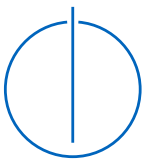
SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

AI-Based Turbocharger Identification for Automated Model Recognition and Repair Support

Ivana Peneva



SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY

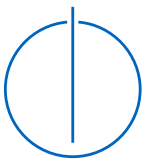
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

AI-Based Turbocharger Identification for Automated Model Recognition and Repair Support

KI-basierte Turbolader-Identifikation zur automatisierten Modellerkennung und Reparaturunterstützung

Author:	Ivana Peneva
Supervisor:	Prof. Alois C. Knoll
Advisors:	Prof. André Borrmann & Panagiotis Petropoulakis & Yonghan Kim
Submission Date:	May 5, 2026



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, May 5, 2026 Ivana Peneva

Acknowledgments

First and foremost, I would like to thank my thesis advisor, Panagiotis Petropoulakis, for trusting me with the opportunity to pursue a topic I selected myself. I offer my heartfelt gratitude for his constant motivation, invaluable guidance, his important insights, and his unwavering commitment to nurturing this study. I would also like to extend my appreciation to Yonghan Kim. His problem-solving skills, out-of-the-box thinking, and invaluable support were an exceptional help to me. Working alongside both of them taught me critical thinking and showed me how to effectively approach and dismantle complex problems.

I extend my sincere thanks to Prof. Alois C. Knoll and Prof. André Borrmann for their support and for providing the academic framework that made this thesis possible. I am incredibly grateful to Unitech Turbo Ltd. for giving me the foundational idea that sparked this master's thesis. I would like to show my gratitude to my best friend who has been physically and mentally by my side throughout my whole master's studies. Lastly, I want to express my profound gratitude to my parents for their unwavering presence in my life, providing me with invaluable encouragement and opportunities from the very beginning, extending far beyond just this thesis but throughout my entire academic journey. Their constant support, even from a distance, has been a true blessing, and I will forever be thankful to them. This accomplishment would not be possible without all of the people involved. Thank you!

Abstract

In Maintenance, Repair, and Overhaul (MRO) environments, manual logging of industrial part numbers is time-consuming and highly prone to human error. Reliable asset tracking is needed, but its automation is severely hindered by nameplate degradation, including oil, rust, and glare, alongside perspective distortions and the widespread use of dot-peen engraving, which fundamentally breaks the continuous-stroke assumptions of traditional Optical Character Recognition (OCR) systems. To address this issue, this thesis develops and systematically evaluates an end-to-end computer vision pipeline designed to extract Garrett turbocharger part numbers from heavily degraded nameplates. The workflow integrates spatial preprocessing with advanced visual decoding and a deterministic fuzzy-matching database engine. The findings demonstrate that background removal, perspective rectification, and orientation correction are essential, increasing overall system accuracy by over 34 percentage points while simultaneously reducing token consumption. Furthermore, generative Multimodal Large Language Models (MLLMs) vastly outperformed OCR engines, with the open-weight Qwen2-VL architecture achieving a peak accuracy of 76.03% when deployed with an unconstrained transcription strategy. Utilizing constrained prompts to directly extract the part number severely degraded the reasoning capabilities. Through rigorous stress testing, the pipeline's operational limits were quantified. While the pipeline is resilient to global speckle noise, its performance reaches a functional ceiling when deep physical scratches that obscure more than one of the digits of the part number are present. Finally, an execution time analysis revealed a per-image processing time of approximately 55 seconds on an NVIDIA L4 Tensor Core GPU (24 GB VRAM), identifying spatial preprocessing as the primary computational bottleneck. Having a digitized, verified part number enables automated model recognition, allowing technicians to seamlessly cross-reference the specific replacement components and maintenance protocols required for repair.

Contents

1 Introduction

As global manufacturing transitions toward Industry 4.0 and the circular economy, accurately tracking, maintaining, and recycling physical assets has become a critical operational challenge [1, 2, 3]. The ability to precisely and reliably identify a part and its specifications is critical in sectors like the automotive industry, aerospace and defense, oil and gas, heavy machinery, metalworking, and logistics asset tracking. The recognition of such parts has become a severe operational bottleneck. Technicians still manually read part numbers of physical components, such as nameplates, from a picture and write them into a digital database. Manual logging in industrial settings is very tedious and a time-consuming affair, making this process heavily unreliable and highly prone to human error [4].

Extracting this unique part number is the critical first step in the industrial digitization pipeline. Once the part number is reliably acquired, it acts as the primary key for automated model recognition. This directly enables automated repair support, as technicians can instantly use the extracted part number to query inventory databases, identify the exact model, and determine the specific replacement parts and maintenance kits required for the repair [5].

Recent advances in computer vision and artificial intelligence have opened new possibilities for automating this process. While traditional OCR systems have been effectively developed for the extraction of standard, printed characters, deploying these systems in industrial environments presents a unique set of challenges [6, 7]. In those environments, harsh conditions such as extreme physical wear are the norm. The nameplates that include the identification number are also frequently exposed to chemicals, such as lubricating oils, degreasers, strong acids and bases, and they often have to remain readable after painting and galvanizing. Furthermore, for tracking high-value assets, security is important. That is why many industries utilize DPM techniques such as dot-peen engraving. This process uses hemispherical indentations to engrave the metal surface, creating characters out of a series of indented, disconnected dots [8, 9].

While dot-peen marking provides physical durability and security, it creates a formidable challenge for automated data extraction. Furthermore, in real-world environments, those nameplates suffer degradation, such as oil, dirt, and scratches, which make some digits from the part number unreadable. The pictures taken of these

nameplates also introduce complex perspective distortion, multi-axis skew, and glare from the reflective metallic surfaces.

This research evaluates the necessity of spatial preprocessing, investigates whether traditional OCR or modern MLLMs yield higher accuracy, gives insights about the token consumption, prompting strategies, and execution latency, and measures system robustness against specific, escalating levels of synthetic industrial noise. To achieve this, the study develops an end-to-end automated pipeline for part number recognition, utilizing a dataset of turbocharger nameplates as the primary case study. By separating the workflow into distinct, measurable stages — image preprocessing, part number extraction, and database-powered verification — the pipeline allows each component to be tested and improved independently. The evaluation shows where failures originate and how they propagate through the system, providing a clear evidence base for practical deployment decisions.

Additionally, the text of this thesis was reviewed using language-assistance tools such as ChatGPT and Gemini to improve grammar, punctuation, and overall readability. These tools were used solely for language editing purposes. The ideas, methodology, experiments, and substantive content of this thesis were conceived and written by the author.

1.1 Problem Statement

As established, extracting critical identification data, like for example part numbers, from images of industrial nameplates in real-world MRO environments presents a severe computer vision challenge. Years of physical wear result in severe surface degradation, leaving nameplates dirty, oily, rusty, and with deep scratches. Furthermore, inconsistent environmental lighting interacts with reflective metallic surfaces to create extreme glare in pictures, frequently washing out low-contrast engravings. The photographic capture process itself inherently alters the image geometry, resulting in images that suffer from multi-axis skew and perspective distortions. The frequent use of dot-peen typography, where characters are formed by multiple disconnected dots rather than continuous strokes, fundamentally breaks the assumptions of standard text recognition algorithms.

These conditions collectively challenge traditional OCR systems, which typically rely on assumptions of clean, high-contrast, and continuously rendered text. While recent advances in MLLMs suggest improved robustness to noisy visual inputs and dot-peen text through enhanced visual reasoning capabilities, a critical technology gap remains. It remains unclear whether MLLMs can reliably replace traditional OCR systems in this domain. Furthermore, the extent to which explicit spatial preprocessing improves

extraction accuracy and how different prompting strategies affect both accuracy and computational efficiency has yet to be fully quantified.

Additionally, existing literature lacks systematic stress testing to quantify exactly how and when advanced pipelines fail under ascending, controlled levels of specific visual degradations, such as surface occlusions and visual noise, and how this noise can be computationally removed to increase the accuracy.

Therefore, the primary research problem addressed in this thesis is the lack of a systematically evaluated, computationally efficient pipeline for reliably extracting verified part numbers from degraded industrial nameplates.

1.2 Research Objectives and Research Questions

To address the problem statement, this study aims to design, optimize, and stress test a part number recognizing pipeline. Consequently, this research is guided by the following primary Research Questions (RQs):

- **RQ1:** How do spatial preprocessing (specifically background elimination, geometric rectification, and orientation correction) and the choice of extraction strategy (Constrained Structural Parsing versus Unconstrained Raw Transcription) impact the text extraction accuracy and token usage on distorted dot-peen nameplates?
- **RQ2:** To what extent do State-of-the-Art (SOTA) MLLMs with and without constrained prompting outperform traditional deep-learning OCR engines in terms of accuracy, and how do these architectures compare regarding token consumption?
- **RQ3:** To what extent do specific typologies and escalating intensities of industrial noise, namely isolated surface occlusion (scratches) and global visual noise (speckles), degrade the performance of an optimized extraction pipeline, and how can the impact of this noise be computationally mitigated?

1.3 Expected Contributions

Answering the research questions above requires more than a working pipeline. It requires an evaluation design that can isolate the contribution of each stage and test extraction strategies under identical conditions with the same datasets. This thesis makes the following contributions toward these requirements:

1. **Preprocessing and Prompting Strategies:** By utilizing Segment Anything Model 2 (SAM 2)-based background removal and geometric rectification and orientation correction, this research isolates and measures how spatial preprocessing influences downstream extraction. Furthermore, it defines how the choice of prompting strategy (Constrained Structural Parsing versus Unconstrained Raw Transcription) impacts both the pipeline's accuracy and the token consumption.

2. **MLLMs vs OCR:** This research provides a quantified evaluation of SOTA MLLMs (GPT-4o [10], Gemini 2.5 Flash [11], InternVL3 [12], and Qwen2-VL [13]) with and without constrained prompting against traditional deep-learning OCR frameworks (PaddleOCR [14], Tesseract [15], EasyOCR [16]). It establishes a clear performance benchmark for extracting dot-peen characters without semantic linguistic context and quantifies the computational latency of the pipeline.

3. **Stress Testing:** By systematically stress-testing the optimized extraction pipeline against a controlled, six-tier dataset of escalating synthetic noise, this research quantifies at what levels of occlusion the developed pipeline fails and what methods could help limit the influence of the noise.

4. **Datasets and Databases:** This thesis introduces a proprietary dataset of real-world images of turbocharger nameplates, alongside a multi-tier synthetically degraded dataset generated from a targeted subset of 40 images. Furthermore, it develops a fully automated, modular pipeline that bridges raw optical extraction with deterministic data verification, utilizing a custom visually-weighted fuzzy matching algorithm to ground predictions in a master part number reference database.

To ensure the reproducibility of the results and facilitate further research, the complete implementation of the AI-based turbocharger identification system is made publicly available on GitHub [17] and on LRZ Sync and Share [18].

1.4 Scope and Assumptions

To ensure a rigorous, focused, and measurable evaluation, the boundaries of this research are defined as follows.

1. **Hardware and Image Capture Environment:** This research strictly assumes the use of decentralized, handheld smartphone images of MRO environments. All the original images in the dataset are taken in a real-world setting at an active turbocharger repair workshop. Some images are created by adding artificial

noise (like dirt, oil, scratches) to the nameplates. Developing any hardware-based solutions is explicitly outside the scope of this work.

2. **Primary Real-World Image Dataset Size:** The primary evaluation is bounded to a curated dataset of 127 real-world images. While this size is sufficient to isolate failure mechanisms and benchmark model architectures, broad statistical generalization across all possible industrial conditions remains a subject for future investigation.

3. **Target Identification and Database Limit:** The pipeline extracts only part numbers from Garrett turbocharger nameplates. Expanding the architecture to recognize other manufacturers (e.g., BorgWarner, Holset) or extracting secondary identification strings (such as Turbo Models or OEM numbers) is reserved for future work.

4. **Master Part Number Reference Database Integrity:** The downstream fuzzy-matching pipeline relies on a proprietary database of 7,871 historical turbocharger part numbers. An assumption of this research is that this database is exhaustively complete and accurate. Handling out-of-database part numbers or correcting part numbers within the database itself falls entirely outside this scope.

5. **Ground Truth Annotation Accuracy:** The manual annotations used to establish the primary ground truth dataset, including spatial bounding boxes and part numbers, are assumed to be 100% accurate. Identifying or correcting annotation errors falls outside the scope of this study.

6. **Degradation Typologies:** The synthetic stress testing is limited to six escalating tiers of physical occlusion (scratches) and impulse noise (speckles/dirt). While extreme specular glare is a known challenge, synthesizing accurate 3D light reflection on 2D images is highly complex. Therefore, the isolated impact of specular glare was not systematically simulated and falls outside the scope of this master's thesis.

7. **Architectural and Model Selection:** The evaluation of text extraction is bound to three traditional deep-learning OCR frameworks (Tesseract, EasyOCR, PaddleOCR) and four MLLMs (GPT-4o, Gemini 2.5 Flash, InternVL3, Qwen2-VL). Furthermore, the MLLMs are evaluated strictly on their extraction capabilities using two defined prompting paradigms (Constrained Structural Parsing and Unconstrained Raw Transcription). Domain-specific fine-tuning of the open-weight architectures is outside the scope of this evaluation.

8. **Computational Efficiency and Edge Deployment:** While evaluating token consumption and measuring the baseline per-image inference latency on a standard GPU are within the scope of this thesis, optimizing the end-to-end multi-stage pipeline (including SAM 2 and PaddleOCR) for low-latency, real-time edge deployment is outside the immediate scope of this thesis.

2 Theoretical Background

To establish the automated extraction pipeline developed in this thesis, this chapter introduces the fundamental concepts, methods, OCR engines, and MLLMs utilized in this research. It details the underlying mechanics of all of the steps of the pipeline, providing the necessary theoretical framework to understand their subsequent application and evaluation.

2.1 Optical Character Recognition

Optical Character Recognition (OCR) is a specialized branch of pattern recognition dedicated to identifying and classifying optically processed characters. In this research, we aim to answer the question of whether SOTA MLLMs (e.g., Gemini 2.5 Flash, GPT-4o, InternVL3, Qwen2-VL) outperform traditional OCR engines (e.g., PaddleOCR, EasyOCR, Tesseract) in accurately extracting part numbers from industrial images. After concluding this evaluation, the goal is to choose the best text extraction engine for the described domain. That is why we compare multiple OCR engines with each other in Experiment 3 in Section 5.3.

An OCR system consists of a multi-stage pipeline [19, 20]:

1. An image of an analog document is captured.
2. Text-containing regions of the document are located and distinguished from non-text elements, such as graphics. During segmentation, the text regions are further isolated into individual characters or words.
3. The characters are preprocessed to eliminate noise like small breaks or thin line widths and variations of symbol size and rotation, which could degrade recognition rates.
4. Features of the characters are extracted through template matching, which directly compares the raster image matrix to stored prototypes, or feature-based techniques, which extract significant measurements.
5. During classification, the extracted features are analyzed to identify the character and assign it to the correct character class. This is frequently achieved using

methods like minimum distance classifiers, statistical classifiers, and Neural Networks (NNs).

6. In the final stage, individual recognized symbols are grouped together based on their spatial location to reconstruct words and numbers.

Tesseract

Tesseract is an open-source OCR engine. The engine features built-in support for Unicode and has the ability to recognize more than 100 languages. Tesseract uses a specialized, multi-stage heuristic and machine-learning pipeline [21, 15]:

1. **Preprocessing and Binarization:** Images undergo preprocessing such as grayscale conversion and morphological operations like dilation and erosion to unify text objects. For binarization, For binarization, Tesseract utilizes a global thresholding technique based on Otsu's method. Otsu's method searches for the optimal threshold t that maximizes the inter-class variance $\sigma_b^2(t)$ between foreground and background pixels:

$$\sigma_b^2(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2, \tag{2.1}$$

   where ω_0 and ω_1 represent the probabilities of the two classes, and μ_0 and μ_1 represent their respective means.

2. **Page Layout Analysis:** The engine performs connected component analysis to group candidate tab-stops into lines and column partitions, separating text regions from non-text elements.

3. **Feature Extraction:** Instead of relying on raw pixel matrices, Tesseract extracts features based on the polygonal approximation of a character's outline. This creates a 4-dimensional feature vector consisting of the x-position, y-position, direction, and length. Dot-peen characters do not consist of continuous strokes but rather of multiple disconnected hemispherical indentations. That means that those characters lack a polygonal outline. This architectural reliance on continuous edges explains why Tesseract fundamentally struggles with DPM typography.

4. **Two-Pass Classification:** Tesseract utilizes a unique two-pass recognition system. In the first pass, words are recognized using a static classifier. During this stage, the classifier calculates the weighted distance d_f of each extracted feature from its nearest stored prototype using the equation:

$$d_f = d^2 + w\theta^2, \tag{2.2}$$

where d is the Euclidean distance of the feature coordinates from the prototype line. θ is the angular difference from the prototype. This distance is converted into feature evidence E_f using a decay constant k:

$$E_f = \frac{1}{1 + kd_f^2}. \tag{2.3}$$

The features and prototypes are then normalized to compute a final distance metric d_{final} used by the K-Nearest Neighbors (KNN) algorithm for character classification:

$$d_{final} = 1 - \frac{\sum_f E_f + \sum_p E_p}{N_f + \sum_p L_p}. \tag{2.4}$$

Words recognized with satisfactory confidence are passed to an adaptive classifier to serve as training data. A second pass is then run over the page, allowing this newly trained, font-sensitive adaptive classifier to re-evaluate words that were poorly recognized initially.

EasyOCR

EasyOCR is a modern, lightweight, and open-source Python OCR package built on the PyTorch framework [22]. While Tesseract relies heavily on geometric heuristics and two-pass classification, EasyOCR uses an end-to-end deep learning approach. It is particularly suitable for handling text in natural, complex environments and supports over 80 languages [23].

