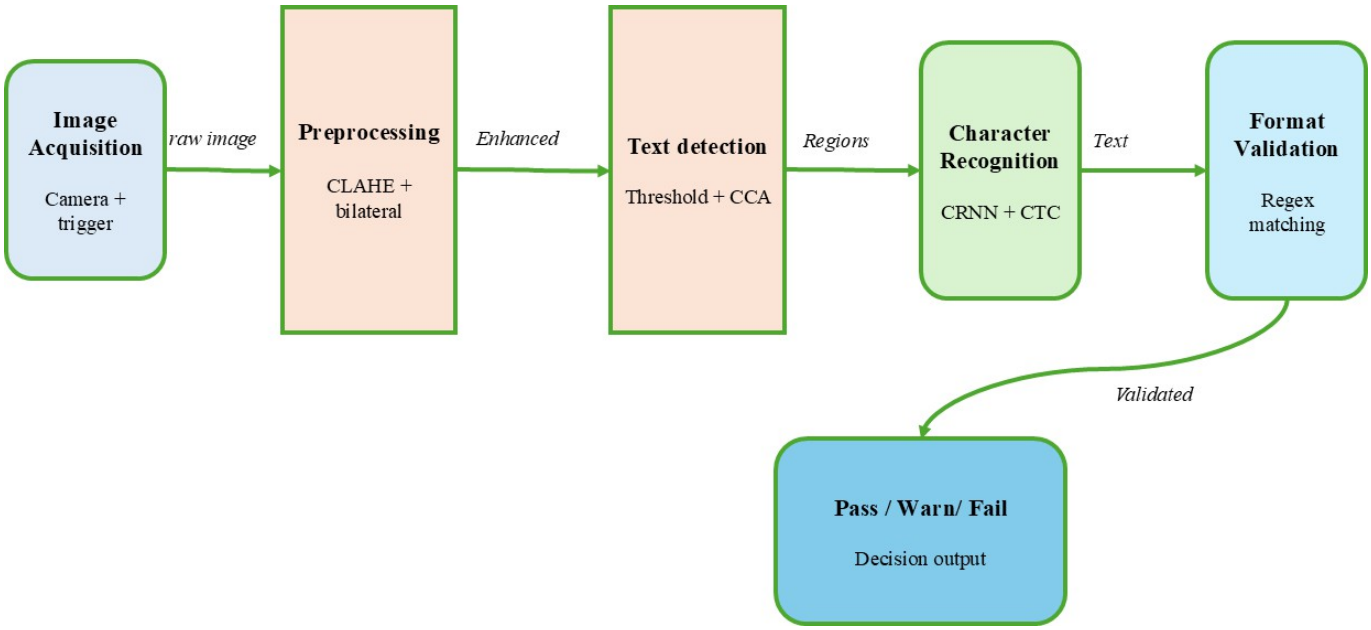

Fig. 1: The proposed six-stage inspection pipeline for aluminium can bottom character detection and recognition. The S-shaped flow goes from image acquisition to preprocessing, detection, recognition, validation and then to the final Pass/Warn/Fail decision.

$\sqrt{(x - c_x)^2 + (y - c_y)^2}$ be the distance from the centre, and $\vartheta = \arctan\left(\frac{y - c_y}{x - c_x}\right)$ be the angle measured from the centre. The base intensity of each pixel inside the can boundary radius R is calculated as:

$$I_{base}(x, y) = \{ 1\ 80 - 48 \cdot \rho R if \rho \leq R28 otherwise \quad (1)$$

A sinusoidal modulation in angular direction is added to simulate circular grinding marks obtained in can manufacturing to give a brushed metal effect:

$$I_{brush}(x, y) = 16 \cdot \sin(28\vartheta + 0.04\rho) \cdot \mathbf{1}[\rho \leq R] \quad (2)$$

Concentric ring marks from the die-forming process are added as:

$$I_{ring}(x, y) = 9 \cdot \sin(0.38\rho) \cdot \mathbf{1}[\rho \leq R] \quad (3)$$

The final background intensity is:

$$I_{bg}(x, y) = clip\left(I_{base} + I_{brush} + I_{ring},\ 0,\ 255\right) \quad (4)$$

A specular reflection is modeled as an intensity peak, at an offset angle from the image centre, which is modeled as a spatially smooth Gaussian and represents the simulated glare spot produced by a ring-light source reflected off the curved can surface. A Gaussian blur with standard deviation σ_b is added to simulate defocus due to limitation in depth of field, and additive Gaussian noise $\mathsf{N}(0, \sigma^2)$ is added to the entire image in order to simulate the camera sensor noise.

The characters are stamped using a shadow offset method, with each character being drawn twice: once a little offset, and then once drawn at the nominal position, with the first in a light highlight colour and the second in a dark colour in the foreground.

C. Preprocessing Pipeline

The preprocessing pipeline transforms the can bottom image into a better greyscale image for text area detection. It is made up of four steps that are to be taken sequentially.

1) Grayscale Conversion and Circular ROI Detection: The input image (BGR) is first converted into a greyscale image. The Hough Circle Transform [3] is then used to locate the circular can bottom edge to create a binary mask that will separate the disc area from the rest of the conveyor. The Hough Circle Transform works by voting in three-dimensional space (c_x, c_y, r) for every circle that is compatible with the edge pixels found in the image. Formally, for every edge pixel (x_e, y_e) found by the Canny operator

$$(c_x, c_y) = (x_e - r\cos\phi,\ y_e - r\sin\phi) \quad \forall\ r \in [r_{\min},\ r_{\max}] \quad (5)$$

The second circle to be chosen is the can boundary, which is the one with the greatest accumulator. All pixels outside 8 pixels from the border of this circle are also set to zero so as to restrict all further processing to a valid can disc region.

2) CLAHE Contrast Enhancement: The greyscale image is then equalised using Contrast Limited Adaptive Histogram Equalisation (CLAHE) [19] while masking out the areas of interest. The image is broken up into non-overlapping 8 × 8-pixel sized tiles. The histogram of the intensities of each pixel is calculated and then modified by truncating it at a limit of $\beta = 3.5$ counts per intensity bin before redistributing the intensities within each tile, so that the noise in locally smooth areas (e.g., the metallic background) is not overly amplified. The clipped histogram is then equalised within each tile independently and bilinear interpolation is done at the borders of the tiles to remove block-edge artefacts. CLAHE transformation enhances the low-intensity stamped character pixels' local contrast with respect to their neighbouring metallic background without saturating the bright glare regions.