1. **Text Detection (CRAFT):** To locate text regions, EasyOCR utilizes the Character Region Awareness for Text Detection (CRAFT) algorithm. CRAFT takes use of Convolutional Neural Network (CNN) to estimate the probability of a pixel belonging to an individual character and to evaluate the adjacency between characters to group them into cohesive words. To generate the region scores, CRAFT maps bounding box annotations to a 2D Gaussian distribution. For a pixel at coordinates (x, y) relative to a character center (x_c, y_c), the Gaussian heatmap is formulated as:

$$G(x, y) = \exp\left(-\left(\frac{(x - x_c)^2}{2\sigma_x^2} + \frac{(y - y_c)^2}{2\sigma_y^2}\right)\right). \tag{2.5}$$

While CRAFT's Gaussian heatmaps are highly effective for standard fonts, the disconnected nature of dot-peen indentations can cause the affinity score map to fragment, mistaking a single character for multiple distinct entities.

2. **Text Recognition (Convolutional Recurrent Neural Network (CRNN)):** Once bounding boxes are detected and cropped, the regions are fed into a CRNN. A CRNN backbone extracts visual feature sequences across the cropped region, while recurrent layers process these sequences to capture contextual dependencies.

3. **Connectionist Temporal Classification (CTC):** Finally, a CTC decoder translates the per-frame probability distributions into the final text sequence, removing the need to segment individual characters before classification. The CTC calculates the conditional probability of an alignment path $\boldsymbol{\pi}$ of length T given the input feature sequence $\mathbf{X}$:

$$p(\boldsymbol{\pi}|\mathbf{X}) = \prod_{t=1}^{T} y^t_{\pi_t}, \tag{2.6}$$

where $y^t_{\pi_t}$ is the probability of observing label π_t at time step t. The final probability of a target text sequence $\mathbf{l}$ is the sum of the probabilities of all valid alignment paths $\boldsymbol{\pi}$ that map to $\mathbf{l}$ (after applying a mapping function $\mathcal{B}$ that removes repeated characters and blank tokens):

$$p(\mathbf{l}|\mathbf{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{l})} p(\boldsymbol{\pi}|\mathbf{X}). \tag{2.7}$$

PaddleOCR

PaddleOCR is an open-source engine developed by Baidu. It represents a shift towards highly optimized, multimodal document AI. Unlike EasyOCR, which relies on a traditional CNN-RNN-CTC pipeline, modern iterations of PaddleOCR introduce significant structural advancements to maximize both speed and semantic accuracy [14].

- **Text Detection:** To locate text regions, PaddleOCR employs Differentiable Binarization (DBNet). Traditional segmentation networks produce a probability map and apply a static threshold to generate binary text regions. This discrete step function is non-differentiable, preventing end-to-end optimization. DBNet solves this by simultaneously predicting a probability map $\mathbf{P}_{i,j}$ and a dynamic threshold map $\mathbf{T}_{i,j}$. Binarization is then approximated using a differentiable step function:

$$\hat{\mathbf{B}}_{i,j} = \frac{1}{1 + e^{-k(\mathbf{P}_{i,j} - \mathbf{T}_{i,j})}}, \tag{2.8}$$

where $\hat{\mathbf{B}}_{i,j}$ is the approximate binary map and k is an amplifying factor. This allows the network to dynamically learn optimal binarization thresholds for varying lighting conditions and contrast levels.

It is expected that PaddleOCR handles extreme specular glare and low-contrast, which are the norm in the domain of this master's thesis, better than the other OCR engines, because DBNet dynamically learns optimal binarization, not like, for example Otsu's method.

- **Single Visual Model for Text Recognition (SVTR):** For character recognition, PaddleOCR replaces the recurrent Long Short-Term Memory (LSTM) layers found in traditional CRNNs with a SVTR. SVTR processes the cropped text images as a sequence of patches using a Vision Transformer. It utilizes a two-stage mixing mechanism. By eliminating the bottleneck of LSTM, SVTR accomplishes significantly higher parallelization and lower inference latency.

2.2 Multimodal Large Language Model (MLLM)

MLLMs represent a paradigm shift in artificial intelligence, extending the reasoning capabilities of traditional text-based Large Language Models (LLMs) to encompass visual, auditory, and dynamic spatial inputs. Utilizing billions of parameters, MLLMs bridge the gap between language understanding and visual perception, allowing the AI to achieve a comprehensive, human-like understanding of context. In this research, we aim to answer the question of whether SOTA Multimodal Large Language Models (e.g., Gemini 2.5 Flash, GPT-4o, InternVL3, Qwen2-VL) outperform traditional deep-learning OCR pipelines.

According to Liang et al. [24], MLLM's architecture utilizes specified NN components such as:

- **Transformer Backbone:** The core framework relies on self-attention mechanisms to dynamically focus on relevant parts of the input sequence, capturing complex, long-range dependencies in both sequential and parallel data.
- **Vision Encoders and Language Decoders:** MLLMs use separate, specialized modules to interface with raw data. Vision encoders (like CNNs or Vision Transformers) extract hierarchical features from images. Language decoders generate text autoregressively while simultaneously attending to these visual features, grounding their textual outputs in the visual context.
- **Multimodal Embedding:** Text and visual inputs are encoded separately and then projected into a shared, high-dimensional vector space. This aligns their

dimensionality and creates a joint representation, allowing the model to compute semantic relationships across modalities.

- **Multimodal Fusion and Cross-Attention:** A fusion module synthesizes the aligned visual and textual data. This relies heavily on cross-attention layers, which allow the model to dynamically focus on relevant aspects of one modality while processing the other. This enables fine-grained alignment, contextual understanding, and improved interpretability.

Gemini 2.5 Flash

Unlike older architectures that utilize a separate, bolted-on vision encoder to translate images into text before reasoning, the Gemini 2.X series is designed to be natively multimodal. This native integration allows the model to evaluate low-level spatial relationships, such as the disconnected dot-peen indentations, without losing visual fidelity during the encoding process. Furthermore, Gemini 2.5 Flash is specifically engineered to optimize the capability-compute Pareto frontier. The capability-compute Pareto frontier represents the optimal trade-off curve in artificial intelligence, defining the absolute maximum performance a model can achieve at a specific computational cost before requiring more resources to increase its accuracy. Because this thesis explicitly evaluates the computational token efficiency of various extraction strategies (as detailed in Experiment 2 in Section 5.2), Gemini 2.5 Flash serves as an ideal baseline for evaluating whether high-speed, structurally efficient MLLMs can reliably extract degraded dot-peen text without incurring the severe computational token overhead typical of heavier models [11].

GPT-4o

GPT-4o from OpenAI is an autoregressive "omni" model trained end-to-end across text and vision. Because all inputs and outputs are processed by the exact same NN, the model achieves highly robust cross-modal reasoning. In the context of industrial digitization, this unified architecture theoretically provides a tighter mapping between visual character geometry and semantic alphanumeric constraints. When a dot-peen character is severely occluded (such as the artificial scratches applied in Experiment 4 in Section 5.4), GPT-4o's cross-modal reasoning allows it to leverage the surrounding textual syntax to contextually infer the missing visual data [10]. This makes it a critical benchmark for testing zero-shot semantic reconstruction against traditional, optically rigid OCR engines like we do in Experiment 3 in Section 5.3.

InternVL3

InternVL3 represents a critical shift toward high-performance, open-weight multimodal architectures. Unlike proprietary cloud models (such as GPT-4o and Gemini 2.5 Flash) that require data to be transmitted via API, InternVL3 is open-weight, which provides a crucial advantage for secure industrial applications like localized deployment and the potential for domain-specific fine-tuning. While this thesis evaluates the model strictly on its baseline zero-shot inference capabilities (as detailed in Experiment 3 in Section 5.3), its architecture is fundamentally designed to support advanced post-training optimizations, which are out of the scope of this study and are reserved for future work. By utilizing an optimized training infrastructure and sophisticated visual reasoning capabilities, InternVL3 provides a highly competitive open-source alternative capable of decoding degraded industrial text, mitigating the severe data privacy concerns inherent to transmitting proprietary MRO asset databases to external servers [12].

Qwen2-VL

Qwen2-VL is another open-weight MLLM that introduces a Naive Dynamic Resolution mechanism, fundamentally altering how visual tokens are processed. Traditional vision encoders frequently force input images into fixed, squared resolutions, which can severely distort the aspect ratio of textual elements. By dynamically allocating visual tokens based on the image's native resolution, Qwen2-VL can process the heavily rectangular nameplate crops generated during the preprocessing stage of this pipeline without structural distortion. Furthermore, the model integrates Multimodal Rotary Position Embedding (M-RoPE) to effectively fuse 2D spatial positional information with 1D text tokens. In the context of industrial digitization, this helps the model to map the specific coordinate locations of disconnected dot-peen characters to their semantic alphanumeric sequence [13].

2.3 RANSAC

The Random Sample Consensus (RANSAC) algorithm is a robust parameter estimation approach specifically designed to process input data that contains a large proportion of outliers. Unlike traditional methods that utilize the entire dataset to compute an initial solution, RANSAC employs a resampling technique that randomly selects the strict minimum number of data points necessary to estimate the underlying model parameters. The algorithm iteratively generates candidate models and evaluates them against the rest of the dataset to identify inliers, which are data points that fit the proposed model within a predefined tolerance. To ensure a high probability (p typically

0.99) that at least one randomly selected sample set is completely free of outliers, the algorithm must run for a specific number of iterations (N) [25]. The underlying probability relationship is defined as:

$$1 - p = (1 - u^m)^N. \tag{2.9}$$

The required number of iterations N can be calculated as:

$$N = \frac{\log(1-p)}{\log\left(1-(1-v)^m\right)}, \tag{2.10}$$

where:

- u: The probability that any selected data point is an inlier.
- v: The probability of observing an outlier $(1 - u)$.
- m: The minimum number of points required to determine the model parameters.

RANSAC is utilized in the preprocessing steps of the pipeline to do the perspective rectification of the cut-out nameplate. The nameplate doesn't always have precise boundaries, because of the mask extraction method (using SAM 2), the non-perfect square architecture of the nameplate, or because of bolts. Standard edge-detection and line-fitting algorithms like Ordinary Least Squares (OLS) attempt to minimize the global error across all detected contour points. A single point acts as a high-leverage outlier, mathematically dragging the fitted boundary line askew and resulting in a tilted perspective crop [26]. RANSAC relies strictly on consensus. By drawing candidate lines through minimal point pairs, neglecting curves, and maximizing the inlier count, RANSAC completely ignores the severe geometric outliers.

2.4 Segment Anything Model 2 (SAM 2)

Segment Anything Model 2 (SAM 2) is a foundation model for solving promptable visual segmentation in images and videos. SAM 2 is designed to operate in two distinct segmentation modes. In its zero-shot Automatic Mask Generation mode, the model systematically grids the entire image to produce candidate masks for all visible objects simultaneously, requiring no human intervention. Conversely, in its Promptable mode, the architecture accepts explicit spatial constraints—such as a single pixel coordinate (point prompt) or a bounding box and produces only one mask. This dual capability allows the model to function autonomously or, when it fails, as a highly precise, human-in-the-loop extraction tool.

A critical challenge of the MRO domain is that the dirty metallic nameplate and the dirty turbocharger it is mounted on are made from identical, if not the same, material, have survived the same amount of wear and tear, and consequently share the same optical properties. Because of the absence of easily distinguishable semantic names, attempting to isolate them using traditional vision-language models (e.g., instructing an AI to "find the turbocharger nameplate") fundamentally fails. SAM 2 bypasses this semantic limitation by relying entirely on spatial and geometric cues.

To accurately process high-resolution static inputs, SAM 2 utilizes a hierarchical Hiera image encoder. This multi-scale architecture discards redundant background patches early in the processing pipeline and allows the model to allocate higher computational density to complex, low-level structural boundaries [27].

2.5 Regular Expression (Regex)

Traditional OCR and modern MLLMs can produce unstructured text outputs that need to be reliably filtered to fit into a predefined pattern. Originating from formal language theory and automata theory, a Regular Expression is a specialized sequence of characters that defines a precise search pattern [28]. A Regex engine processes a string deterministically, character by character, evaluating it against a rigidly defined sequence of literals and operators. The syntax leverages several core computational constructs to define allowable data shapes such as characters that match themselves exactly, metacharacters, negated classes, quantifiers, and boundaries. In this master's thesis (specifically Strategy B: Unconstrained Raw Transcription), Regex extracts the part numbers from the raw transcription and sanitizes them. For extraction, the specific pattern for Garrett turbocharger part numbers, defined as `\b\d{6}-\d{4}[A-Za-z]?\b`, instructs the engine to find exactly six digits, a literal hyphen, exactly four digits, and an optional terminal letter, all bounded as a standalone string. For sanitization, the Regex deletes any non-conforming characters.

2.6 Fuzzy Matching

Approximate string matching or fuzzy matching is utilized to quantify the morphological similarity between text strings, effectively mitigating inconsistencies such as typographical errors and OCR noise. The accuracy of raw exact-match evaluations is often insufficient. Instead, researchers rely on a spectrum of distance metrics tailored to specific error typologies:

- **Character Error Rate (CER) and Word Error Rate (WER):** The foundational benchmarking metrics in modern OCR research. CER calculates the minimum

number of character-level insertions, substitutions, and deletions required to match a ground truth string.

- **Damerau-Levenshtein Distance:** Includes transpositions (swapping two adjacent characters) as a single operational edit, which is critical for correcting common OCR sequencing errors.
- **Jaro-Winkler Distance:** A string comparison metric that specifically assigns a higher mathematical weight to matching characters at the beginning of a string (the prefix). This is highly relevant for structured industrial identification, where the prefix typically defines the primary asset category.

The **Levenshtein distance** is the foundational algorithm driving many of these metrics. It calculates the minimum number of single-character operations required to transform string **a** into string **b**. For strings of length $|\mathbf{a}|$ and $|\mathbf{b}|$, the recursive function is defined as:

$$\mathrm{lev}_{\mathbf{a},\mathbf{b}}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{\mathbf{a},\mathbf{b}}(i-1,j)+1 \\ \mathrm{lev}_{\mathbf{a},\mathbf{b}}(i,j-1)+1 \\ \mathrm{lev}_{\mathbf{a},\mathbf{b}}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{2.11}$$

Here, $1_{(a_i \neq b_j)}$ represents the indicator function, which equals 0 if the characters a_i and b_j are identical, and 1 if they differ (representing the substitution cost).

To establish a threshold for probabilistic matching, this absolute edit distance is normalized into a similarity score S ranging from 0 to 100 (where 100 represents an exact match) [29]:

$$S(\mathbf{a},\mathbf{b}) = \left(1 - \frac{\mathrm{lev}_{\mathbf{a},\mathbf{b}}(|\mathbf{a}|,|\mathbf{b}|)}{\max(|\mathbf{a}|,|\mathbf{b}|)}\right) \times 100. \tag{2.12}$$

2.6.1 The "Optical Blindness" of Standard Metrics

While standard metrics like the Levenshtein distance and CER are the gold standards for evaluating general text extraction, they share a fundamental flaw when applied to physical industrial environments: they are mathematically rigid and optically blind. These algorithms calculate the mathematical cost of a substitution, but they possess no semantic understanding of visual geometry.

In the developed automated matching part of the pipeline, fuzzy string matching is employed to query extracted part numbers against the master reference database

to compensate for inevitable character recognition errors. To address the limitation of optical blindness, a custom, visually weighted Levenshtein distance algorithm during the re-ranking phase is implemented. This heuristic function dynamically reduces the standard edit penalty by 80% when a substitution involves predefined sets of highly similar characters, such as confusing a "0" with an "O" or a "5" with an "S". This customized distance measure prevents common OCR mistakes from disproportionately degrading the match score.

2.7 Median Filtering

Median filtering, also known as median blur, is a non-linear digital image processing technique widely utilized for noise reduction, specifically targeting impulse noise such as salt-and-pepper degradation. In this research, we use it for the removal of oil and dust speckles, because they look a lot like salt-and-paper noise. The dot-peen characters are prone to being blurred because they are formed by small disconnected dots and look a lot like the salt-and-pepper noise. In this thesis, we test to what extent the median filtering can help with the readability of six tiers of differently degraded nameplates.

As defined by standard computational frameworks, the median filter runs through each element of the image signal and replaces each pixel with the median of its neighboring pixels, which are located in a predefined square neighborhood (or window) $\mathbf{W}$ around the evaluated pixel [30, 31].

Mathematically, for an input image $\mathbf{f}$, the filtered output image $\mathbf{g}$ at coordinates (x, y) is computed as:

$$\mathbf{g}(x, y) = \text{median}\{\mathbf{f}(x + i, y + j) \mid (i, j) \in \mathbf{W}\}. \tag{2.13}$$

With the foundational algorithms and computational models of the proposed pipeline established, the following chapter reviews the current state-of-the-art literature to evaluate existing approaches to industrial text extraction and identify the critical research gaps this thesis aims to bridge.

3 Literature Review

A detailed review of the SOTA technology and literature is presented in this section to address the research gap associated with this master's thesis.

3.1 Characteristics and Challenges of Industrial Text

Optical Character Recognition is one of the most popular technologies used to convert analog text into machine-readable documents. It is a foundational technology with extensive historical application across various domains. To address the time constraints and high error rates associated with manual logging of text on images, extensive literature has been dedicated to the automated reading of images and its limitations [32].

While traditional Optical Character Recognition (OCR) systems are highly effective on standardized, printed documents, their transition to uncontrolled industrial environments remains a fundamental computer vision challenge. As shown in the comprehensive review by Raisi et al. [6], real-world text detection suffers from several critical degradation typologies like perspective distortion, severe illumination reflection, and occlusion of characters. The authors note that despite advances in deep-learning architectures, existing models frequently underperform in these chaotic environments because they struggle to generalize to unseen, highly variable data. The paper also aligns to a great extent with the problems in MRO environments, where turbocharger nameplates are photographed via handheld smartphones, resulting in perspective distortion of the image, are made of highly reflective metal, which introduces specular glare, suffer from years of physical wear and oil stains, creating occlusion of characters, and the text on the nameplates consists of multiple engraved dots and not continuous strokes.