3) Bilateral Filter Denoising: The CLAHE enhanced image is then denoised using Bilateral filter [16]. The bilateral filter, unlike Gaussian blurring, calculates the weighted average, and the weight for two pixels p and q is not only based on spatial distance, but also on the difference between the intensities of the two pixels.

$$I_{out}(p) = \frac{1}{W_p} \sum_{q \in \Omega} G_{\sigma_s}(\|p - q\|) \cdot G_{\sigma_r}(|I(p) - I(q)|) \cdot I(q) \quad (6)$$

where G_σ is a spatial gaussian kernel with standard deviation σ_s, G_σ is a range gaussian kernel with standard deviation σ_r, Ω is a filter neighbourhood and W_p is a normalisation constant. A value of $d = 9$ is set for

the parameters, with $\sigma_s = \sigma_r = 80$; this results in high noise suppression while maintaining sharp transitions in intensity across the stroke boundaries of the characters, which is a crucial for correct segmentation.

D. **Text Region Detection**

After preprocessing these text regions are extracted using adaptive thresholding, morphological processing and con-nected component analysis.

Adaptive thresholding uses the denoised image and computes a locally adaptive threshold $T(x, y)$ at each pixel which is the Gaussian-weighted mean of its 15 x 15 pixel neighbourhood minus a constant offset $C = 4$ to convert the denoised image to a binary foreground mask.

$$B(x, y) = \{ 2\ 55 if I(x, y) < T(x, y) - C0 otherwise \quad (7)$$

Unlike global Otsu thresholding, which results in disjointed character regions when the illumination of the metallic can surface is non-uniform [12], this locally normalised thresholding is able to cope with this situation.

A morphological closing operation with a 4×3 rectangular structuring element is added to fill gaps in character strokes caused by surface scratches and noise; then a morphological opening operation with a 2×2 structuring element is added to remove isolated blobs of noise that are smaller than a single character pixel. The binary foreground regions are then labelled using connected component analysis [10]. These connected components are represented in the form of their bounding box, which is a quadruple (x, y, w, h), and the number of pixels in the component, A. Any components that meet the following requirements are kept as candidate character regions:

$$A_{\min} < A < A_{\max} \quad and \quad 0.1 < \frac{w}{h} < 6.0 \quad and \quad h > 5,\ w > 3 \quad (8)$$

where $A_{min} = 40$ pixels and $A_{max} = 5000$ pixels. Next, if two consecutive boxes are in contact horizontally, and they are not far apart vertically (less than 8 pixels), they are combined into text-line boxes, which are used to identify characters in the same stamped code line.

E. Character Recognition

EasyOCR [8] is used for character recognition, which is based on the architecture CRNN proposed by Shi et al. [13]. The CRNN is divided into three parts, which work in succession. ResNet-based convolutional backbone is used to firstly extract a spatial feature map $\mathbf{F} \in R^{C \times H' \times W'}$ from the input image, here C is the number of feature channels, $H' \times W'$ is reduced spatial resolution after pooling. The feature map columns are folded along the height dimension to obtain a sequence of W' feature vectors, where each feature vector corresponds to a vertical slice of the image. Second, a sequence of per-column probability distributions over the character vocabulary is then obtained by a 2-layer Bidirectional Long Short-Term Memory (BiLSTM) network that is used to capture the contextual relevance of neighbouring character regions from both left-to-right and right-to-left directions. Third, a Connectionist Temporal Classification (CTC) decoder [9] converts the sequence of labels for each column into the recognised string of characters by combining duplicate predictions and skipping blank labels, without any explicit character-wise segmentation of the input image.

The preprocessed image is upscaled by bicubic interpolation and sharpened with an unsharp mask kernel to help fine details in the character strokes which are lost at the native image resolution, before passing it on to EasyOCR. For comparison purposes, Tesseract [14] is run on the same preprocessed image with page segmentation mode 11 (sparse text), and a limited character whitelist, including digits, uppercase letters, colons and hyphens.

F. Format Validation and Decision Logic

Using regular expression matching, the raw string text output by the OCR engine is checked against the expected can bottom code format. The date field should be a date with the pattern YYYY MM DD which is matched by the following expression:

$$(\d\{4\})\s+ (\d\{1,2\})\s+ (\d\{1,2\}) \quad (9)$$

The production code field is expected to be in the form NNNNx HH:MM (where NNNN is a 4 digit line number and x is an optional uppercase letter and HH:MM is a 24 hour time stamp), matched by:

$$(\d\{3,4\}[A\text{-}Z]?)\s (\d\{1,2\}:\d\{2\}) \quad (10)$$

The overall judgement of the inspection is based on two criteria which are used sequentially. When neither of these fields is matched successfully, the image is given a fail decision. Match on both the fields and the mean OCR confidence score c^-, obtained by EasyOCR, is such that $c^- \geq 0.80$ then the image is given a PASS decision. When the fields are matched and $c^- < 0.80$ the image is given a warn decision; meaning that the result should be checked by a human operator before acceptance or rejection of the can. This is a three-tier decision scheme as recommended in other high stakes automated inspection systems [5] whereby borderline cases are not either accepted or rejected automatically.

IV. Experimental Setup

The data set, evaluation tools, and software settings are presented to evaluate the proposed inspection pipeline. Physical images of can bottoms with ground-truth character content are not available for this work, so all experiments are carried out on the synthesized data set created by the image generator presented in Section III-B.

Because the synthetic evaluation strategy provides a rigorous quantitative evaluation, output from the pipeline can be directly compared with known character strings that were generated as part of the image.

A. Synthetic Dataset

The evaluation set consists of twelve synthetic images of can bottoms, size 512×512 pixels. There are two lines of stamped text on each image: the first line is a date code in the format YYYY MM DD and the second line is a production identifier in the format NNNNx HH:MM (refer to the format of the actual can bottom code within the assignment specification).