3.1.1 Direct Part Marking (DPM)

Direct Part Marking (DPM) is the industry term for dot-peen and laser engraving. Dragičević et al. [8] highlight dot-peen technology as a durable and cost-effective alternative to labels or tags for tracking components throughout their life cycle in environments with high levels of wear and tear. However, DPM creates a formidable barrier

to automated extraction via OCR. DPM characters are formed by multiple disconnected hemispherical indentations rather than continuous strokes and are associated with low contrast between text and background color and thick fonts. This frequently makes the OCR recognition imprecise, inaccurate, and susceptible to errors.

The master's thesis of Hannes Edvartsen [33] tackles the problem of reading low-contrast dot peen characters on metal. Results indicate that the deep learning model is significantly more robust, achieving 60% full-image accuracy compared to only 20% for the traditional method. Dating to 2018, this methodology predates the arrival of MLLMs and spatial foundation models like SAM 2. The images in the research are taken with a maximum camera tilt of 25 degrees, which cannot be guaranteed in the domain in which we are working. The limitations of this research also include that if dark shadows are present, the model fails.

3.1.2 Traditional OCR on Metal Surfaces

Both Rao et al. [34] and Lee et al. [35] established a functional baseline for character recognition on forged entities by combining YOLOv4 [36] for text localization with Tesseract [15] for character extraction. One of the limitations of these studies is that current industrial baselines rely heavily on rigid, traditional computer vision techniques for image preprocessing, such as Otsu binarization (typical for Tesseract and explained in Section 2.1) and morphological dilation, which are highly sensitive to lighting variations. Lee et al. [35] also propose specialized preprocessing steps like grayscale conversion and thresholding. These techniques fail catastrophically under extreme specular glare, severe occlusion (dirt and rust), and multi-axis skew combined with perspective distortions produced by handheld camera image capture, which is the norm in MRO environments. Similarly, Gao et al. [37] create a text spotting framework designed to accurately identify characters on curved metal surfaces. They also identify strong light reflections as a major source of interference that negatively impacts character features. Reflection interference remains a challenging question for future research. The study by Lee et al. [35] states its limits and says that further refinement is needed to prevent text recognition errors caused by small image sizes or the presence of mounting hardware like bolts.

Conversely, Vilasan et al. [38] develop a comprehensive AI-driven system that combines YOLOv7 [39] for object detection, Tesseract [15] for OCR, and a Residual Variational Autoencoder (ResVAE) [40] to accurately recognize characters and detect structural print defects in laser-engraved industrial nameplates. However, the pipeline is designed exclusively for quality control on pristine manufacturing lines, meaning it is entirely unequipped to handle real-world degradation, such as dirt or scratches on the text. An important finding is that while some models focus on merely detecting the

presence of a scratch [41, 42], far fewer attempt to semantically extract the text beneath it using a reference database.

To tackle these optical failures, some researchers have proposed hardware-level interventions, such as multi-directional illumination setups to eliminate glare [43] or "active cameras" that physically zoom and stitch high-resolution mosaics [44]. While effective in controlled laboratories, these hardware dependencies render the solutions completely unviable for decentralized MRO environments, where technicians rely solely on standard smartphone photography. This research relies entirely on a software-driven approach. While models like PaddleOCR [14] utilize dynamic binarization to handle standard lighting variations, extreme specular glare remains a formidable obstacle.

3.1.3 Automatic License Plate Recognition (ALPR)

The severe limitations of industrial OCR become especially apparent when contrasted with Automatic License Plate Recognition (ALPR) [45]. Numerous studies have successfully deployed multi-stage pipelines—utilizing YOLO paired with Tesseract [46], EasyOCR, or PaddleOCR [47, 48] — to extract vehicle license plates with over 96% accuracy.

Shashirangana et al. [49] utilize multi-stage methods. These systems process images in three distinct steps: license plate detection, character segmentation, and character recognition, and a single deep NN trained to perform end-to-end detection, localization, and recognition. Recent studies [47, 48] emphasize the importance of robust image preprocessing and plate localization, with the second study specifically integrating the YOLOv9 [50] algorithm for the initial detection phase. Ultimately, both research efforts concluded that PaddleOCR [14] consistently outperformed alternatives like EasyOCR [16], and Tesseract [15], achieving the highest reliability and character recognition accuracy even in challenging image conditions.

However, these successful results cannot be equivalently applied to the industrial MRO settings, as ALPR pipelines rely on structural advantages that turbocharger nameplates lack:

- **High Contrast:** License plates feature dark text engineered to stand out against bright backgrounds, whereas dot-peen turbocharger nameplates are inherently low-contrast, metal-on-metal surfaces.
- **Continuous Strokes:** License plates utilize standardized, continuous fonts, unlike the disconnected indentations of dot-peen engraving.
- **Universally Standardized Syntax:** Vehicle plates adhere to strict, expected regional formats, simplifying the downstream recognition task [49]. While specific

industrial components like Garrett turbochargers possess a strict, manufacturer-specific syntax, industrial nameplates as a whole lack a universal formatting standard.

Because industrial turbocharger dot-peen nameplates lack these broad visual and universally standardized structural advantages, deploying off-the-shelf ALPR-style pipelines in MRO environments yields unacceptably high error rates.

3.2 Spatial Standardization and Noise Reduction

3.2.1 Traditional Preprocessing

To overcome the severe environmental challenges of poor illumination and low contrast, early research frequently relied on extensive image preprocessing techniques. Patil and Dhanvijay [51] and Prakash et al. [52] developed robust OCR frameworks tailored to extract vehicle engine and chassis numbers utilizing morphological operations, filtering, and edge detection. Prakash et al. [52] demonstrated that raw smartphone images of engraved metal yield a poor baseline extraction accuracy (64.28%) when fed into legacy OCR engines like Tesseract and EasyOCR. By preprocessing the images beforehand, their system successfully increased extraction accuracy to approximately 90%.

However, their methodology relies on severe constraints that are impractical for real-world MRO environments, where images suffer 3D geometric distortions such as skewing. Their system required a physical grid attached to the camera to maintain a rigid, steady angle of view. Furthermore, their preprocessing required manual categorization and manual adjustment of lighting parameters.

Similarly, Yu et al. [53] proposed a machine vision-based recognition system utilizing optimized Laplace edge detection and binary image treatments to achieve high-efficiency contour extraction for engraved words. Building upon these foundational pipelines, Kronenberger et al. [54] introduced an automated tracking system designed to read unique identification strings engraved on industrial metal sheets. To overcome difficult lighting conditions, they developed a multi-step pipeline combining morphological operations for sheet detection and edge detection to isolate text areas, ultimately utilizing greyscale pattern matching to classify characters. The algorithm achieved a high classification accuracy of 97.56%, proving that reliable character recognition is possible in a factory environment by leveraging fixed font assumptions and correlation-based templates. However, those conditions cannot be achieved in decentralized MRO digitization.

3.2.2 Background Removal

Isolating the targeted text region from a visually complex background is a critical preprocessing step. Traditional background removal techniques frequently rely on temporal data to separate foreground objects from complex environments. As outlined in the comprehensive survey by Elhabian et al. [55], conventional algorithms separate foreground objects by modeling a static background over time and detecting object movement across sequential frames. Because turbocharger nameplates are captured as single, static smartphone photographs without the benefit of motion, temporal isolation is fundamentally impossible.

To overcome the need for multiple frames at different times, modern pipelines must leverage a spatial foundation model. While Segment Anything Model 3 (SAM 3) currently represents the SOTA for open-vocabulary segmentation, its prompt architecture struggles with engineered industrial parts, which are defined by subtle geometric differences rather than broad, recognizable semantic names [56]. In contrast, SAM 2 [27] interprets spatial cues strictly as geometric signals. By bypassing semantic reasoning, SAM 2 achieves a much higher sensitivity to low-level structural boundaries.

Recent empirical studies validate this architectural divergence. Sapkota et al. [57] demonstrate the failure of concept-driven segmentation on non-semantic objects, while Tang et al. [58] reinforce the necessity of geometry-conditioned models for industrial applications. Consequently, geometrically bound foundation models like SAM 2 are strongly preferred for the reliable, static isolation of metallic nameplates.

3.2.3 Salt and Pepper Noise Removal

A critical hurdle in automating optical extraction in MRO environments is the presence of surface contaminants like oil and metallic dust, which manifest as heavy impulse (salt-and-pepper) noise. Mitigating this is uniquely challenging because dot-peen typography consists of shallow, disconnected indentations. Standard cleaning algorithms frequently confuse these critical text dots with background speckles, destroying the data before extraction.

The traditional baseline for removing this noise is the Standard Median Filter (SMF), which replaces targeted pixels with the median value of their immediate neighbors. While the empirical study by Al-Azzeh et al. [31] shows that the SMF effectively reduces Root Mean Square Error (RMSE) and improves Peak Signal-to-Noise Ratio (PSNR) under mild conditions, they also demonstrate that the filter's quality degrades linearly as the noise ratio increases.

Chan et al. [59] reinforce this limitation, noting that when noise levels exceed 50%, traditional median filters indiscriminately alter pixels and severely smear structural

details. For industrial nameplates, this indiscriminate filtering means that heavy MRO degradation causes standard algorithms to irreversibly blur the critical dot-peen characters into the metallic background. Although Chan et al. (2005) propose a robust two-phase scheme for removing salt-and-pepper impulse noise, this method is unsuitable for this research because it assumes corrupted pixels take only the absolute maximum or minimum values within the dynamic range, which fails to adequately capture the continuous grayscale variations and complex shadowing typical of physical wear and tear on dot-peen components.

3.3 The Occlusion Problem and Semantic Reasoning

3.3.1 Physical Occlusion as Anomaly Detection

Rather than attempting to extract text through damage, many contemporary methodologies treat physical occlusion strictly as an anomaly detection problem. For instance, Wang et al. [60] developed an Occluded Boundary Representation Learning (OBRL) module to identify and localize physically occluded regions on identity documents. By tracking the distinct boundaries created by physical interference, their system successfully flags occluded documents as fraudulent. However, on turbocharger nameplates, physical degradations, such as metal scratches and dirt speckles, are not fraudulent anomalies, but unavoidable operational realities. Simply detecting that a scratch exists is insufficient. An architecture recognising the part number should extract the whole string despite the damage.

3.3.2 Failures of Geometric and Generative Restoration

Historically, the restoration of broken or partially occluded characters has relied heavily on geometric reconstruction. Boundary-based algorithms, such as the geodesic active contours deployed by Cohen et al. [61], successfully restore standard text by relying on predefined shape priors. However, because these boundary-based algorithms inherently rely on continuous strokes to bridge gaps, they fail to converge on the disconnected indentations characteristic of dot-peen typography. This limitation highlights why geometric reconstruction is unreliable for MRO environments, necessitating the shift toward semantic inference and database. This paper, as well as this master's thesis and many other works, relies strongly on a reference framework.

More recent approaches attempt to visually restore missing pixels prior to running OCR. To achieve this, generative methods such as patch-based inpainting [62], Generative Adversarial Networks (GANs) [63], and Stable Diffusion [64] extrapolate missing visual features from surrounding continuous textures or known semantic object

classes. For instance, patch-based restoration infers missing pixels by analyzing neighboring continuous strokes and regular spatial patterns. This foundational assumption completely breaks down on dirty industrial nameplates. Because dot-peen characters consist of disconnected indentations rather than continuous strokes, pixel-restoration algorithms cannot reliably reconstruct the text if a scratch obscures a critical sequence of dots.

While these generative models excel in other domains, such as Noh and Chang [64], utilizing a Stable Diffusion pipeline to reconstruct occluded human bodies via joint heatmaps and anatomical expectations, they are fundamentally misaligned with industrial text extraction. The SOTA approaches discussed in these studies [62, 63, 64] fail in this specific domain for two critical reasons. First, models trained to remove discrete semantic objects (e.g., a person blocking a view) cannot effectively classify or separate non-semantic, amorphous dirt from a metallic background [63]. Second, forcing a generative model to physically reconstruct disconnected dot-peen characters introduces a severe operational risk: the confident hallucination of entirely incorrect part numbers. Because both geometric and generative visual restorations fail under these strict optical constraints, extraction must rely on semantic inference that is deterministically grounded in a verified reference database.

3.3.3 Alphanumeric Lexicon Gap

Recognizing that traditional OCR struggles with physically occluded text, recent research frequently leverages Natural Language Processing (NLP) to infer missing characters. Across multiple studies, Mittal et al. [65] have explored various contextual prediction models to address this limitation. In one approach, they combined text detection with NLP to predict occluded characters based on surrounding grammatical context—a strategy similarly advanced by Das et al. [66] utilizing a CRNN. In a separate methodology [67], Mittal et al. paired the Google Vision API's global scene context with a 90,000-word BERT dictionary to infer missing words. Crucially, they explicitly noted that this system fundamentally fails if the target string is not present in the dictionary.

These methodologies reveal a critical gap for industrial asset tracking: there is zero linguistic context for alphanumeric strings. Standard NLP lexicons are insufficient because proprietary turbocharger part numbers (e.g., "758219-0005") will never exist in a generic English dictionary. Furthermore, visual scene context provides no analytical advantage. Identifying a background as "metal" or "machine" does not help an algorithm deduce whether a heavily scratched character is a '5' or a '6' [67]. Standard NLP architectures cannot predict strings they were never trained to spell, highlighting the absolute necessity of utilizing a domain-specific industrial database.

3.4 From Narrow AI to Multimodal Large Language Models (MLLMs)

Supervised machine learning has been utilized to read text from handwritten images and printed nameplates for a long time now [68]. Recent literature [69] discusses the current state of image recognition technology specifically applied to electric power equipment nameplates. This research identifies deep learning as the necessary future trajectory for industrial image recognition, decisively moving away from rigid traditional computer vision.

Although traditional OCR has served as the foundational technology for optical text extraction, the advent of MLLMs represents a profound paradigm shift in the field. Unlike legacy OCR systems, modern MLLMs, such as OpenAI's GPT-4o, Google's Gemini 2.5 Flash, and open-weight architectures like Qwen2-VL and InternVL3, fuse advanced visual encoders with massive LLMs to achieve native cross-modal understanding. This architectural change enables zero-shot semantic reasoning. Rather than merely detecting discrete geometric edges and shapes, these models can synthesize spatial visual context and structural patterns to infer missing or degraded characters. For instance, a MLLM can theoretically deduce a heavily scratched turbocharger part number by using the manufacturer-specific part number format rather than relying purely on visible pixels.

However, despite these theoretical semantic capabilities, a critical research gap remains: the comparative computational token efficiency and extraction accuracy of MLLMs on highly degraded, dot-peen nameplates remains completely unquantified in current literature. Consequently, this thesis addresses this gap by developing a systematically evaluated, software-driven pipeline that benchmarks MLLMs zero-shot extraction against traditional OCR engines in the domain of degraded nameplates. Ultimately, this research evaluates whether these generative models can reliably replace rigid optical transcribers when paired with a deterministic fuzzy-matching algorithm designed to ground the AI's semantic reasoning in a verified industrial reference database.

3.5 Token Consumption

Recent empirical findings highlight massive growth and distinct disparities in how tokens are consumed across tasks, languages, and modalities. Wang et al. [70] detail the immense scale of multimodal model development, noting that their InternVL3.5 model consumed 250 billion tokens during pretraining and 130 billion during finetuning, while successfully introducing a dynamic routing method to compress visual token

consumption by 50%, down to just 64 tokens per image patch. From a linguistic perspective, Kumar [71] tested GPT and Gemini tokenizers and demonstrated an inherent bias in commercial tokenizers that artificially inflates consumption for non-English users. Tracking real-world user behavior across models, including Qwen, GPT, and Gemini, Aubakirova et al. [72] reveal that average sequence lengths have more than tripled to over 5,400 tokens per request. This surge is driven largely by programming workflows that routinely demand massive contexts exceeding 20,000 input tokens, with over 50% of all tokens now being routed through multi-step, reasoning-optimized models. Ultimately, as Stryker et al. [73] emphasize in their comparative analysis of GPT-4 and Gemini, processing these escalating token volumes to support complex reasoning and multimodal generation continues to present significant computational and environmental sustainability challenges for modern AI systems.

3.6 Summary

In conclusion, the existing literature reveals a compounding series of limitations when attempting to automate optical text extraction in uncontrolled MRO environments. Traditional OCR architectures and standard ALPR pipelines rely heavily on continuous strokes, high-contrast environments, and constrained geometric preprocessing. That is why they fundamentally fail when confronted with degraded dot-peen typography and perspective distortions. Furthermore, standard digital interventions introduce unacceptable operational risks. Traditional median filters indiscriminately smear critical dot-peen geometries, modern generative models risk the confident hallucination of incorrect data, and standard NLP contextual models falter entirely when faced with proprietary alphanumeric sequences that lack linguistic context.

While the advent of MLLMs theoretically offers the advanced cross-modal semantic reasoning necessary to overcome these rigid visual limitations, a critical research gap remains in the current SOTA: the comparative extraction accuracy and computational token efficiency of these advanced models on highly degraded, dot-peen industrial nameplates remain completely unquantified.

Addressing this exact gap, this research provides the first quantified benchmark evaluating both the extraction accuracy and computational token efficiency of modern MLLMs against traditional OCR in this domain. Furthermore, rather than relying on error-prone generative pixel restoration or linguistically limited NLP dictionaries, this thesis introduces a novel, software-only architecture that successfully bridges geometry-conditioned spatial preprocessing with a custom, visually weighted fuzzy-matching engine. Ultimately, this hybrid approach overcomes the rigid limitations of traditional computer vision and the hallucination risks of generative AI, establishing a highly

reliable blueprint for automated asset tracking in uncontrolled MRO environments. The architectural design and experimental structure used to evaluate the pipeline developed in this thesis are detailed in the following Methodology chapter.