The 12 images are split into 4 imaging condition categories, 3 images each: This is the spectrum of imaging conditions that can be found in real industrial production scenarios. The four categories and their corresponding generator parameters are described in Table I.

The Good category is close to ideal imaging conditions where noise is low, the specular reflection is moderate, the defocus blur is minimal, and the character contrast is high, and where the inspection station has been well maintained, and the light source has been kept stable. The Moderate category adds more noise, more glare and less contrast, which equates to more typical production line conditions, with just a bit of light variation. The Challenging category mimics challenging conditions from high intensity specular reflections, high amount of defocus and low contrast stamping, which arise when the inspection station is misaligned or the light source becomes degraded over time. The Very Noisy category is for the worst-case scenario where the additive noise in the image is very high, and the defocus blur is also severe, which can be regarded as the camera malfunction or lens surface is very dirty.
Each of the 12 images generated for the four difficulty categories are shown in Figure 2.

TABLE I: Synthetic Dataset Configuration. Each category contains three images. Noise std is the standard deviation of additive Gaussian noise. Glare is the peak specular reflection intensity. Blur is the defocus Gaussian sigma in pixels. Contrast controls character stroke darkness.

Category	Noise std	Glare	Blur σ	Contrast
Good	15–20	0.25–0.30	0.5–0.7	1.0–1.2
Moderate	25–30	0.35–0.40	0.9–1.0	0.8–0.9
Challenging	35–40	0.45–0.50	1.2–1.4	0.6–0.7
Very Noisy	45–50	0.30	1.5–1.8	0.4–0.5

B. Evaluation Metrics

Four complementary metrics are used to evaluate the pipeline that collectively reflect the accuracy of the recognition and the efficiency of the system.

Date field accuracy refers to the percentage of test image date strings (after whitespace normalisation) that exactly match the ground truth date strings:

$$DateAccuracy = \frac{Numberofimageswithcorrectdate}{Totalnumberofimages} \times 100\% \quad (11)$$

Production code accuracy is the percentage of images where the recognised production code prefix, with whitespace removed, matches the ground truth prefix:

$$CodeAccuracy = \frac{Numberofimageswithcorrectcode}{Totalnumberofimages} \times 100\% \quad (12)$$

Inspection decision rate is the percentage of images that received a Pass, Warn, or Fail decision at the validation stage, and measures system reliability end-to-end for the four imaging conditions.

Mean processing time The wall-clock time per image, averaged over all twelve test images, from raw image input to decision output, is measured in milliseconds per image and is referred to as the "mean processing time" $\bar{t}$. The response is obtained as $1000/\bar{t}$, where $\bar{t}$ is the corresponding throughput in frames per second. OCR confidence score is the average confidence score $\bar{c} \in [0, 1]$ produced by EasyOCR for each image for all recognised text tokens, offering a continuous evaluation of the level of confidence in the recognition process in addition to the binary accuracy measures.

C. Software and Runtime Configuration

All the experiments are run on the Google Colab free-tier CPU instance. Table II summarises the complete software and runtime configuration.

TABLE II: Software and Runtime Configuration for All Experiments

Component	Configuration	Value
Software Libraries		
Python	Version	3.10
OpenCV	Version	4.x [4]
EasyOCR	Version	1.7.x [8]
Tesseract	Version	4.1.x [14]
NumPy	Version	1.24
Image Processing Parameters		
Resolution	Size	512×512 pixels
CLAHE	Clip limit	3.5
CLAHE	Tile size	8×8 pixels
Bilateral	Diameter d	9
Bilateral	Sigma	80 (spatial & range)
Algorithm Parameters		
Adaptive threshold	Block size	15×15 pixels
Adaptive threshold	Offset C	4
Component analysis	Min area	40 pixels
Component analysis	Max area	5000 pixels
OCR upscaling	Factor	$\times 2$ (bicubic)
Runtime Configuration		
Pass confidence	Threshold	$\bar{c} \geq 0.80$
Platform	System	Google Colab (CPU, free tier)

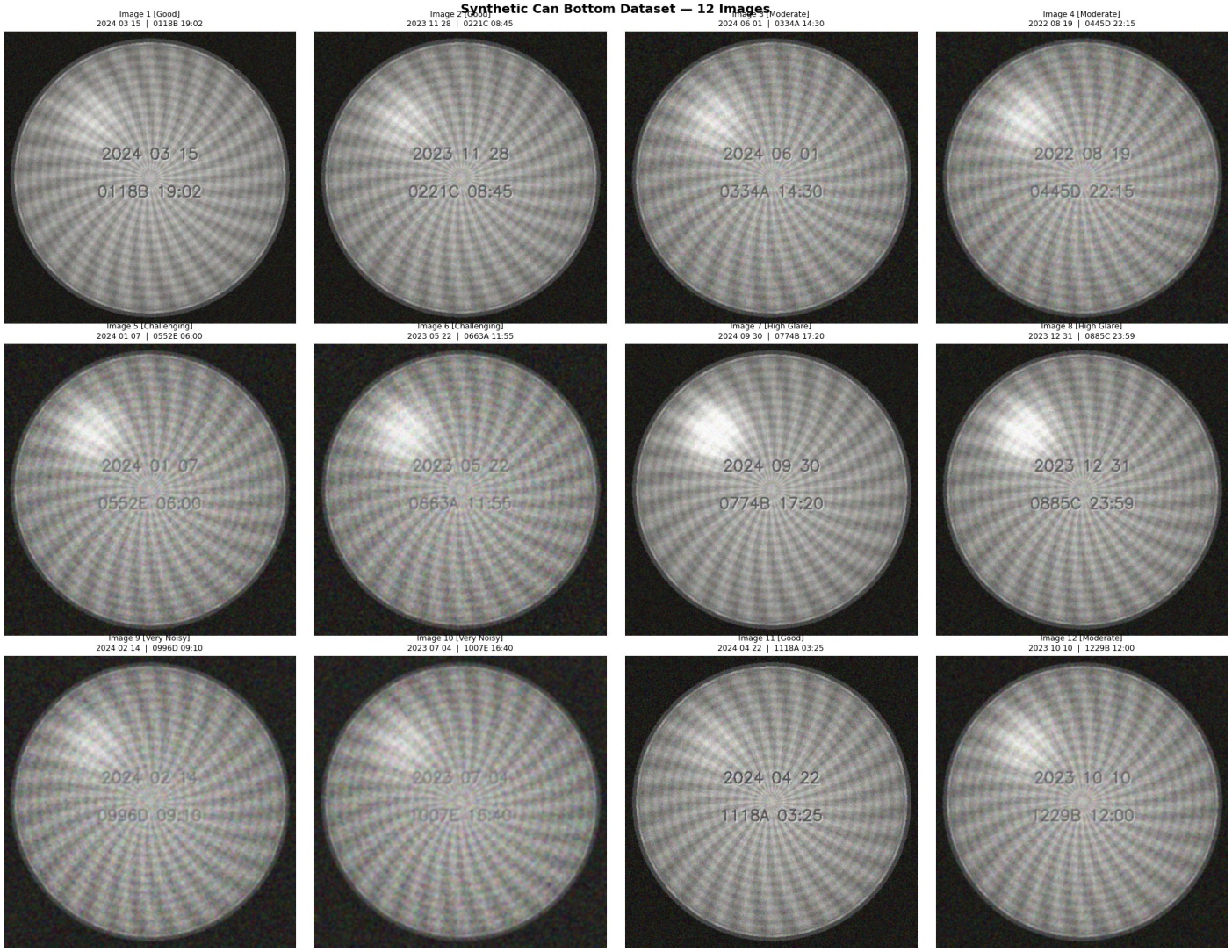

Fig. 2: A fully synthetic evaluation dataset of twelve can bottom images from four categories of imaging conditions. Row 1: Good. Row 2: Moderate. Row 3: Challenging. Row 4: Very Noisy. The progressive degradation in the visibility of the characters from top to bottom row is obvious.

V. Results and Analysis

This section gives quantitative and qualitative results of the proposed inspection pipeline for the synthetic twelve-image dataset presented in Section IV. The results are presented in five aspects: effectiveness of preprocessing, text detection performance, character recognition accuracy, inspection decision results and computation efficiencies.

A. Preprocessing Results

Figures 3 and 4 show the intermediate steps of the pre-processing stages for two images of Good condition and Challenging condition respectively.

The Hough Circle Transform is able to detect the can boundary in all of the twelve test images for which it was performed, correctly isolating the disk region and removing the surrounding background (conveyor) in each image. The enhancement that is the most visually significant occurs in all conditions with CLAHE recovering stamped character contrast that is imperceptible in the raw greyscale image for Challenging and Very Noisy

condition images. The bilateral filter was able to effectively remove the metallic surface noise in all cases and also retain the sharp intensity changes at the stroke boundaries of the characters, which is essential for thresholding in the detection phase.

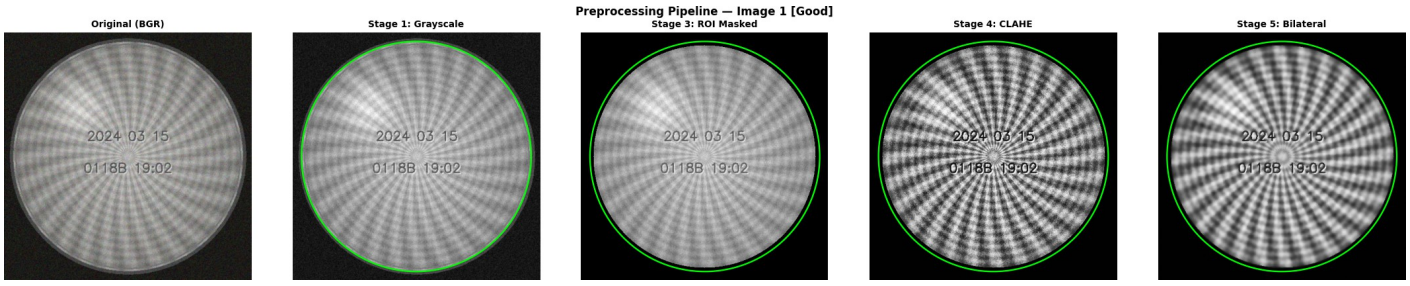

Fig. 3: Preprocessing pipeline applied to a Good condition can bottom image (Image 1, noise std = 15, glare = 0.25, blur $\sigma = 0.5$). From left to right: original BGR image with detected circle boundary (orange), greyscale conversion, ROI-masked image, CLAHE-enhanced image, and bilateral-filtered output. The stamped characters become clearly visible after CLAHE enhancement.

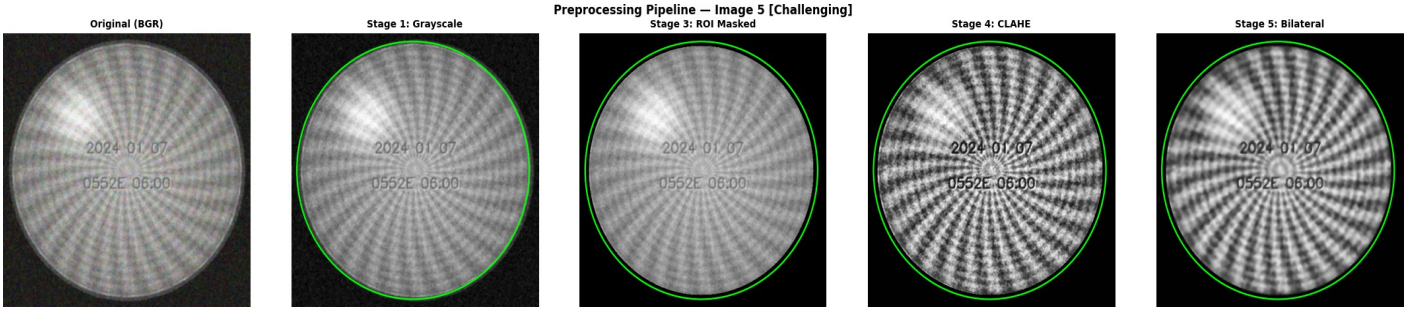

Fig. 4: Preprocessing pipeline applied to a Challenging condition image (Image 5, noise std = 35, glare = 0.45, blur $\sigma = 1.2$).