4 Methodology

To address the challenges of extracting and standardizing part numbers from severely degraded industrial nameplates, this research develops a pipeline and tests it by adopting a quantitative experimental approach. The stepwise evaluation aims to isolate the impact of spatial preprocessing, extraction architecture, and the postprocessing validation. This chapter outlines in Section 4.1 the data collection and reference materials utilized, in Section 4.2 the architecture of the proposed evaluation pipeline with its three stages, and in Section 4.3 the specific design of the five core experiments and the metrics used to evaluate system performance.

4.1 Data Collection and Reference Materials

4.1.1 Primary Real-World Image Dataset

The dataset consists of 127 real-world images captured with a handheld smartphone from an active real-world turbocharger repair workshop. In industrial MRO environments, a lack of data is the primary bottleneck for automation. While a sample size of 127 images is small for standard computer vision tasks, this dataset has been carefully curated to emphasize real-world relevance and practical applicability over sheer size. The images show various degradations that are scarce in sterile datasets, including minor scratches, dirt, rust, and oil. The angles and lighting of the images vary strongly, and the text is dot-peen on most of the images.

4.1.2 Synthetic Degradation Image Dataset

To systematically evaluate model robustness against specific typologies of industrial wear (addressing RQ3 in Section 1.2), a controlled, augmented dataset was generated. A subset of 40 preprocessed images (isolated nameplates with backgrounds removed and spatial preprocessing done) was selected from the primary real-world image dataset to establish a Tier 0 degradation. While these field-captured images inherently exhibit minor, nominal wear, they are operationally defined as the "clean" control group for these experiments.

A stochastic algorithm that utilized OpenCV [30] and NumPy [74] libraries was developed to artificially and gradually apply noise to the clean images. Scratches were constructed through stratified, overlapping line strands with randomized trajectories and structural dropouts to mimic deep metal gouges, while global dust was modeled by distributing variable-radius particles across the image and applying a Gaussian blur to seamlessly composite the noise onto the nameplate's surface.

The resulting dataset consists of six tiers, with an example provided in Figure 4.1:

1. **Tier 0 (Clean):** The original preprocessed clean images.
2. **Tier 1 (1 Scratch):** One artificial scratch on the part number.
3. **Tier 2 (Off-Target Occlusion):** One artificial scratch not on the part number.
4. **Tier 3 (2 Scratches):** Two artificial scratches on the part number.
5. **Tier 4 (Speckles):** Speckle noise across the whole image to simulate oil, dirt, and stains in the form of dots.
6. **Tier 5 (Compound Degradation):** A combination of 2 artificial scratches on the part number and speckles across the whole image.

The synthetic subset of 40 images is not intended for broad statistical generalization, but rather serves as a controlled, qualitative stress test to isolate and observe specific model failure mechanisms under ascending noise tiers. We add artificial 2D degradation as a simplified approximation of 3D geometry. Although this approach is not perfect and real-world damage consists of multiple visual variables, it's a good way to isolate specific occlusion types and precisely measure the visual decoders' failure thresholds.

4.1.3 Ground Truth Annotation Database

The 127 images from the primary real-world image dataset are labeled using PPOCR-Label [75]. The exported labels are structured as a hierarchical JSON dictionary where the image names serve as root keys, mapping to the exact four-point pixel coordinates of the nameplate and part number. The field `edited_pn` provides a standardized, zero-padded format of the part number. Zero-padding ensured the suffix is exactly four digits long.

4.1.4 Master Part Number Reference Database

The developed pipeline matches its predictions to a structured database with 7,871 historical Garrett part number entries, sourced from the same turbocharger repair workshop as the images. All of the suffixes of the numbers are zero-padded.

(a) Tier 0: Clean.

(b) Tier 1: One Scratch on Part Number.

(c) Tier 2: Off-Target Scratch.

(d) Tier 3: Two Scratches on Part Number.

(e) Tier 4: Speckles.

(f) Tier 5: Compound Degradation.

Figure 4.1: Visual Progression of Synthetic Degradation Applied to a Preprocessed Nameplate.

4.2 Proposed Pipeline Architecture

The pipeline consists of three stages, and each stage is described in its respective section.

1. **Image Preprocessing** in Section 4.2.1.
2. **Part Number Extraction** in Section 4.2.2.
3. **Postprocessing, and Database Matching** in Section 4.2.3.

4.2.1 Stage 1: Image Preprocessing

The image preprocessing consists of the steps background removal, perspective rectification, and orientation correction. The sequential flow of these operations is illustrated in Figure 4.2.

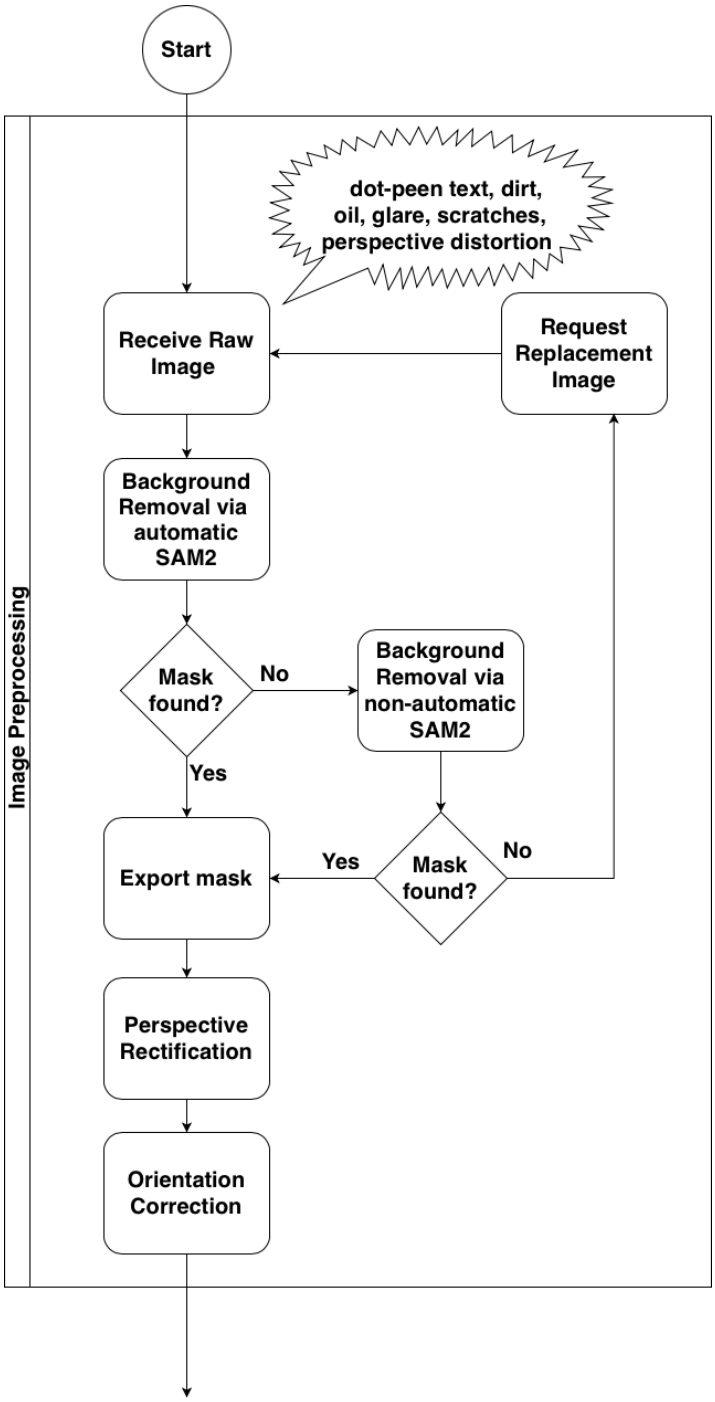


Figure 4.2: Flowchart of Image Preprocessing.

Step 1: Background Removal

Figure 4.3: Example of an Image from the Primary Real-World Image Dataset.

The first step in the developed pipeline is to isolate the nameplate from the visually complex background. This significantly reduces the size of the image that has to be processed, which avoids potential confusion from the text extraction engine and likely saves computing power (addressed in RQ1). To do that, a two-step pipeline utilizing SAM 2 and PaddleOCR is implemented. SAM 2 supports automatic mask generation on images. This function is used to produce candidate segmentation masks for all objects across the entire image. Concurrently, PaddleOCR is run one time on the entire image to detect text instances, yielding spatial bounding boxes, text strings, and confidence scores (filtered at a > 0.6 threshold). The algorithm calculates the geometric centroid of each valid bounding box and determines which SAM 2 mask it intersects. The mask containing the highest cumulative count of alphanumeric characters is selected as the optimal region of interest. If this pipeline doesn't produce any mask, or the resulting mask is too big, or too small, with a minimum threshold for width of 70 pixels and a maximum threshold of 1700 pixels, we use the standard SAM 2 image predictor without the automatic function. It uses a point from the nameplate as a prompt to calculate the mask. Figure 4.4 shows the nameplate from Figure 4.3 without background.

While SAM 3 represents the state of the art in semantic, open-vocabulary segmen-

tation, this master's thesis employs SAM 2, as empirical results and recent studies indicate that the concept-oriented design of SAM 3 can reduce performance in industrial settings. It's emphasized that vision-language foundation models perform well on broad semantic categories, but are fundamentally less suitable for industrial components. These parts often lack descriptive semantic names and are instead distinguished by subtle geometric differences that vision-language models have difficulty resolving. For example, if we prompt SAM 3 with "find the nameplate," we are likely to get a false prediction or no prediction at all. In contrast, SAM 2 is architecturally designed to interpret spatial cues as purely geometric signals, performing no internal semantic reasoning. Because SAM 2 optimizes for geometric tracking rather than multimodal semantic alignment, it exhibits a higher sensitivity to low-level structural boundaries [57, 58, 56].

Figure 4.4: Example of Nameplate with Removed Background.

Step 2: Perspective Rectification and Orientation Correction

Once the background is successfully removed, the isolated nameplates must undergo perspective rectification and orientation correction. These steps are critical because handheld smartphone photography inherently introduces multi-axis skew and projective distortion. A custom perspective rectification and OCR-driven orientation correction was developed. The process consists of three primary steps:

1. **Contour Extraction and Random Sample Consensus (RANSAC) Line Fitting:** As can be seen in Figure 4.5, the input image is binarized to isolate the largest external contour representing the target object. Contour points are clustered to

the four edges of a preliminary bounding box. To ensure robustness against edge noise and artifacts, RANSAC line fitting is applied to each cluster. The intersections of these four mathematically fitted lines dictate the precise corner coordinates of the object (in yellow).

2. **Perspective Warping:** Using the calculated corners, a perspective transform is applied to the original image. This flattens and perfectly crops the region of interest into a standardized rectangle, resolving any skew or tilt present in the original photograph.

3. **PaddleOCR-Driven Orientation Fixing:** The previous step rotates the image so that it lies on its longest side using minimum degree rotations. However, the nameplate is a rectangle and has 2 longest sides, which means that it can still be upside down (rotated 180 degrees). An algorithm using PaddleOCR is implemented to run text detection on both the produced cut-out image and a rotated version. It sums up the total count of recognized alphanumeric symbols for both orientations and outputs the version with the highest symbol count as the correct, upright final output.

As an example, after all preprocessing steps are executed, the image from Figure 4.3 is transformed into Figure 4.6.

4.2.2 Stage 2: Part Number Extraction

After the preprocessing stage is completed, the next stage is the Part Number Extraction. This stage is shown in the upper part of the flowchart in Figure 4.7. The text extraction phase is responsible for translating the optical data from the whole or preprocessed nameplate images into machine-readable strings. To compare the performance across different SOTA technologies, the pipeline is designed to be able to integrate both traditional Optical Character Recognition (OCR) systems and Multimodal Large Language Models (MLLMs).

Traditional OCR

To establish a baseline for standard text recognition capabilities, three deterministic OCR engines were deployed: Tesseract, EasyOCR, and PaddleOCR. These systems operate without contextual prompting, attempting to extract all text within the provided image as described in the Literature Review Section 2.1.

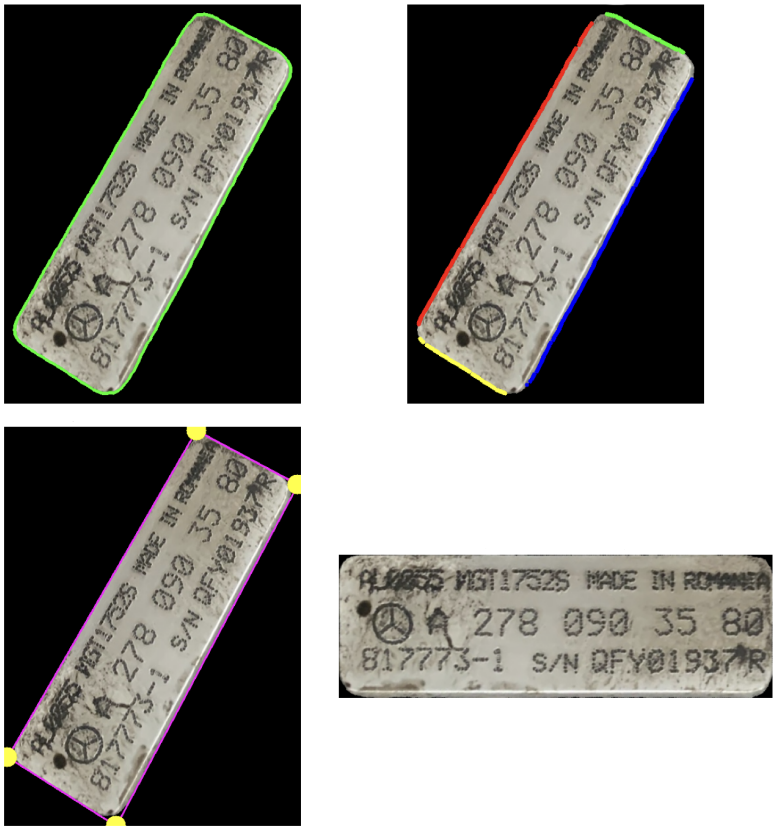


Figure 4.5: Perspective Rectification Steps. **Top-left:** Extracted Contour. **Top-right:** Straight Edge Clusters (Curves Ignored). **Bottom-left:** Virtual Trapezoid (Intersections). **Bottom-right:** Final Straightened Image.

Figure 4.6: Example of a Fully Preprocessed Nameplate.

Multimodal Large Language Models (MLLMs)

To evaluate the zero-shot reasoning and visual decoding capabilities of modern generative architectures, four MLLMs were utilized.

- **Proprietary Cloud Models:** GPT-4o and Gemini 2.5 Flash, accessed securely via

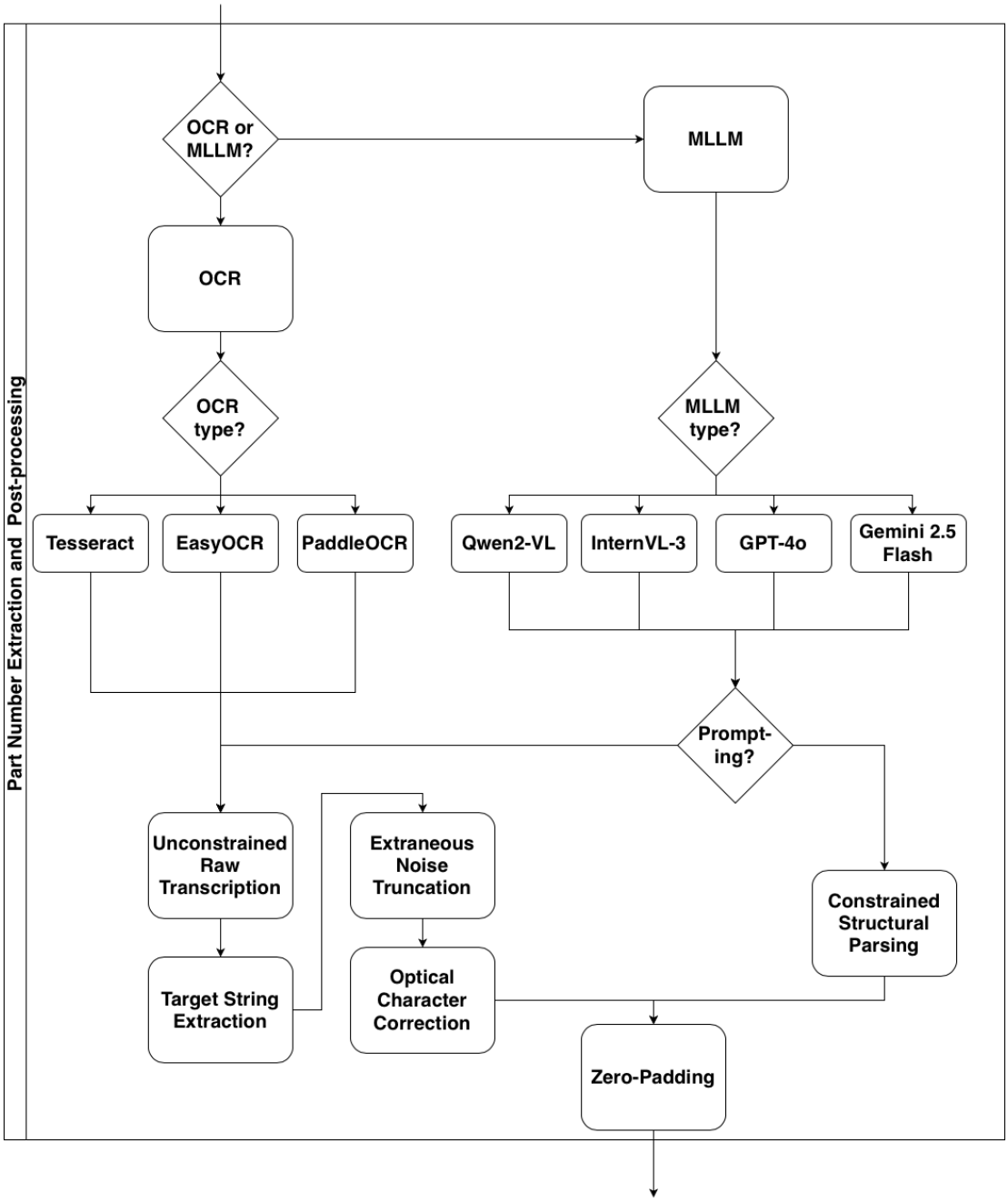


Figure 4.7: Flowchart of Part Number Extraction and Postprocessing.

their respective REST APIs.