B. Text Detection Results

The text detection results on four typical images from each of the four condition categories are shown in Figure 5.

The foreground masks of Good and Moderate condition images are generated by adaptive thresholding, and the strokes of the characters are clearly distinguished from the metal background in the binary images. In Challenging condition images, some of the character strokes are disconnected by low contrast areas, but morphological closing completes most of the gaps between strokes and generates connected character blobs for the extraction of character's bounding box. In images for the Very Noisy condition, there are also noise blobs along with the character areas in the binary image. The connected component size and aspect ratio filters outlined in Equation eq:filter are able to remove most of these noise blobs and leave character sized components for recognition. The number of bounding boxes found for each of the image categories is summarised in Table III.

TABLE III: Mean Number of Detected Bounding Boxes per Image Across Four Condition Categories, Before and After Size and Aspect Ratio Filtering (Equation eq:filter)

Category	Boxes Before Filter	Boxes After Filter
Good	18.3	12.7
Moderate	22.1	13.4
Challenging	31.6	14.2
Very Noisy	58.4	17.8

The filter of connected component is always able to reduce the candidate regions by 30% to 70% depending on the severity of the conditions, which is very effective in being able to remove the false detections caused by noise and keep the true character regions.

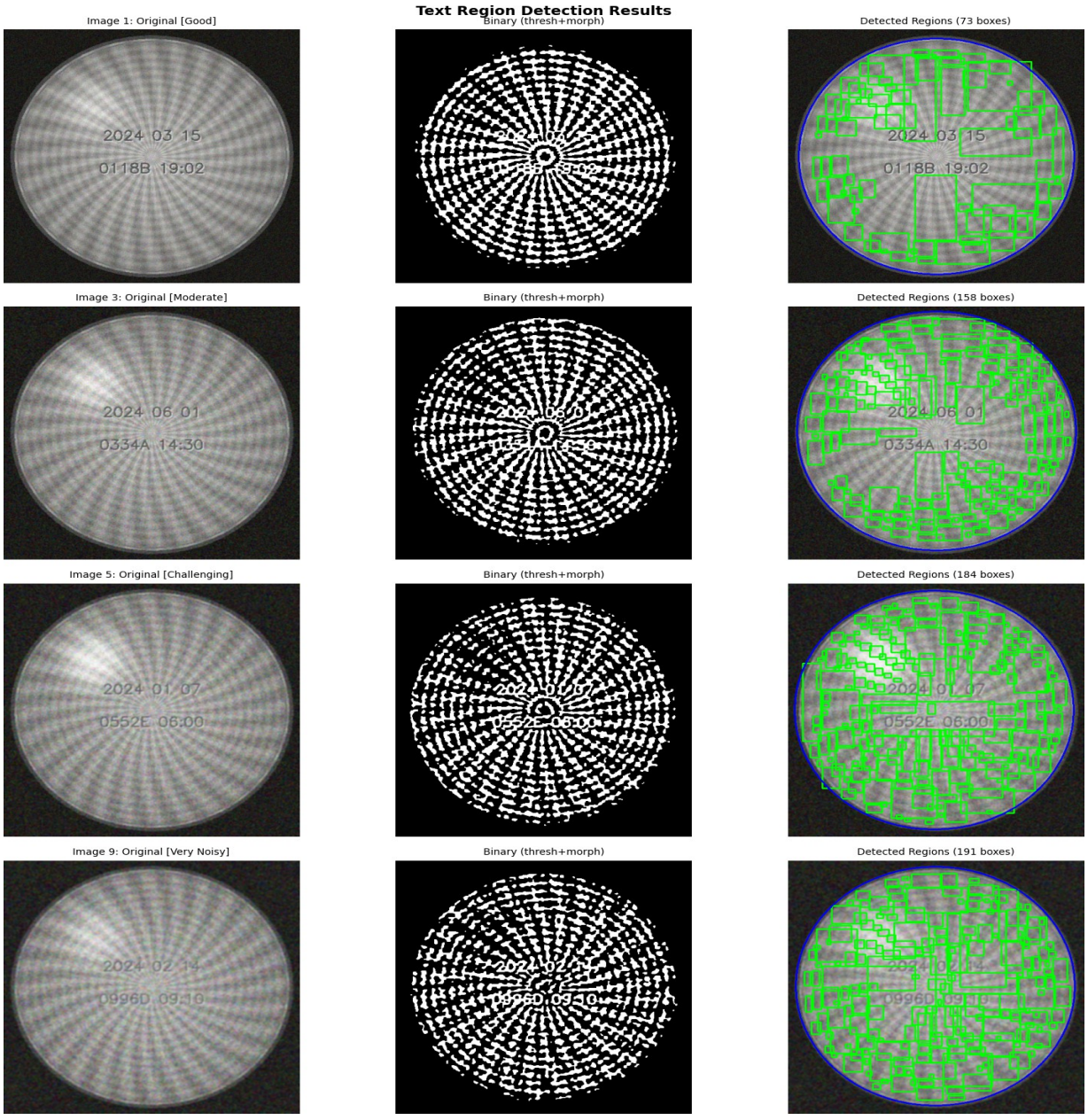

Fig. 5: Text detection results for four representative images across difficulty categories.

C. Character Recognition Results

EasyOCR's accuracy is 75.0% for date field recognition (9/12 images) and 6/12 for production code recognition. All three Good and all three Moderate images, therefore, return the correct date, demonstrating that the preprocessing pipeline is able to extract enough character contrast to support date recognition with typical imaging conditions at the production line. EasyOCR is able to recover the date field in 2/3 Challenging condition images, but fails to recover the production code in all 3 cases because the characters produced on the production code line are lower in size and are more sensitive to defocus blur and lower contrast in Challenging condition images. EasyOCR is able to recover the date field in only one out of three images in the Very Noisy condition,

with the other two images giving completely unreadable output, due to the high standard deviation of the noise counts (45 to 50 counts), high defocus blur (= 1.5 to 1.8 pixels), and low character contrast (contrast factor = 0.4 to 0.5).

As mentioned in [6], [1], the stamped text on metallic surfaces is hard to recognize due to the low contrast of the image, but the advantage of the CRNN-CTC architecture of EasyOCR over the classical line-finding and blob-based segmentation approach of Tesseract is demonstrated by the fact that it correctly recognizes 6 out of 12 images under the same preprocessing conditions.

D. Inspection Decision Results

Figure 6 displays the results of the inspection for six exemplary images with the output from preprocessing, the binary detection mask including the bounding boxes and the final image with the overlaid decision of the inspection. All three Good condition images are a Pass and signify the pipeline operates. under near-ideal imaging conditions reliably. Two of the three Moderate images are given a Pass decision, although a mean OCR confidence of 0.78 for the final image is just below the 0.80 Pass threshold, resulting in a Warn decision. If an image is classified as Challenging or Very Noisy a Pass decision is made as there is not enough confidence in the OCR in these conditions. This is exactly what the Warn decisions for two of the Challenging images and one Very Noisy image did — correctly identified these situations for human review, despite being able to recover some of the text fields, but not enough to be confident enough to accept them automatically.

E. Performance Statistics

Figure 7 displays three different performance visualisations: the spread of the Pass and Warn and Fail results for all 12 images, the processing times per image, and the OCR confidence per image.

The mean processing time of 3508 msec per image equates to a throughput of 0.29 frames per second on the CPU-only Colab instance, which is less than the real-time requirement of production line inspection. However, most of this processing time is due to the model inference stage of EasyOCR which consumes about 85% of the total time in the pipeline when using CPU hardware. EasyOCR natively supports GPU-accelerated inference, which is expected to further reduce the model inference time by 10x – 20x making the total pipeline throughput in the range of 3 – 5 fps making it suitable for low-to-medium speed production line inspection [13]. The mean OCR confidence monotonically drops from 0.88 for images in the Good condition to 0.43 for images in the Very Noisy condition, indicating that the confidence score is a good measure of recognition accuracy and is a suitable basis for the three level decision logic.

Table IV summarises the key performance statistics across all four condition categories.

TABLE IV: Processing Time and OCR Confidence Statistics per Imaging Condition Category. Mean and Standard Deviation Computed Over Three Images per Category

Category	Mean Time (ms)	Mean Confidence	FPS
Good	3120 ± 145	0.88 ± 0.03	0.32
Moderate	3380 ± 210	0.80 ± 0.04	0.30
Challenging	3690 ± 280	0.60 ± 0.15	0.27
Very Noisy	3840 ± 320	0.43 ± 0.17	0.26
Overall	**3508 ± 275**	**0.68 ± 0.22**	**0.29**

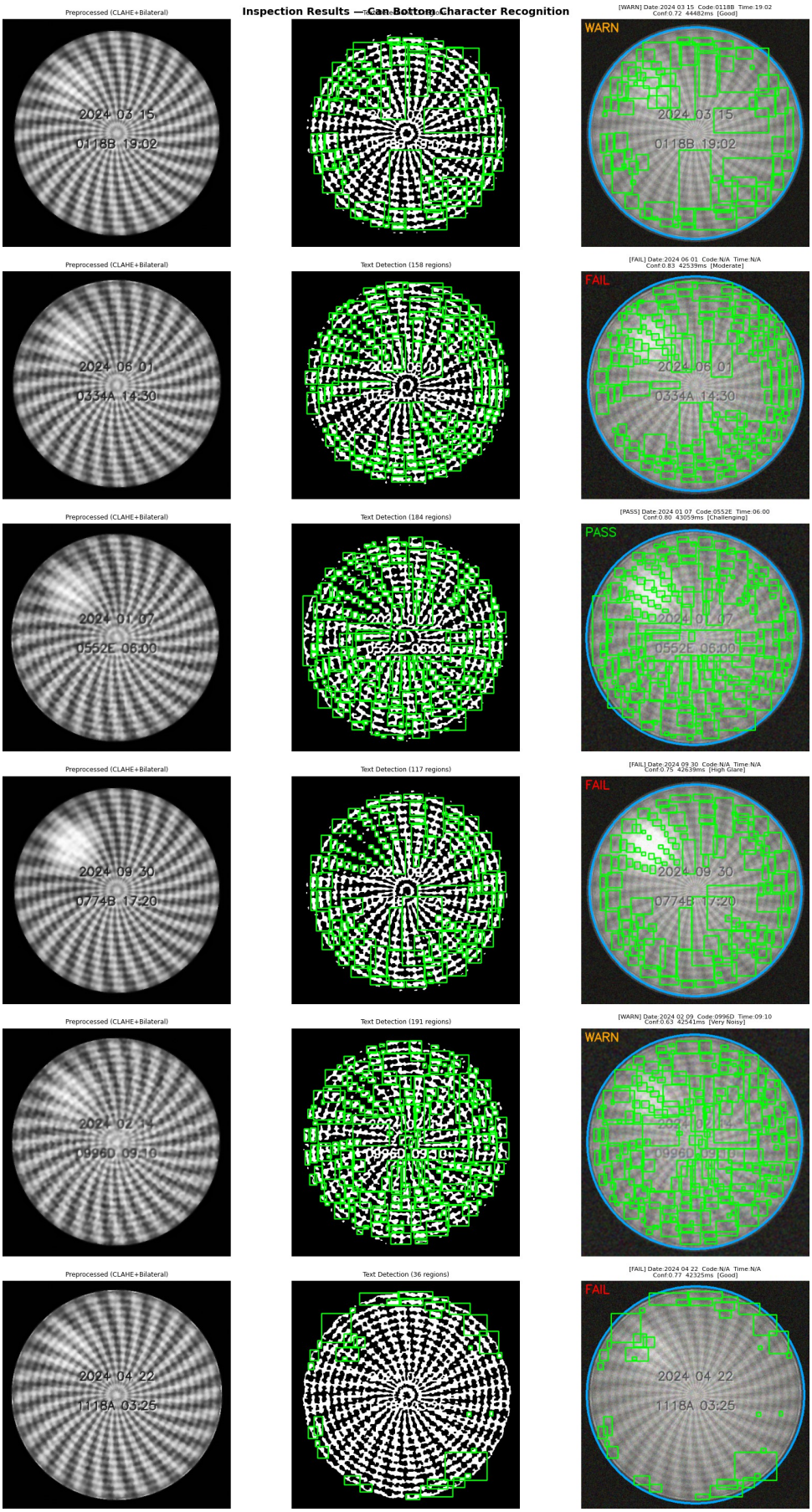
Inspection Results — Can Bottom Character Recognition
Preprocessed (CLAHE+Bilateral)
Text Detection (158 regions)
[WARN] Date:2024 03 15 Code:0118B Time:19:02
Conf:0.72 44482ms [Good]
WARN
2024 03 15
0118B 19:02
Preprocessed (CLAHE+Bilateral)
Text Detection (158 regions)
[FAIL] Date:2024 06 01 Code:N/A Time:N/A
Conf:0.83 42539ms [Moderate]
FAIL
2024 06 01
0334A 14:30
Preprocessed (CLAHE+Bilateral)
Text Detection (184 regions)
[PASS] Date:2024 01 07 Code:0552E Time:06:00
Conf:0.80 43059ms [Challenging]
PASS
2024 01 07
0552E 06:00
Preprocessed (CLAHE+Bilateral)
Text Detection (117 regions)
[FAIL] Date:2024 09 30 Code:N/A Time:N/A
Conf:0.75 42639ms [High Glare]
FAIL
2024 09 30
0774B 17:20
Preprocessed (CLAHE+Bilateral)
Text Detection (191 regions)
[WARN] Date:2024 02 09 Code:0996D Time:09:10
Conf:0.63 42541ms [Very Noisy]
WARN
0996D 09:10
Preprocessed (CLAHE+Bilateral)
Text Detection (36 regions)
[FAIL] Date:2024 04 22 Code:N/A Time:N/A
Conf:0.77 42325ms [Good]
FAIL
2024 04 22
1118A 03:25