- **Open-Weights Local Models:** Qwen2-VL and InternVL3, deployed locally utilizing the Hugging Face `transformers` library [76, 77] to ensure strict control over inference parameters and token logging.

More information about the engines can be found in the theoretical Background chapter in Section 2.2.

Prompting Strategies for MLLMs

Unlike traditional OCR systems, MLLMs allows for explicit natural language instructions. To directly address RQ1, the MLLMs were tested under two distinct instructional system prompts, which we will refer to as strategies:

Strategy A: Constrained Structural Parsing

This prompt shifts the burden of finding and formatting part numbers from the nameplate onto the MLLMs itself. By embedding domain-specific heuristic rules regarding the formatting of Garrett turbocharger part numbers, the model was forced to actively identify, extract, and structure the part number. The system prompt dictated a strict JSON output format:

System Prompt: Constrained Structural Parsing

```
Analyze this Garrett turbocharger nameplate and extract the information
into a strict JSON format.

Follow these two rules:
1. "raw_text": Extract all visible text from the nameplate. Keep it
exact, with no introductory text, descriptions, or added formatting.
The text on this nameplate consists ONLY of Latin letters, numbers,
hyphens (-), and slashes (/). You are strictly forbidden from
outputting any other symbols like Cyrillic letters or punctuation.

2. "part_number": Extract ONLY the Garrett Part Number. It must follow
the format of 6 digits, a hyphen (-), and 1 to 4 digits (e.g., 758219-3).
It MUST start with the digit 4, 7, or 8. If you cannot find a matching
sequence, return null.
```

```
CRITICAL: Return ONLY valid JSON. Do not include markdown backticks (‘‘‘)
or any other conversational text.

Expected format:
{
  "raw_text": "all the text from the plate goes here...",
  "part_number": "XXXXXX-X..."
}
```

Strategy B: Unconstrained Raw Transcription

This prompt was designed to utilize the MLLMs purely as an advanced OCR engine. The model has to extract all text from the nameplate with some limitations. As can be seen in Figure 4.7, we perform target string extraction, extraneous noise truncation, and optical character correction as postprocessing (more in Section 4.2.3). Raw outputs from this strategy rely entirely on the downstream Regex module to locate the specific part number in the extracted raw text. The system prompt is:

System Prompt: Unconstrained Raw Transcription

```
Extract all text from this turbocharger nameplate.
Maintain the line structure.
If you see serial numbers or part numbers, ensure they are exact.
CRITICAL: The text on this nameplate consists ONLY of Latin letters,
numbers, hyphens (-), and slashes (/). You are strictly forbidden from
outputting any other symbols like Cyrillic letters or punctuation.
Return ONLY the extracted text, no introductory sentences.
```

4.2.3 Stage 3: Postprocessing, and Database Matching

Both Strategy A and B of Step 1 are illustrated in the lower portion of Figure 4.7.

Step 1 (Strategy A).: Postprocessing for (Strategy A) Prompt-Guided Extractions

When we use MLLMs with rule-based part number extraction as described in Section 4.2.2, the output we receive in the JSON is structured. That is why the postprocessing that can be applied is minimal. The zero-padding algorithm is applied before the suffix of the part number. This algorithm ensured the suffix is exactly four digits long (e.g., standardizing "454232-5" to "454232-0005"). The formatting of the nameplate often does not include these "0"-s, but the entries of the Master Part Number Reference Database, as described in Section 4.1.4, have them. Once the postprocessing is complete, the string is passed to the Database Grounding and Re-ranking (see Section 4.2.3).

Step 1 (Strategy B): Postprocessing for Unconstrained Raw Transcription (Strategy B) and Traditional OCR

When utilizing traditional OCR engines like Tesseract, EasyOCR, and PaddleOCR, and also when the MLLM is instructed in the prompt to perform like OCR, and extract just raw text, like in Section 4.2.2, the text extraction leads to an unstructured block of alphanumeric characters. In this block, we have to find the part number, preprocess it, and pass it to the Database Grounding and Re-ranking step as described in Section 4.2.3. That is why a multi-step Regex- pipeline is required.

1. **Target String Extraction:** The pipeline utilized priority-based Regex to search the raw text block for specific length patterns characteristic of Garrett (e.g., searching for 6 digits followed by a hyphen).

2. **Extraneous Noise Truncation:** In the specific domain we are operating in, part numbers are often on the same line as serial numbers. As the beginning of the serial number is often marked with "S/N", the different engines we use often confuse those symbols as part of the part number sequence. That is why algorithms were applied to detect and strip suffixes attached to the extracted string, such as "S/N", "S/M", or "/N".

3. **Optical Character Correction:** OCR and MLLMs without added context in the prompt can misclassify visually similar characters on degraded dot-peen surfaces. That is why a deterministic substitution function was applied. We know that a part number consists only of numbers and corrected common optical letter-to-number hallucinations (e.g., replacing "O" with "0", "S" with "5", "Z" with "2", "A" with "4", and "B" with "8").

4. **Formatting:** Finally, similar to the prompted pipeline, we use the zero-padding algorithm to ensure that all part numbers are in the same format.

Step 2: Database Grounding and Re-ranking

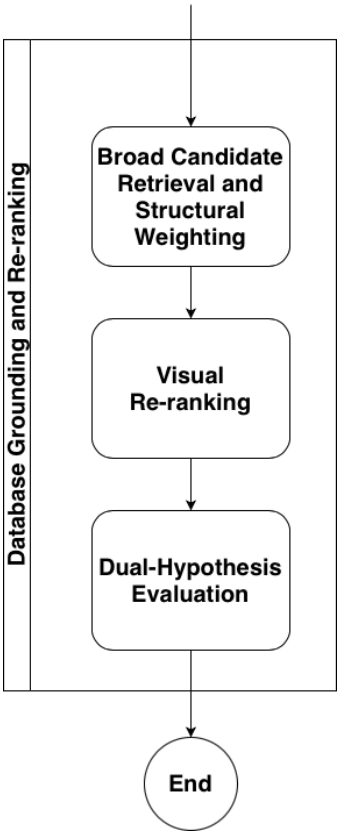


Figure 4.8: Flowchart of Database Grounding and Re-ranking.

Once the part numbers are extracted and formatted, the pipeline proceeds to the database matching steps. The standardized predictions are queried against the Master Part Number Reference Database (see Section 4.1.4) with 7,871 entries. A custom multistage visual similarity engine that uses re-ranking was developed. The algorithm operates through three primary mechanisms, as shown in Figure 4.8:

Broad Candidate Retrieval and Structural Weighting:

To maintain computational efficiency across the 7,871-entry database, the system utilizes the `rapidfuzz` library [78] to calculate the Levenshtein distance, retrieving the most

mathematically similar candidates. These candidates are then split into their prefix and suffix components based on the hyphen. A weighted scoring algorithm evaluates these components independently, applying 75% importance to prefix accuracy and 25% to suffix accuracy. The 75/25 ratio is mathematically grounded in the information density of the total ten-digit part number (comprising a 6-digit prefix and a 4-digit zero-padded suffix). In cases where the suffix contains a single non-zero digit, the core identifier consists of 7 meaningful characters, where the suffix represents only $\frac{1}{7} \approx 14.3\%$ of the total data. However, when the suffix contains two non-zero digits, the meaningful identity expands to 8 total digits, with the suffix contributing exactly $\frac{2}{8} = 25\%$ of the identifying information. By selecting a 75/25 split rather than an 80/20 ratio, the algorithm maintains full mathematical sensitivity to this second meaningful suffix digit, ensuring that minor manufacturing variations are not undervalued during the matching process.

Visual Re-ranking:

The top candidates are passed through a custom scoring function. The algorithm iterates through the extracted string and the candidate string character by character. If a mismatch occurs, the engine checks if the two characters belong to a predefined set of known visual confusions (e.g., {8, O} or {5, 6} or {1, 4} or {2, 7}). If the mismatch is a known visual confusion, the penalty applied to the similarity score is algorithmically forgiven by 80%. This forgiveness rate was established through empirical hyperparameter tuning to account for predictable optical swaps common on reflective metallic nameplates. If the mismatch falls outside of the sets, a standard 100% character penalty is applied.

Dual-Hypothesis Evaluation:

For images with more than 1 digit at the suffix, the engine evaluates 2 distinct part number hypotheses. As already explained in Section 4.2.3, often there is an "S/N" after the part number that the text extraction engine could confuse with a number or not read at all. Because "S" looks similar to "6", an extracted example string "454232-1 6/N" can be the part number "454232-1" or "454232-16". That is why the 2 possible part numbers with different suffixes are tested.

The part number with the highest score is selected as the final part number estimation. Then this number is compared to see if it matches exactly our Ground Truth (see Section 4.1.3). We use an exact match and not CER, because the part number serves as a discrete primary key and even a single-character discrepancy would prevent the system from

successfully querying the master reference database for critical maintenance protocols and replacement components.

4.3 Experiments

Five experiments are constructed to evaluate the pipeline. A summary of all of the experiments can be found in Table 4.1.

4.3.1 Experiment 1: Validation of Preprocessing Execution

To quantitatively validate the reliability of the preprocessing stage from Section 4.2.1 before continuing to the next steps of the pipeline, and use the preprocessed images in the next steps, an Intersection over Union (IoU) experiment was conducted. The four-point polygon coordinates of the SAM 2-predicted masks were exported to a JSON array and compared against the exact manually labeled four-point coordinates of the nameplate from the Ground Truth Annotation Database (Section 4.1.3).

The evaluation utilized the `shapely` Python library [79] to calculate the exact geometric overlap between the prediction and the Ground Truth. The IoU was calculated as the area of the intersection divided by the area of the union of the two polygons:

$$IoU = \frac{\text{Area}(\text{Polygon}_{\text{Pred}} \cap \text{Polygon}_{\text{GT}})}{\text{Area}(\text{Polygon}_{\text{Pred}} \cup \text{Polygon}_{\text{GT}})}. \tag{4.1}$$

A strict threshold was applied: a segmentation was only classified as successful if the $IoU > 0.50$, the standard threshold for bounding-box predictions in computer vision tasks. Only images that passed this validation threshold continued to the next steps.

To account for instances where the automatic mask generation yielded missing predictions or failed to meet the validation threshold, an interactive, point-prompted implementation of SAM 2 was deployed as a fallback mechanism. The success rate of this manual intervention was similarly evaluated using the $IoU > 0.50$ standard.

Finally, images that successfully passed the background removal stage were evaluated on the performance of their perspective rectification and orientation correction. This phase was assessed by tracking the overall execution failure rate, as well as observing the frequency of subsequent errors such as residual tilt and incorrect 180-degree upside-down rotations.

4.3.2 Experiment 2: Impact of Preprocessing and Text Extraction Strategy on Accuracy and Token Consumption

The goal of this experiment is to determine if the preprocessing (background elimination, perspective rectification, and orientation correction) and structured prompt engineering influence the downstream accuracy and token efficiency of the pipeline. This experiment addresses RQ1. In this experiment, Qwen2-VL will be utilized. As input data, the 127 not preprocessed images from the primary real-world image dataset (see Section 4.1.1) will be used alongside their successfully preprocessed counterparts from Experiment 1 (Section 5.1).

The experiment investigates the accuracy and token consumption in four distinct cases:

- Strategy A applied to whole images
- Strategy B applied to whole images.
- Strategy A applied to preprocessed images
- Strategy B applied to preprocessed images

Across all four cases, the final estimation of the part number is calculated using Database Grounding and Re-ranking as described in Section 4.2.3. A part number prediction is considered correct if it matches perfectly the Ground Truth (from Section 4.1.3) part number. For each of the cases, the tokens used for the text extraction and the part number accuracy are measured to determine the most efficient pipeline architecture.

4.3.3 Experiment 3: Benchmarking of Traditional OCR versus MLLMs

To address RQ2, we structured an experiment that evaluated the accuracy of our pipeline using 3 OCR engines, 2 proprietary MLLMs and 2 open-weight MLLMs. In Experiment 2, we first established that preprocessing increases the accuracy and decreases the token consumption. For Experiment 3, we will use this finding and use the 121 spatially preprocessed nameplate crops (from Experiment 1) as the experiment data set. To ensure a standardized comparison, we use the raw text output from the OCR engines, and the output from all MLLMs using both the Constrained Structural Parsing method (Section 4.2.2) and Unconstrained Raw Transcription method (Section 4.2.2), plus their respective downstream postprocessing (see Section 4.2.3). The token consumption will also be measured for each of the cases, which utilize a MLLM.

For the text extraction, we use:

1. Tesseract
2. EasyOCR
3. PaddleOCR
4. OpenAI's GPT-4o
5. Google's Gemini 2.5 Flash
6. InternVL3
7. Qwen2-VL

A part number prediction (see Section 4.2.3) is considered if it matches the ground truth part number perfectly (see Section 4.1.3). The ultimate performance of each of the engines will be strictly defined by its achieved pipeline accuracy and its token consumption.

4.3.4 Experiment 4: Stress Testing with Artificially Added Degradation

The final experiment addresses RQ3 by stress testing the optimized pipeline against six ascending levels of artificially added degradation. With this experiment, we aim to show to what levels of dust and scratches our pipeline is effective and how we can limit their negative influence. We use our Synthetic Degradation Image Dataset with 40 successfully isolated, nameplate cutouts (established in Section 4.1.2). The simulated noise incorporates deep structural scratches that obscure one or more digits of the identification number, off-target scratches to test spatial distraction, and global dot-like speckles imitating heavy dust and oil stains. Because dot-peen typography relies on the algorithmic clustering of disconnected indentations, introducing artificial dot-like speckles fundamentally challenges the visual decoder's ability to separate signal from noise. The following text extraction engines were utilized to process and evaluate all six tiers of the dataset (clean, 1 scratch, off-target scratch, 2 scratches, speckles all over the nameplate, and compound degradation consisting of 2 scratches and speckles):

- **Qwen2-VL with Unconstrained Raw Transcription:** The best prompting strategy identified in Experiment 2 (Section 5.2) and the best-performing MLLM architecture identified in Experiment 3 (Section 5.3).
- **PaddleOCR:** The best-performing OCR identified in Experiment 3.
- **GPT-4o with Constrained Structural Parsing** is the best performing MLLM architecture, while utilizing Constrained Structural Parsing, identified again in Experiment 3.

Furthermore, the median filter is introduced to the pipeline to remove noise from the nameplate. We use this approach because it is a non-linear digital filtering technique specifically proven to remove salt-and-pepper noise, like the speckles [31, 59]. The chosen kernel size is 7x7, which was determined through empirical hyperparameter tuning designed to optimize the critical trade-off between noise and the characteristics of dot-peen morphology. Then Qwen2-VL with Unconstrained Raw Transcription is used to evaluate how the median filter applied on all tiers from the database (see Section 4.1.2) influences the accuracy of the predictions of the whole pipeline. Finally, to evaluate whether the median filter can be deployed universally as a preprocessing step, the blurring is applied across the entire dataset of 121 successfully preprocessed real-world images (from Section 5.1) to measure its aggregate impact on general pipeline accuracy.

4.3.5 Experiment 5: Evaluation of Pipeline Computational Latency

The objective of this experiment is to quantify the computational latency of the optimized extraction pipeline. While previous experiments established the system's accuracy and token efficiency, this experiment evaluates the processing time required for each of the four distinct pipeline stages to identify computational bottlenecks. The distinct stages are:

- Background Removal
- Perspective Rectification and Orientation Correction
- Text Extraction
- Postprocessing, Database Grounding, and Re-ranking

To ensure standardized and reproducible time measurements, all latency benchmarking was recorded in seconds using a Google Colab environment equipped with an NVIDIA L4 Tensor Core GPU (24 GB VRAM). The experiment analyzes the optimal pipeline architecture identified in Experiments 2 and 3: spatial preprocessing paired with Qwen2-VL with Unconstrained Raw Transcription strategy (Strategy B) alongside its corresponding postprocessing. The latency was measured by calculating the average inference time across a total of 5 images. To isolate the true operational latency of the pipeline, environmental setup, dependency installation, and initial model weight loading times were explicitly excluded from the final measurements.

4.3.6 Summary of Experimental Framework

Table 4.1: Summary of Experimental Framework.

Research Question	Experiment	Dataset Used	Engines & Prompts	Target Metric(s)
Prerequisite Validation (Establishes pipeline viability).	**Experiment 1:** Validation of Preprocessing Execution.	Primary Real-World Image Dataset (127 images) & Ground Truth Annotations.	**Engines:** N/A (Focus on spatial preprocessing). **Prompt:** N/A.	Intersection over Union (IoU $>$ 0.5); Preprocessing Execution Success Rate.
RQ1: Impact of spatial preprocessing and extraction strategy on accuracy and tokens.	**Experiment 2:** Impact of Preprocessing and Text Extraction Strategy.	Primary Real-World Image Dataset (127 whole images & 121 preprocessed images).	**Engine:** Qwen2-VL. **Prompts:** Strategy A (Constrained) & Strategy B (Unconstrained).	Exact Part Number Match Accuracy; Token Consumption.
RQ2: Benchmarking MLLMs against traditional OCR for accuracy and token efficiency.	**Experiment 3:** Benchmarking of Traditional OCR versus MLLMs.	Primary Real-World Image Dataset (121 spatially preprocessed images).	**OCR (No Prompt):** Tesseract, EasyOCR, PaddleOCR. **MLLMs (Strategy A & B):** GPT-4o, Gemini 2.5 Flash, InternVL3, Qwen2-VL.	Exact Part Number Match Accuracy; Token Consumption.
RQ3: System robustness against synthetic noise and the efficacy of median filtering.	**Experiment 4:** Stress Testing with Artificially Added Degradation.	Synthetic Degradation Image Dataset (6 tiers each with 40 images); Primary Real-World Image Dataset (121 spatially preprocessed images).	**Qwen2-VL:** Strategy B. **GPT-4o:** Strategy A. **PaddleOCR:** No Prompt.	Exact Part Number Match Accuracy (Standard vs. Median Filter).
Computational Latency (Evaluates computational bottlenecks).	**Experiment 5:** Evaluation of Pipeline Computational Latency.	Subset of 5 representative images.	**Engine:** Qwen2-VL. **Prompt:** Strategy B.	GPU Execution Time (seconds) per pipeline stage.