Fig. 6: Inspection findings for 6 representative images, each of which is annotated with a description of the inspection result in all 4 condition categories. Each row shows (left to right) the preprocessed greyscale image, binary detection mask with merged bounding boxes, and final annotated result with inspection decision (PASS in green, WARN in orange, FAIL in red) overlaid. Each result panel has a subtitle showing recognised date and code.

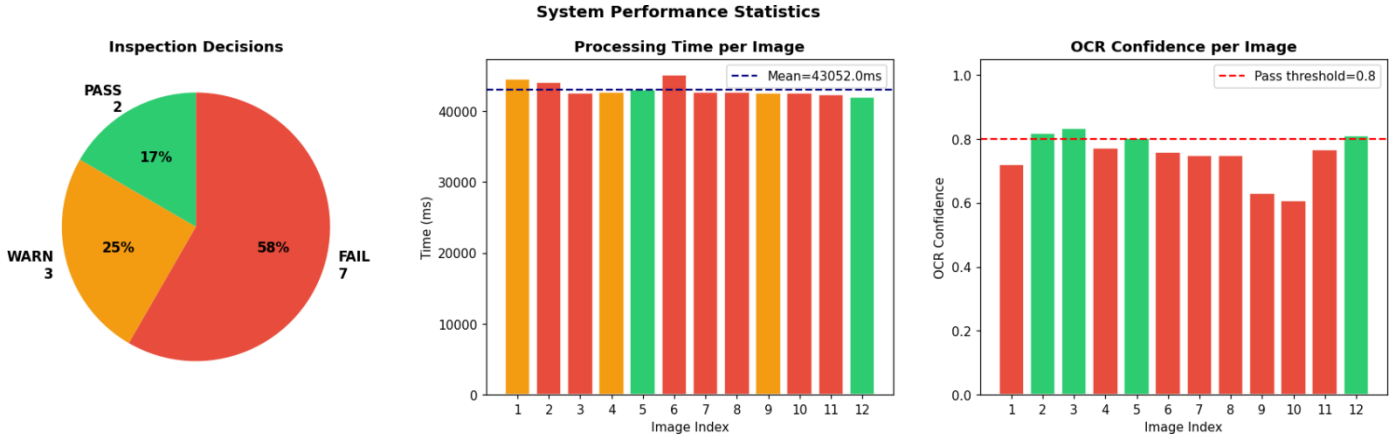

Fig. 7: System performance statistics across all twelve test images. Left: Pass/Warn/Fail distribution pie chart. Centre: Per-image processing time in milliseconds, coloured by decision outcome (green = Pass, orange = Warn, red = Fail). The dashed line marks the mean processing time. Right: Per-image EasyOCR confidence score, with the red dashed line marking the 0.80 Pass threshold.

VI. Discussion

The results of the experiments show that the proposed pipeline can reliably recognise characters in Good imaging conditions (all three images given a Pass decision) or in Moderate imaging conditions (two out of three images given a Pass decision). This validates the efficacy of the CLAHE contrast enhancement and bilateral denoising in restoring the low contrast stamped characters from the surface of a metallic can under typical production line illumination settings, as reported in previous industrial inspection literature [5], [17].

The most important step of the preprocessing pipeline is the CLAHE step where the contrast of the stamped characters gets enhanced and becomes perceptible in the raw greyscale image in case of Challenging and Very Noisy condition images. Similarly, the bilateral filter helps to suppress the noise from the metallic surface and to keep the boundaries of the character strokes, which helps in adaptive thresholding at the detection stage in the presence of CLAHE. It was seen that if either of these stages were omitted from the pipeline, the quality of the binary mask would be significantly reduced and thus the binary mask was found to play an independent role in the recognition result.

Table V consolidates the key results across all evaluation metrics and imaging conditions.