5 Results and Discussion

In this chapter, the proposed pipeline is stepwise validated by conducting five experiments, which are presented, and their results are discussed. The three research questions stated in Section 1.2 are answered in this chapter. Together, these results provide the empirical foundation necessary to determine the limits of deploying the pipeline in uncontrolled MRO environments.

5.1 Experiment 1: Validation of Preprocessing Execution

The first step of the developed pipeline is the preprocessing of the image as described in Section 4.2.1, which includes background removal, perspective rectification, and orientation correction. The two preprocessing steps utilize SAM 2, RANSAC, and PaddleOCR. The background removal step is validated against the coordinates of the nameplate in the Ground Truth Annotation Dataset (see Section 4.1.3). The goal is to show how well an automated script could reliably locate and crop the nameplate without human intervention and how a simple human prompt could influence it.

Table 5.1: Summary of Automated SAM 2 Segmentation Performance Evaluated on the 127-image Primary Real-World Image Dataset.

Indicator	**Measurement**
Missing Predictions	10
Failed Images (IoU ≤ 0.5)	8
Average IoU (Valid only)	0.83
Rate of Valid Masks (Total Valid Images Analyzed / All Images)	92.13% (117/127)
Rate of Successfully Produced Masks (Successful Images [IoU > 0.5] / All Images)	**85.83% (109/127)**

The results shown in Table 5.1 explain that the automated background removal is highly effective, with 85.83% success rate in producing correct masks. There are 8 failed images, where the IoU ≤ 0.5. This indicates that the overlap between the predicted

segmentation mask and the manual annotations from the Ground Truth Annotation Database (see Section 4.1.3) was less than 50%. Typically, this failure occurs due to over-segmentation, where the model incorrectly isolates the entire turbocharger assembly rather than just the target nameplate. There are 10 images with missing predictions. That means that from 127 valid images, in only 10 there was no prediction generated, achieving a success rate of 92.13%. From the predicted 117 masks, 109 were correctly predicted with an Average IoU of 0.83.

The masks of the Failed Images (8) and Images with Missing Predictions (10) were successfully computed using the standard SAM 2 image predictor without the automatic function. In this type of SAM 2, the human has to select a point from the nameplate as a prompt, and then the mask is generated. In the conducted experiment, this boosts the success rate of the background removal step to 100%. All of the images can be successfully processed using this non-automatic SAM 2.

Of the 127 images that successfully passed the background removal stage, 121 were processed by the perspective rectification and orientation correction step described in Section 4.2.1. This achieved a **95.28%** execution success rate, with 6 images failing the process entirely. These 6 failures occurred because the irregular, non-rectangular shape of the nameplates prevented the RANSAC algorithm from finding the four straight edges required for rectification. We use this preprocessed set of images for the next experiments. However, a subsequent review of the 121 processed images revealed that the perspective rectification was not flawless, as a subset retained residual tilt, and 12 images were erroneously finalized with a 180-degree upside-down rotation, failing the orientation correction step.

5.2 Experiment 2: Impact of Preprocessing and Text Extraction Strategy on Accuracy and Token Consumption

Table 5.2: Pipeline Accuracy across Prompting Strategies and Preprocessing Conditions.

	Whole Image		**Preprocessed Image**	
	Strategy A (Constrained)	**Strategy B** (Unconstrained)	**Strategy A** (Constrained)	**Strategy B** (Unconstrained)
Total Images Evaluated	127	127	121	121
Correct DB Matches	26	53	37	92
Incorrect DB Matches	101	74	84	29
Accuracy	**20.47%**	**41.73%**	**30.58%**	**76.03%**

Table 5.3: Token Consumption Comparison for Constrained vs. Unconstrained Prompting.

	Whole Image		**Preprocessed Image**	
	Strategy A (Constrained)	**Strategy B** (Unconstrained)	**Strategy A** (Constrained)	**Strategy B** (Unconstrained)
Total Images	127	127	121	121
Total Input Tokens	81,011	63,104	71,258	54,197
Total Output Tokens	8,264	6,411	12,028	6,834
Grand Total	**89,275**	**69,515**	**83,286**	**61,031**

This experiment has as a main goal to answer RQ1. It evaluates how spatial processing (specifically background elimination, geometric rectification, and orientation correction) described in Section 4.2.1 and the choice of extraction strategy (Constrained Structural Parsing versus Unconstrained Raw Transcription), described in Section 4.2.2, paired with their respective downstream postprocessing, described in Section 4.2.3, impact the text extraction accuracy and token usage on distorted dot-peen nameplates. The results from the experiment are shown in Tables 5.2 and 5.3. Qwen2-VL is employed to extract text from the Primary Real-World Image Dataset (see Section 4.1.1) using two distinct prompting strategies. While traditional OCR engines do not support prompt-based interaction, the open-weight architecture of Qwen2-VL enables precise, local logging of token consumption per request, which is a core metric for evaluating the efficiency of the proposed pipeline.

5.2.1 The Impact of Preprocessing on Accuracy

Preprocessing the image, by removing the background of the image, straightening it, and rotating it as described in Section 4.2.1, utilizing SAM 2, contour extraction and RANSAC line fitting, perspective warping, and PaddleOCR text orientation fixing, dramatically improved system performance across all metrics. While the preprocessing stage slightly reduced the sample size from 127 to 121 images due to pipeline failures, as already discussed in Experiment 1 in Section 5.1, it provided a definite advantage for Qwen2-VL. For Strategy B (Unconstrained Raw Transcription), preprocessing nearly doubled the accuracy of the pipeline, which went from 41.73% on whole images to a peak achieved accuracy for this thesis of 76.03% on preprocessed photographs. The accuracy of the pipeline using Strategy A (Constrained Structural Parsing) increases by around 10.11 percentage points (11 more correctly recognized part numbers) by preprocessing the images.

5.2.2 The Impact of Prompting Strategies on Accuracy

Another finding of this experiment is the performance gap between the two prompting strategies. Strategy B (Unconstrained Raw Transcription), paired with downstream Regex postprocessing, outperformed Strategy A (Constrained Structural Parsing) in both accuracy and token consumption across every test condition.

When the model was forced to actively identify and format the part number (Strategy A), it struggled heavily, achieving only 30.58% accuracy on preprocessed images. In contrast, allowing the model to just extract the raw text from the image (Strategy B) and relying on its Regex-dependent postprocessing (explained in Section 4.2.3) to locate and correct the part number achieved 76.03% accuracy.

5.2.3 The Impact of Prompting Strategy on Token Consumption

As anticipated, embedding rigid formatting constraints directly into the prompt (Strategy A) proved to use more tokens than simple transcription (Strategy B). On preprocessed images, Strategy A consumed a total of 83,286 tokens, compared to just 61,031 tokens for Strategy B. This is expected in the input side, as it is due to the fact that, if the prompt is longer, the model has to process more complex logical instructions. However, the cost discrepancy also extends to the output tokens, as Strategy A forces the model to actively identify the part number and format the entire extraction into strict JSON. Strategy B puts all of the burden of identification and formatting of the part number to the downstream Regex module as described in Section 4.2.3.

5.2.4 The Impact of Preprocessing on Token Consumption

Preprocessing proved highly token-efficient as shown in Table 5.3. By eliminating the background noise and irrelevant visual context, the MLLM was able to focus more efficiently on the nameplate and required significantly fewer input tokens. For Strategy B, utilizing preprocessed images reduced total token consumption from 69,515 to 61,031 tokens, despite the vastly superior accuracy. For Strategy A, the token grand total was decreased to 83,286 from 89,275. While spatial preprocessing successfully reduced the input token burden—dropping from 81,011 to 71,258 for Strategy A, as the model no longer had to process complex background noise—it caused a significant spike in output tokens. For Strategy A, output tokens jumped from 8,264 on whole images to 12,028 on preprocessed crops. A similar, though smaller, output increase occurred in Strategy B (6,411 to 6,834). This suggests that because the preprocessed images were cleaner and standardized, the visual decoder was able to successfully detect and read far more text. This inverse relationship between input and output tokens indicates that

when confronted with the severe optical noise of unedited whole images, the model frequently abandoned the extraction process early or output shorter, failed strings.

5.2.5 Summary of Experiment 2

The results reveal a clear hierarchy of performance, demonstrating that preprocessing and Unconstrained Raw Transcription prompting strategy and its corresponding preprocessing, when considered together or alone, increase the accuracy and decrease the token usage. The aggregate impact of choosing between preprocessing and not, and Strategies A and B, is significant. Notably, when utilizing Strategy B and preprocessing, compared to Strategy A without preprocessing, the accuracy increases by 55.56 percentage points, and the token usage decreases by 28,244 tokens. There is also the assumption that if it works so well with the MLLM Qwen2-VL, the results will be similar if another MLLM is used. It is favorable that the postprocessing of the better-performing Strategy B can also be used for OCR engines.

5.3 Experiment 3: Benchmarking of Traditional OCR versus MLLMs

Table 5.4: Benchmarking for Traditional OCR Engines.

	Tesseract	**EasyOCR**	**PaddleOCR**
Total Images	121	121	121
Correct Matches	7	29	65
Incorrect Matches	114	92	56
Accuracy	**5.79%**	**23.97%**	**53.72%**

Table 5.5: Benchmarking for MLLMs using Unconstrained Raw Transcription Strategy.

	InternVL3	**Qwen2-VL**	**GPT-4o**	**Gemini 2.5 Flash**
Total Images	121	121	121	121
Correct Matches	63	92	77	77
Incorrect Matches	58	29	44	44
Accuracy	**52.07%**	**76.03%**	**63.64%**	**63.64%**

Table 5.6: Token Consumption across MLLMs using Unconstrained Raw Transcription Strategy.

	InternVL3	**Qwen2-VL**	**GPT-4o**	**Gemini 2.5 Flash**
Total Images	121	121	121	121
Total Input Tokens	10,285	54,197	86,648	41,418
Total Output Tokens	8,656	6,834	3,638	5,262
Grand Total	**18,941**	**61,031**	**90,286**	**46,680**

Table 5.7: Benchmarking for MLLMs using Constrained Structural Parsing.

	InternVL3	**Qwen2-VL**	**GPT-4o**	**Gemini 2.5 Flash**
Total Images	121	121	121	121
Correct Matches	56	37	78	77
Incorrect Matches	65	84	43	44
Accuracy	**46.28%**	**30.58%**	**64.46%**	**63.64%**

Table 5.8: Token Consumption across MLLMs using Constrained Structural Parsing.

	InternVL3	**Qwen2-VL**	**GPT-4o**	**Gemini 2.5 Flash**
Total Images	121	121	121	121
Total Input Tokens	27,346	71,258	103,104	59,760
Total Output Tokens	11,042	12,028	5,921	8,760
Grand Total	**38,388**	**83,286**	**109,025**	**68,520**

To answer RQ2, a comprehensive benchmarking was conducted to determine the extent to which SOTA MLLMs (e.g., InternVL3 and Qwen2-VL, GPT-4o, Gemini 2.5 Flash) with and without constrained prompting outperform traditional deep-learning OCR pipelines (e.g., Tesseract, EasyOCR, PaddleOCR) in terms of accuracy of the developed pipeline. For each prompting strategy, the respective downstream postprocessing is utilized. To fully answer RQ2, the token consumption of all text extraction engines is analyzed.

For better understanding, a plot with all of the compared text extraction engines with their accuracy and token consumption from both Experiment 2 and Experiment 3 can be found in Figure 5.1.

5.3.1 The Traditional OCR

The results (see Table 5.4) highlight the severe optical challenge posed by degraded dot-peen typography. As it was suspected in the literature review in Section 3.1.2 and in the theoretical background in Section 2.1, where the architecture of the Tesseract was described, Tesseract failed almost entirely, achieving only 5.79% accuracy (7 out of 121 correct matches), which is also the lowest pipeline accuracy achieved in this thesis. Dot-peen characters lack a polygonal outline because they consist of multiple dots near each other. This architectural reliance on continuous edges explains why Tesseract fundamentally struggles with DPM typography. EasyOCR performed marginally better by utilizing an end-to-end deep learning approach, but it still yielded a low accuracy rate (23.97%) because its text detection heatmaps fragmented on disconnected dot-peen indentations. PaddleOCR emerged as the most robust OCR engine, by employing differentiable binarization to dynamically adapt to varying lighting conditions, achieving a 53.72% accuracy rate (65 correct matches). However, these results show that using OCR even with pre- and postprocessing steps is insufficient for highly reliable industrial deployment.

5.3.2 The Impact of Prompting Strategy on MLLMs Accuracy

As seen in Table 5.5, the implementation of Multimodal Large Language Model (MLLM) led to an increase in the pipeline accuracy, but the data reveals that performance is heavily dictated by the chosen prompting strategy.

When utilizing the Unconstrained Raw Transcription strategy (Strategy B) (see Section 4.2.2) plus downstream Regex (see Section 4.2.3), the open-weight model Qwen2-VL performed the best. As shown in Experiment 2 in Section 5.2, the pipeline achieves a peak accuracy of 76.03% (92 out of 121 correct matches), the highest in this master's thesis. The proprietary models, GPT-4o and Gemini 2.5 Flash, both performed at a

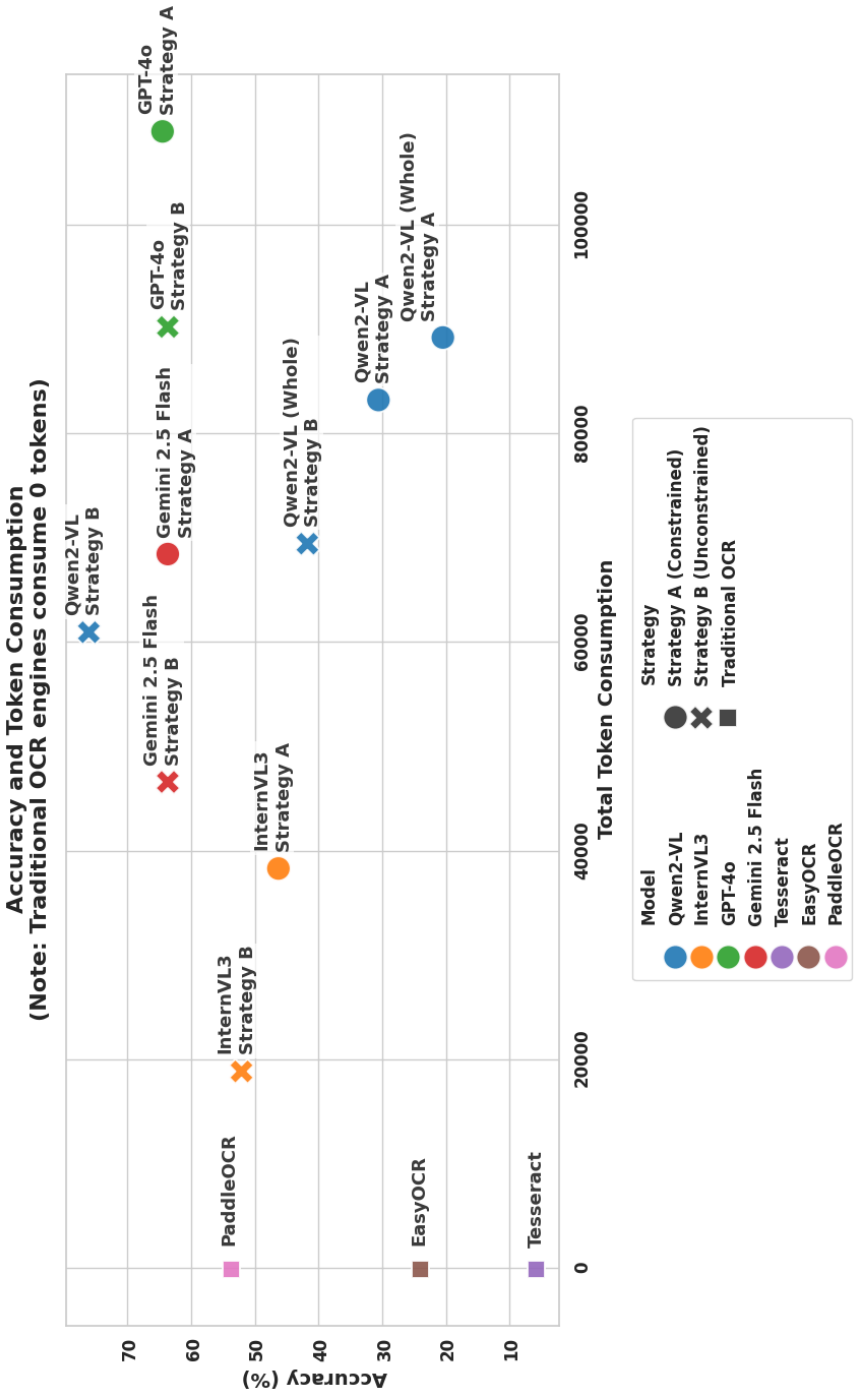


Figure 5.1: Accuracy and Token Consumption.

respectable, but with 12.39 percentage points lower 63.64%. Interestingly, the open-weight InternVL3 (52.07%) performed slightly worse than PaddleOCR (53.72%).