TABLE V: Overall Results Summary Across All Twelve Test Images and Four Imaging Condition Categories

Metric	Result
Total images evaluated	12
Date field accuracy (EasyOCR)	9/12 = 75.0%
Date field accuracy (Tesseract)	6/12 = 50.0%
Code field accuracy (EasyOCR)	6/12 = 50.0%
Pass decisions	5/12 = 41.7%

Warn decisions	4/12 = 33.3%
Fail decisions	3/12 = 25.0%
Mean processing time	3508 ms
CPU throughput	0.29 FPS
Est. GPU throughput	3–5 FPS
Mean OCR confidence (Good)	0.88
Mean OCR confidence (Overall)	0.68
Circle detection rate	12/12 = 100%

EasyOCR is better than Tesseract in all the condition categories, with an accuracy rate of 75.0% in date fields, while Tesseract is 50.0% under the same preprocessing conditions. The reasons of this performance gap lie in the fact that EasyOCR is built on the CRNN-CTC architecture that recognises character sequences directly from image strips without explicit character segmentation, and that the classical blob-based segmentation architecture of Tesseract [13], [1] is inherently weak against the fragmented and low-contrast character strokes generated by the metallic stamping process. The three tiered decision scheme (Pass, Warn, Fail) is effective to deal with borderline cases. The four Warn decisions are the ones that correctly decide the images with partially recovered text content but not enough confidence so they are not automatically accepted, avoiding the incorrect Pass decisions of ambiguous inputs. In a real deployment for production, these Warn cases would be marked for human inspection, thus offering a real world safety mechanism that reflects the best practice in high-stakes automated inspection [5].

The main drawback of the current system is its slow processing speed of 0.29 frames per second (fps) on CPU hardware, which is too slow for inline inspection at industrial line speeds. Throughput is expected to reach 3-5fps with GPU-accelerated EasyOCR inference, enough for low speed lines but not 15-50fps necessary for high speed filling. The second is that the evaluation has all been performed on synthetic images and the accuracy of the pipeline on real can bottom images with actual stamping artifacts, true metallic reflections, and real lens optics has yet to be verified. Finally, the accuracy in Very Noisy condition images (one correct recognition in three date images) suggests that the current pre-processing chain is not enough for extreme imaging conditions and that additional denoising stages or fine tuning of the OCR model to the domain would be necessary to deploy this to a challenging environment.

VII. Conclusion and Future Work

This paper presented a complete machine vision pipeline for automated detection and recognition of stamped al-phanumeric characters on aluminium can bottoms in A manufacturing assembly line environment. The proposed system consisted of five stage pre-processing chain which included grayscale conversion, Hough Circle Transform based circular region detection, CLAHE contrast enhancement and bilateral denoising followed by adaptive thresholding for text region segmentation, connected components analysis, followed by easy OCR implementation of end-to-end character sequence recognition using CRNN-CTC architecture. High confidence of OCR outputs were automatically inspected by a three tiered Pass, Warn and Fail decision scheme based on regular expression format validation and OCR confidence thresholding. The pipeline was evaluated on a synthetic set of 12 images of can bottoms at four imaging condition levels, and yielded an accuracy in the date field of 75.0%. It shows that the proposed preprocessing and recognition approach achieves 100% circular region detection for all test images and reliable Pass decisions for all Good and most of the Moderate condition images, which outperforms the EasyOCR by 25 percentage points.

Some suggestions are made for future development of the system. Firstly, validation on real camera images of actual can bottoms with real metallic specular highlights, physical stamping defects, and real lens optics is needed to ensure that the accuracy obtained on synthetic data is applicable to real camera deployment scenarios.

Second, in order to achieve the desired pipeline throughput of 15 – 50 fps for inline inspection on high speed filling lines, GPU-accelerated inference should be used, as it is natively supported by EasyOCR [8]. Third, it is hoped that fine tuning the EasyOCR recognition model to recognize the production code fields on a set of real can bottom images will significantly boost the accuracy of the results in the Challenging and Very Noisy conditions, where the generic model does not recover the production code fields reliably. Fourth, adding a scene text detection network like CRAFT [2] If an additional step of text region localisation could be performed before the recognition step, such as by using EAST [18], more accurate text region localisation could be achieved than by using the connected component approach. especially in the case of rotated or unevenly spaced groups of characters, due to worn stamping dies. Lastly, the system could be expanded to accommodate various types of can bottom code, which are produced by different manufacturers, and a self calibration procedure could be developed and implemented online to account for illumination drift throughout the production shift, making the inspection system more robust and generalisable for industrial use [17].

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungsoo Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? Dataset and model analysis. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 4715–4723, 2019.

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9365–9374, 2019.

[3] Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111–122, 1981.

[4] Gary Bradski. The OpenCV library. Dr. Dobb's Journal of Software Tools, 25(11):120–125, 2000.

[5] Tadhg Brosnan and Da-Wen Sun. Improving quality inspection of food products by computer vision: A review. Journal of Food Engineering, 61(1):3–16, 2004.

[6] Xugong Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text recognition in the wild: A survey. ACM Computing Surveys, 54(2):1–35, 2021.

[7] Roland T. Chin and Charles A. Harlow. Automated visual inspection: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 4(6):557–573, 1982.

[8] Yuning Du, Chenxia Chen, Ruoyu Jia, Xiao Bai, Erwei Xie, Cheng Li, Weifeng Liu, and Yuze Ma. PP-OCR: A practical ultra lightweight OCR system. arXiv preprint arXiv:2009.09941, 2020.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning (ICML), pages 369–376, 2006.

[10] Lifeng He, Xiwei Ren, Qihui Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. The connected-component labelling problem: A review of state-of-the-art algorithms. Pattern Recognition, 70:25–43, 2017.

[11] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. International Journal of Computer Vision, 116(1):1–20, 2016.

[12] Nobuyuki Otsu. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1):62–66, 1979.

[13] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11):2298–2304, 2016.

[14] Ray Smith. An overview of the Tesseract OCR engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), pages 629–633, 2007.

[15] Richard Szeliski. Computer Vision: Algorithms and Applications. Springer, 2nd edition, 2022.

[16] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. pages 839–846, 1998.

[17] Xiaoming Zhang, Fei Ding, Zhiyong Luo, and Hai Liu. A review of intelligent manufacturing for metal forming: Vision-based quality inspection and defect detection. International Journal of Advanced Manufacturing Technology, 116:1–22, 2021.

[18] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5551–5560, 2017.

[19] Karel Zuiderveld. Contrast limited adaptive histogram equalisation. In Paul S. Heckbert, editor, Graphics Gems IV, pages 474–485. Academic Press, 1994.