Applying the Constrained Structural Parsing prompt fundamentally changed the performance hierarchy among the models (Table 5.7). Giving rigid formatting instructions to both open-weight models made them decrease their performance. InternVL3 accuracy went from 52.07% down to 46.28% and Qwen2-VL lost more than half of its accuracy and went from the best-performing model with 76.03% to the worst, while using this prompting mechanism, with just 30.58%. Constrained prompting approach overloads the reasoning capabilities of models like Qwen2-VL and InternVL3. When these models are forced to simultaneously decode highly degraded visual data and adhere to strict JSON formatting rules, their cognitive load is exceeded, which severely cripples extraction accuracy and inflates token consumption.

The proprietary models showed that they are resilient to the choice of prompting strategy and are the only models that have accuracy more than 50% using the constrained prompting strategy. GPT-4o had one more correct match, so its accuracy slightly improved to 64.46%, while Gemini 2.5 Flash remained perfectly stable at 63.64% as seen in Table 5.7. Because these foundational models undergo extensive fine-tuning specifically for complex instruction following and structured data generation, they can process rigid formatting constraints without the severe performance drop observed in their open-weight counterparts.

5.3.3 Token Consumption

As already confirmed in Experiment 2 in Section 5.2, a longer prompt increases the token consumption across all models. As seen in Tables 5.6 and 5.8, GPT-4o proved to be the most computationally expensive architecture both when using a constrained and unconstrained prompt, followed by Qwen2-VL, Gemini 2.5 Flash, and InternVL3 as the most efficient model. GPT-4o's token usage jumped from an already high 90,286 tokens (unconstrained) to a massive 109,025 tokens (constrained). Qwen2-VL similarly saw its total token count inflate from 61,031 to 83,286 under constraints. InternVL3 utilized the fewest tokens overall (18,941 unconstrained), though this came at the cost of poor accuracy. The "capability-compute Pareto frontier" introduced regarding Gemini 2.5 Flash in Section 2.2 is empirically validated in this experiment. Under Unconstrained Raw Transcription, both Gemini 2.5 Flash and GPT-4o achieved an identical 63.64% accuracy. Yet, Gemini accomplished this using only 46,680 tokens—nearly half of GPT-4o's massive 90,286 token consumption. The difference in token usage between GPT-4o using a constrained prompt and Intern VL3 using an unconstrained prompt is almost 6 times, and the accuracy increases by less than 13 percentage points.

5.3.4 Summary of Experiment 3

The data provides a definitive answer to RQ2. MLLMs significantly outperform traditional OCR, provided they are deployed with the correct strategy. While proprietary models like GPT-4o offer greater resilience to complex prompt constraints, they do so at a severe computational cost. Ultimately, the optimal configuration for extracting degraded industrial part numbers is an open-weight architecture (Qwen2-VL) utilized purely as an unconstrained visual decoder. By offloading the rigid, deterministic formatting rules to a downstream Regex and fuzzy-matching module, this hybrid approach prevents reasoning overload within the generative model. This approach balances the highest system accuracy (76.03%) with moderate token efficiency (61,031 tokens).

5.4 Experiment 4: Stress Testing with Artificially Added Degradation

Table 5.9: Stress Testing across Synthetic Noise Levels with Qwen2-VL and Unconstrained Raw Transcription.

Tier *Degradation*	**Tier 0** *Clean*	**Tier 1** *1 Scratch*	**Tier 2** *Off-Target*	**Tier 3** *2 Scratches*	**Tier 4** *Speckles*	**Tier 5** *Compound*
Total Images	40	40	40	40	40	40
Correct Matches	31	25	29	18	21	15
Incorrect Matches	9	15	11	22	19	25
Accuracy	**77.50%**	**62.50%**	**72.50%**	**45.00%**	**52.50%**	**37.50%**

Table 5.10: Stress Testing across Synthetic Noise Levels with PaddleOCR.

Tier *Degradation*	**Tier 0** *Clean*	**Tier 1** *1 Scratch*	**Tier 2** *Off-Target*	**Tier 3** *2 Scratches*	**Tier 4** *Speckles*	**Tier 5** *Compound*
Total Images	40	40	40	40	40	40
Correct Matches	21	16	20	5	17	4
Incorrect Matches	19	24	20	35	23	36
Accuracy	**52.50%**	**40.00%**	**50.00%**	**12.50%**	**42.50%**	**10.00%**

Table 5.11: Stress Testing across Synthetic Noise Levels with GPT-4o and Constrained Structural Parsing.

Tier *Degradation*	**Tier 0** *Clean*	**Tier 1** *1 Scratch*	**Tier 2** *Off-Target*	**Tier 3** *2 Scratches*	**Tier 4** *Speckles*	**Tier 5** *Compound*
Total Images	40	40	40	40	40	40
Correct Matches	24	19	24	12	23	13
Incorrect Matches	16	21	16	28	17	27
Accuracy	**60.00%**	**47.50%**	**60.00%**	**30.00%**	**57.50%**	**32.50%**

The final experiment addresses RQ3 by stress-testing the optimized pipeline against increasing levels of artificially induced degradation. With this experiment, we aim to show the extent to which our pipeline is effective against dust and scratches, and how this noise can be reduced. We utilize our database with 40 synthetic noise images in six tiers, as described in Section 4.1.2. We use three text extraction engines. Qwen2-VL with Unconstrained Raw Transcription, because in Experiment 2 (Section 5.2) and Experiment 3 (Section 5.3), we identified this strategy and prompting mechanism with this engine as the best-performing one in the whole research. The results for Qwen2-VL are presented in Table 5.9. We compare it with the best-performing OCR identified in Experiment 3, PaddleOCR, and with GPT-4o, which is the best-performing MLLM architecture, while utilizing Constrained Structural Parsing. The results for PaddleOCR and GPT-4o are presented in Tables 5.10 and 5.11, respectively. When interpreting these results, it is important to contextualize the percentage fluctuations within the boundaries of the sample size. Because the synthetic dataset contains exactly 40 images per tier, a single image accounts for a 2.5% change in overall accuracy. This experiment is designed as a qualitative stress test to observe specific failure mechanisms, rather than a broad statistical generalization.

For a better understanding of the results of this experiment, a bar chart can be seen in Figure 5.2.

5.4.1 Clean, Minor Occlusion and Off-Target Scratch (Tiers 0-2)

Images in Tier 0 are the clean and spatially preprocessed workshop images. The pipeline established the highest accuracy at 77.50% using Qwen2-VL, outperforming GPT-4o at 60.00% and PaddleOCR at 52.50%. While these images were spatially preprocessed (background removed and geometrically straightened), they are not pristine, sterile laboratory photos. They are real-world images of turbocharger nameplates captured in an active workshop that retain inherent degradation. The achieved accuracy is very

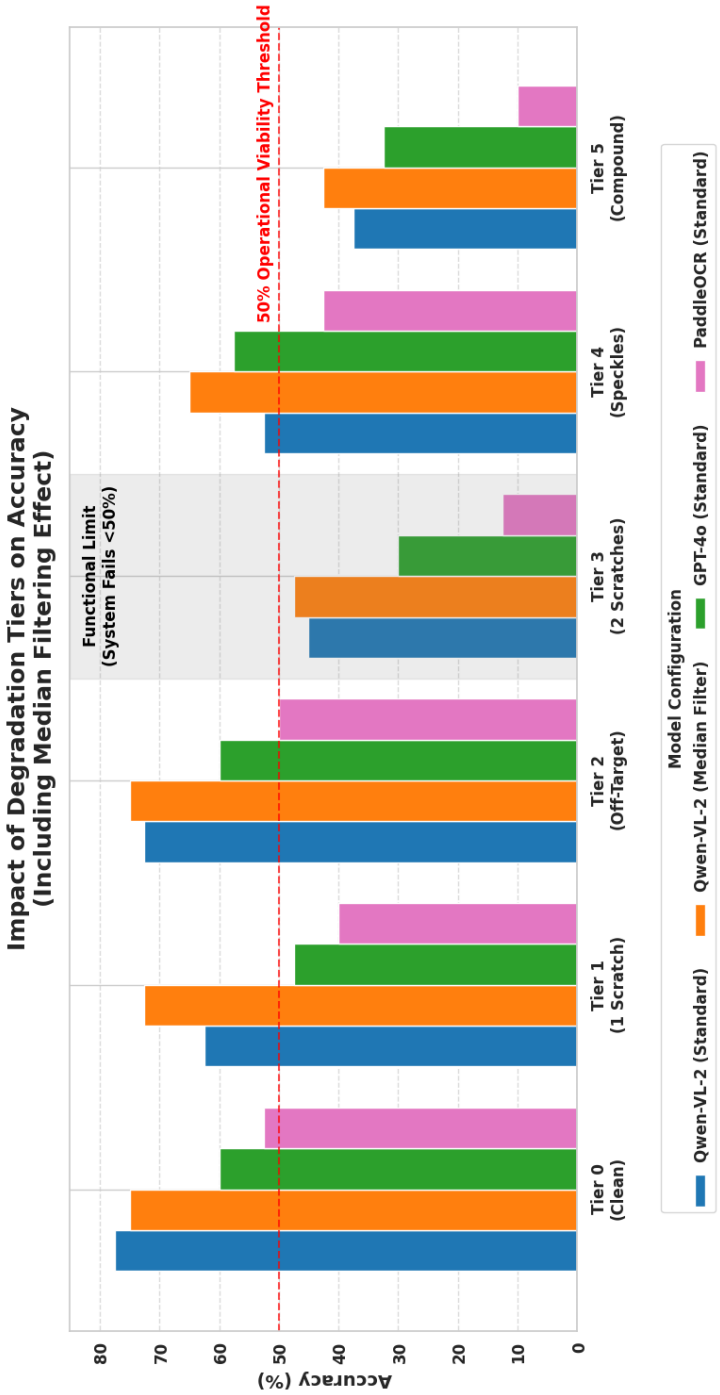


Figure 5.2: Accuracy on Tiers and Engines.

close to the evaluation of the same pipeline, but with the full dataset of 121 images (76.03% for Qwen2-VL, 53.72% for PaddleOCR, and 63.64% for GPT-4o), validating the dataset, used for stress testing (Section 4.1.2) as a representative control set.

Introducing a single scratch directly over the part number (Tier 1) caused an immediate, noticeable drop in performance across all architectures. Qwen2-VL's accuracy fell to 62.50%, GPT-4o's dropped significantly to 47.50%, and PaddleOCR's fell to 40.00%. While a 15 percentage point reduction for Qwen2-VL may initially appear significant, it actually highlights the robustness of the developed pipeline. The scratch partially obscures at least one full digit of an 8 - to 11-symbol sequence. Achieving 62.50% accuracy under these conditions demonstrates that the Qwen2-VL, paired with the downstream fuzzy matching engine, successfully infers and reconstructs the missing character the majority of the time. However, this is not the case for PaddleOCR and GPT-4o.

This experiment proves that extraction failure is highly dependent on the spatial placement of the noise, not just the presence of it. If the artificial scratch had been placed anywhere else on the metallic surface rather than directly over the identification number, the accuracy rebounds. In Tier 2, Qwen2-VL recovered to 72.50%, GPT-4o is stable like it was in Tier 0 with 60%, and PaddleOCR increased to 50.00%.

There is a statistically small accuracy drop between Tier 0 and Tier 2 when we use Qwen2-VL and PaddleOCR. Out of 40 images, Qwen2-VL failed on just 2 additional images (dropping from 31 to 29), and PaddleOCR failed on just 1 additional image (dropping from 21 to 20). In the 2 additional failure cases for Qwen2-VL, the model returned the prompt instead of the raw extracted text from the image. This demonstrates that when visual noise introduces uncertainty, the MLLM can suffer from hallucination loops where the highest-probability output becomes the instruction set (prompt) rather than the visual data. On the other side, PaddleOCR fails because of its architecture, which relies on a probability map to dynamically learn optimal binarization thresholds [14]. Introducing a new, dark scratch changes the overall pixel intensity distribution of the image crop, which can blur out or misclassify a faint dot-peen character that barely passed the threshold in Tier 0. GPT-4o stayed stable because of its self-attention mechanisms and robust multimodal pre-training.

5.4.2 Severe Localized Occlusion (Tier 3)

In Tier 3 (2 scratches, which damage at least 2 digits from the 8 - to 11-symbol sequence), all of the models fail below the operational threshold of 50%. This Tier marks the functional limit of the developed pipeline, as the visual geometry is destroyed beyond what the downstream fuzzy matching engine can reliably reconstruct. Qwen2-VL achieved 45%, GPT-4o went to 30.00%, and PaddleOCR dropped sharply to just 12.50%.

These results can be explained by the fact that when a single digit is missing (Tier 1), the Levenshtein distance to the correct database entry is small enough for safe inference. But when two or more digits are destroyed (Tier 3), the mathematical distance becomes too vast. There are too many similar part numbers in the database, making it impossible to infer the correct sequence without risking a false positive.

5.4.3 Global Speckle Noise vs Compound Degradation (Tiers 4 and 5)

Tier 4 simulates global oil and dirt by applying speckle noise across the entire nameplate. Because dot-peen characters consist of multiple dots, this fundamentally challenges the text extraction engine's ability to separate alphanumeric characters formed by dot-peen dots from speckle noise dots.

Despite this, the results suggest that the models are surprisingly resilient to low-level, widespread background noise. Qwen2-VL achieved an accuracy of 52.50%, which is 10 percentage points lower than its performance under 1 scratch occlusion. PaddleOCR achieved 42.50%, and GPT-4o achieved 57.50%, which is one less match than when using clean images. GPT-4o even performed 10 percentage points better in the presence of speckle noise compared to 1 scratch (Tier 1 vs Tier 4).

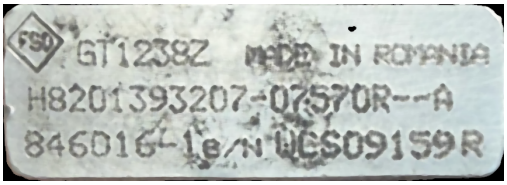
However, under Tier 5's compound degradation, which combines both the global speckles and the severe double-scratch occlusion, the visual decoders were entirely overwhelmed. Accuracy dropped to 37.50% for Qwen2-VL, 32.50% for GPT-4o, and just 10.00% for PaddleOCR. This is expected because, as observed in Tier 3, 2 scratches make it extremely difficult to extract the part number correctly, and even harder if they are combined with speckle noise.

5.4.4 Noise Reduction via Median Filtering

The median filter, also known as median blur, is a non-linear digital filtering technique proven to remove salt-and-pepper noise as already mentioned in the theoretical background in Section 2.7 and literature review in Section 3.2.3. However, applying this technique to dot-peen nameplates carries a significant theoretical risk. Because dot-peen characters are formed by small, disconnected hemispherical indentations, they physically resemble impulse noise, making the text itself highly susceptible to being indiscriminately blurred. To observe how the median filter impacts the accuracy of the pipeline, we apply it to all of the six tiers from Section 4.1.2. The side-by-side comparison of the standard cutout example image vs the image with a median filter applied to it is to be found in Figure 5.3.

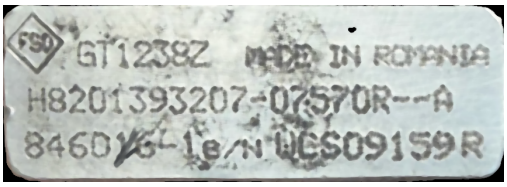
Notably, the corners of the image with the median filter applied are black. A black dot in the upper center of the image is also to be observed. Those regions were empty

(a) Tier 0: Standard.



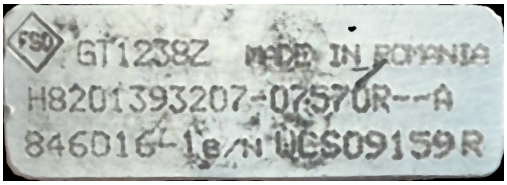
(b) Tier 0: Median Filter.

(c) Tier 1: Standard.



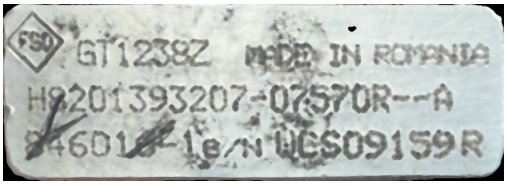
(d) Tier 1: Median Filter.

(e) Tier 2: Standard.



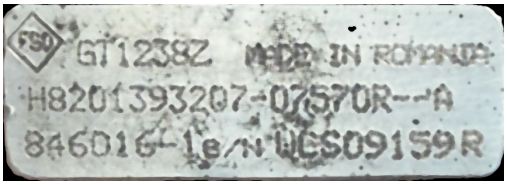
(f) Tier 2: Median Filter.

(g) Tier 3: Standard.



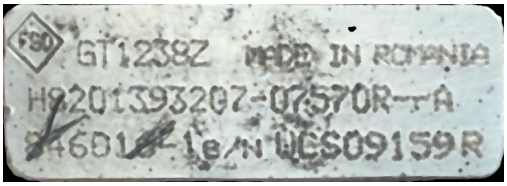
(h) Tier 3: Median Filter.

(i) Tier 4: Standard.



(j) Tier 4: Median Filter.

(k) Tier 5: Standard.



(l) Tier 5: Median Filter.

Figure 5.3: Side-by-side Comparison of Standard Images versus Median Filtering applied across all Degradation Tiers.

Table 5.12: Stress Testing across Synthetic Noise Levels with Qwen2-VL (Standard vs. Median Filter).

Tier *Degradation*	**Tier 0** *Clean*	**Tier 1** *1 Scratch*	**Tier 2** *Off-Target*	**Tier 3** *2 Scratches*	**Tier 4** *Speckles*	**Tier 5** *Compound*
Qwen2-VL						
Total Images	40	40	40	40	40	40
Correct Matches	31	25	29	18	21	15
Incorrect Matches	9	15	11	22	19	25
Accuracy	**77.50%**	**62.50%**	**72.50%**	**45.00%**	**52.50%**	**37.50%**
Median Filter Qwen2-VL						
Total Images	40	40	40	40	40	40
Correct Matches	30	29	30	19	26	17
Incorrect Matches	10	11	10	21	14	23
Accuracy	**75.00%**	**72.50%**	**75.00%**	**47.50%**	**65.00%**	**42.50%**

Table 5.13: Accuracy Comparison on Preprocessed Dataset: Standard versus Median-Filter.

	Qwen2-VL	**Median Filter Qwen2-VL**
Total Images	121	121
Correct Matches	92	89
Incorrect Matches	29	32
Accuracy	**76.03%**	**73.55%**

after the preprocessing stage. They are explained by the architecture of the median filter. When the filter's sliding spatial window traverses sharp external corners or internal voids, the proportion of zero-value background pixels within the kernel frequently exceeds the 50% threshold. The median defaults strictly to zero, forcing the black padding.

The Impact of Median Filtering on Specific Degradation Tiers

The results in Table 5.12 quantitatively exhibit the dual nature of applying the median filter on dot-peen images. As the literature suggests, the median filter is highly effective on global noise (speckles in Tier 4) and increases the accuracy of the pipeline from 52.50% to 65.00% (improvement of 12.5 percentage points). It also provided another substantial improvement of 10.00 percentage points when applied to only one scratch (Tier 1). When applied to two scratches (Tier 3) and one scratch off-target (Tier 2), it resulted in only one more correct prediction, and when applied to the compound degradation of 2 scratches and speckles, there were 2 more correct predictions.

On the other hand, the data reveals a critical theoretical risk: applying the filter to relatively clean images actively destroys viable text. In Tier 0, the median filter mistakenly blurred the actual dot-peen indentations, causing extraction accuracy to drop from 77.50% to 75.00% (one less correct prediction).

Global Dataset Impact and Overall Viability

When we look at Table 5.13, we observe the results of applying the median filter to our whole preprocessed dataset of 121 images. Applying the filter doesn't improve the system but rather drops the peak accuracy of 76.03% (92 correct matches) down to 73.55% (89 correct matches). This again proves that the median filter cannot be deployed as a "one-size-fits-all" default operation within the pipeline.

System Implementation: Conditional Filter Activation

The developed pipeline has to dynamically control when the median filter is applied as a preprocessing step. There are two ways this can be handled.

- **Manual Activation:** After a human visually identifies severe damage on the nameplate, they manually enable the filter.
- **Automated Activation:** In future work, a system that utilizes an image quality assessment module can be added to the pipeline. By analyzing the variations or the pixel intensity histogram of the cropped nameplate, the system can estimate

the ratio of salt-and-pepper noise. If the detected impulse noise exceeds a predetermined threshold, the median filter is applied before continuing to the next steps of the pipeline.

5.4.5 Summary of Experiment 4

Experiment 4 addresses RQ3 by stress-testing the optimized pipeline against increasing levels of artificially introduced degradation. With this experiment, we showed the extent to which our pipeline is effective against dust and scratches, and how this noise can be reduced. The system maintained strong accuracy when faced with minor localized occlusions (Tier 1) and global speckle noise (Tier 4).

Crucially, testing against off-target degradation (Tier 2) revealed a fundamental difference between the different text extraction engines. While the traditional PaddleOCR engine suffered from deterministic binarization shifts that merged distinct dot-peen characters, the generative Qwen2-VL architecture exhibited non-deterministic attention failures, occasionally resulting in a hallucination loop where the model abandoned the visual task and output the prompt. At the same time, the proprietary model GPT-4o kept its accuracy regardless of whether there was a scratch off-target or no scratch at all. The stress testing also definitively established the system's functional threshold at Tier 3. When multiple targeted scratches destroy the geometry of more than one character within the part number sequence, the combined failure of the visual decoder (regardless of whether the underlying engine is Qwen2-VL, PaddleOCR, or GPT-4o) and the downstream fuzzy matching engine causes accuracy to drop below the operational viability threshold of 50%.

Finally, evaluating the median filter as a noise reduction technique demonstrated its dual nature. It proved highly effective at limiting global noise, notably improving Tier 4 (speckles) accuracy from 52.50% to 65.00%. It also showed improvement by 10 percentage points when it's applied to Tier 1 (one scratch). However, applying this filter universally introduces a severe operational risk. It degrades the performance on clean images by filtering the disconnected dots that create the characters of text we want to extract. Furthermore, in Tiers 2, 3, and 5, the filter provided minimal benefit, proving that algorithmic smoothing cannot computationally reconstruct text destroyed by deep scratches. Consequently, the median filter cannot be deployed as a "one-size-fits-all" preprocessing step, but must instead be conditionally activated solely for nameplates exhibiting widespread impulse noise.

5.5 Experiment 5: Evaluation of Pipeline Computational Latency

Table 5.14: GPU Computational Time for each Stage of the Proposed Pipeline.

Pipeline Stage	**GPU Time (s)**
Background Removal	29.64
Perspective Rectification & Orientation Correction	3.73
Text Extraction via Qwen2-VL (Strategy B)	18.50
Postprocessing (Strategy B), Database Grounding & Re-ranking	3.14
Total	**55.01**

Table 5.14 presents the execution times for the optimal pipeline configuration (utilizing Qwen2-VL with Strategy B and its respective postprocessing). The computational time required per single image was recorded using a Google Colab environment equipped with an NVIDIA L4 GPU (24 GB VRAM). To isolate the true operational latency of the pipeline, these measurements strictly evaluate active inference and processing; environmental setup, dependency installation, and initial model loading times are explicitly excluded. A detailed analysis of the latency distribution reveals a clear architectural divide, explaining why certain stages require significantly more time than others:

- **Heavy Neural Network Inference:** The Background Removal and Text Extraction phases account for the vast majority (87.51%) of the execution time (29.64 seconds and 18.50 seconds, respectively). This high latency is a direct result of querying massive, computationally dense foundation models. Background removal relies on the SAM 2 to spatially evaluate the image and dynamically generate precise masks and on PaddleOCR to find the text on the image. Similarly, text extraction requires the MLLM Qwen2-VL to decode complex, degraded visual tokens into text.
- **Lightweight Deterministic Algorithms:** In contrast, the Perspective Rectification and Orientation Correction (3.73 seconds) and Postprocessing, Database Grounding, and Re-ranking (3.14 seconds) stages are highly efficient. These steps rely on fast, deterministic mathematical algorithms—such as RANSAC line fitting, minimal PaddleOCR checks, and Regex-driven fuzzy string matching, which execute rapidly with minimal computational overhead.

Notably, the preprocessing stages (Background Removal combined with Perspective Rectification and Orientation Correction) consume 60.66% (33.37 seconds) of the pipeline's total execution time. However, as demonstrated in Experiment 2 (Section 5.2), this significant computational investment is entirely justified. Spatial preprocessing is essential for the system's viability. By eliminating irrelevant visual context and straightening the text, preprocessing nearly doubles the accuracy of the Qwen2-VL model—jumping from 41.73% on not preprocessed images to a peak accuracy of 76.03% on preprocessed crops, while simultaneously reducing the model's token consumption.

5.6 Discussion of Results

The execution and evaluation of the five experiments demonstrate that automatic and accurate extraction of part numbers from degraded, dot-peen metallic nameplates is achievable, provided the architecture utilizes spatial preprocessing and the correct generative extraction strategy. The achieved results answer the previously stated three research questions and problem statement and show that the designed pipeline is highly reliable, though bounded by specific physical limitations. The findings can be summarized into the following core conclusions:

- **Preprocessing is Essential:** Inputting raw, unedited images directly into the pipeline yields poor results. The preprocessing steps reliably (with 95.28% success rate in Experiment 1) remove the background and straighten the image, which increases the performance of the whole pipeline by 34.3 percentage points (from 41.73% to 76.03%), while simultaneously reducing computational token consumption by eliminating irrelevant visual context as shown in Experiment 2. However, the preprocessing takes 60.66% of the whole pipeline computational time.
- **MLLM Dominance and the Failure of Prompt Constraints** Experiment 3 shows that OCR architectures are largely unequipped to handle degraded dot-peen text. Tesseract failed almost entirely, achieving only 5.79% accuracy, which is also the lowest pipeline accuracy achieved in this thesis. MLLMs significantly outperformed OCR, with the sole exception of InternVL3, which performed slightly below PaddleOCR. Experiment 2 revealed that the performance of the model is strictly dictated by the prompting strategy. The Constrained Structural Parsing approach overloaded the reasoning capabilities of open-weight models, halving Qwen2-VL's accuracy to 30.58% and inflating token costs. Utilizing Unconstrained Raw Transcription, delegating the structural formatting to a downstream Regular Expression (Regex) module, allowed Qwen2-VL to achieve the peak pipeline

accuracy of 76.03%. The accuracy of the proprietary models GPT-4o and Gemini 2.5 Flash stayed around 64% regardless of the prompting strategy.

- **Operational Boundaries and Distinct Failure Modes:** The stress testing of the pipeline in Experiment 4 mapped its limitations. The system proved highly resilient to minor, off-target occlusions and global speckle noise. However, the system's functional threshold was established at Tier 3. When deep, physical scratches destroy the core geometry of multiple characters, the developed pipeline cannot computationally reconstruct the missing data even after database grounding and re-ranking, dropping accuracy below 50%. Furthermore, this stress testing highlighted a fundamental architectural divide in how models fail under noise. PaddleOCR fails due to global binarization shifts, where new dark pixels alter dynamic thresholds and blur character geometry. In contrast, Qwen2-VL suffers from attention disruptions, occasionally abandoning the visual task and outputting the system prompt. Finally, applying a median filter to mitigate noise showed its dual nature. While it effectively reduced heavy speckle noise and one scratch (improving Tier 4 accuracy by 12.5 percentage points and Tier 1 by 10 percentage points), applying it universally actively destroyed viable text on clean images and offered minimal benefit against severe degradation.

The experiment's results indicate that traditional, rigid computer vision is insufficient for the degraded dot-peen turbocharger nameplate domain. The most viable architecture for this industrial challenge is a Qwen2-VL with Unconstrained Raw Transcription and Regex postprocessing on preprocessed images, backed by a deterministic fuzzy-matching database engine to ground and re-rank its generative outputs.

6 Conclusion and Future Work

6.1 Conclusion

As the automotive industry increasingly prioritizes sustainable remanufacturing, reliable automated model recognition has become essential for efficient turbocharger repair support. Yet the harsh realities of Maintenance, Repair, and Overhaul (MRO) environments have historically caused standard optical recognition systems, which typically rely on assumptions of clean, high-contrast, and continuously rendered text, to fail.

The real-life images, used for this research, have a combination of multiple degradation typologies. The nameplates are dirty, oily, rusty, and physically scratched. Because of the lighting and the reflective metallic surfaces of the images, they often suffer from glare. The arbitrary capture angles characteristic of handheld smartphone photography naturally subject the images to multi-axis skew, text rotation, and severe perspective distortion. The frequent use of dot-peen typography—where characters are formed by multiple disconnected dots rather than continuous strokes—fundamentally breaks the assumptions of traditional OCR systems. This thesis developed a systematically evaluated, computationally efficient end-to-end pipeline capable of standardizing geometric distortions and accurately extracting Garrett turbocharger part numbers from severely degraded industrial nameplates. By reliably automating the extraction of this unique identifier, the pipeline successfully bridges the gap between optical data and repair support. Having a digitized, verified part number immediately enables automated turbocharger model recognition, allowing technicians to seamlessly cross-reference the specific replacement components and maintenance protocols needed to repair the turbocharger.

By isolating the extraction process into stages, this research evaluates each stage and gives valuable insights into the topic, such as:

- **Efficiency of Preprocessing:** The first stage of our pipeline is the preprocessing. Its evaluation in Experiment 1 confirmed that the preprocessing can be achieved reliably using minimal human interaction. The background segmentation using SAM 2 candidate masks proved highly effective at locating and isolating the industrial nameplates. The system successfully generated high-fidelity masks for

85.83% of the total dataset completely autonomously, achieving an impressive average IoU of 0.83 against manual ground-truth polygons. To address the segmentation failures, a manual point-prompt fallback utilizing SAM 2 was employed. By requiring a human operator to select a point within the target nameplate, the missing masks were generated, boosting the final segmentation success rate to 100%. However, during the geometric rectification, 6 images failed to be preprocessed entirely, which dropped the success rate of the whole preprocessing pipeline to 95.28%. It is also important to state that the spatial standardization was not flawless, as a subset retained residual tilt, and 12 images were finalized with a 180-degree upside-down rotation.

- **Necessity of Preprocessing:** The results from Experiment 2 in Section 5.2 establish the necessity of preprocessing as an absolute must in this research domain. By implementing a spatial preprocessing pipeline that includes precise background elimination via SAM 2 and geometric rectification using RANSAC, and orientation correction using PaddleOCR, the system effectively eliminated the distracting background noise typical for MRO environments, and straightening the image made the text more readable. This preprocessing pipeline increased the achieved accuracy of the whole pipeline from 41.73% to 76.03% on the preprocessed images using the same prompting strategy and text extraction engine. This is a significant increase of 34.3 percentage points. Furthermore, the preprocessing demonstrated significant computational benefits. By removing irrelevant background context, the preprocessing drastically reduced the cognitive load on the language model, cutting total token consumption from 69,515 to 61,031. This dual improvement confirms that isolating the specific region of interest is essential for maximizing both the semantic accuracy and the computational efficiency of MLLMs.

- **Optimal Prompting Strategy:** A critical finding of this research is the pronounced performance disparity between different MLLM prompting strategies as shown in Experiment 2. The experiment demonstrated that Unconstrained Raw Transcription (Strategy B), where the model functions purely as an advanced optical transcriber while delegating structural formatting to a downstream Regex module, outperforms Constrained Structural Parsing (Strategy A). The model's reasoning capabilities were overloaded by all of the instructions and clues given in the longer prompt in Strategy A. This constrained approach not only inflated token consumption but severely crippled extraction accuracy, yielding an unsatisfactory 30.58% while using Qwen2-VL. In contrast, using Regex postprocessing resulted in way more deterministic behaviour and allowed the Unconstrained Raw Transcription strategy to achieve a 76.03% accuracy rate. This indicates that while MLLMs possess unprecedented visual decoding capabilities, imposing rigid, programmatic

formatting constraints directly within the prompt fundamentally degrades their primary extraction performance.

- **MLLMs over Traditional OCR:** The benchmarking in Experiment 3 in Section 5.3 demonstrated that traditional OCR engines are fundamentally unable to handle degraded dot-peen typography. Tesseract achieves an unusable 5.79% accuracy, EasyOCR reaches 23.97%, and PaddleOCR is the only model that exceeds the 50% usability threshold with 53.72%. These legacy systems rely heavily on continuous edge detection and deterministic binarization, which are easily disrupted by disconnected dots, specular glare, and minor surface noise, which are norms in the turbocharger domain. In contrast, MLLMs successfully leverage zero-shot semantic reasoning. By fusing advanced visual encoders with large language models, MLLMs can infer dirty and dot-peen characters that Tesseract and EasyOCR cannot process. When deployed with the optimal unconstrained prompting strategy, the open-weight Qwen2-VL model vastly outperformed the best-performing OCR, achieving 76.03% pipeline accuracy. This verifies that for harsh industrial digitization tasks, MLLMs with the right prompting strategy are significantly more reliable than OCR.
- **Stress Testing and Operational Boundaries:** In Experiment 4 (Section 5.4), systematic stress testing with ascending levels of synthetic degradation mapped the exact operational boundaries of the different text extraction engines. The generative Qwen2-VL model demonstrated the most robust performance against minor, localized occlusions (Tier 1), maintaining a 62.50% accuracy rate, whereas GPT-4o dropped significantly to 47.50% and the traditional PaddleOCR fell to 40.00%. When faced with global speckle noise simulating heavy oil and dirt (Tier 4), GPT-4o proved surprisingly resilient (57.50%), followed by Qwen2-VL (52.50%) and PaddleOCR (42.50%). Furthermore, testing against off-target degradation (Tier 2) highlighted a stark architectural contrast. PaddleOCR suffered from deterministic binarization shifts, Qwen2-VL occasionally entered non-deterministic hallucination loops, while GPT-4o maintained consistent stability regardless of off-target scratches. However, the testing established a firm functional threshold at Tier 3 degradation for all engines. When multiple targeted scratches destroy the geometry of more than one character within the core part number sequence, the downstream fuzzy matching engine's reconstruction limits are exceeded. This caused accuracy to plummet below the 50% operational viability threshold across the board (Qwen2-VL down to 45.00%, GPT-4o to 30.00%, and PaddleOCR to just 12.50%). Under severe compound degradation combining global noise with two scratches (Tier 5), the visual decoders were entirely overwhelmed, with PaddleOCR's accuracy collapsing to 10.00%.

Finally, the evaluation of the median filter as a targeted noise reduction technique revealed its dual nature. It proved highly effective at limiting superficial global noise, notably improving Qwen2-VL's Tier 4 accuracy by 12.5 percentage points (from 52.50% to 65.00%). However, when applied to clean images, the filter indiscriminately blurred the vital, disconnected dot-peen indentations. This negative side-effect reduced the overall pipeline accuracy from 76.03% to 73.55%, definitively confirming that applying the median filter as a universal preprocessing step introduces a severe operational risk and must instead be conditionally activated.

- **Computational Latency Trade-offs:** While the developed pipeline achieves a high peak accuracy, Experiment 5 revealed that this performance currently comes at the cost of high computational latency. The optimal pipeline configuration requires approximately 55 seconds to process a single image on a standard GPU. The vast majority of this execution time is consumed by the heavy NN foundation models, specifically 29.64 seconds for SAM 2 during background removal and 18.50 seconds for Qwen2-VL during text extraction. This demonstrates a clear architectural trade-off: achieving robust, zero-shot semantic reasoning in degraded industrial environments currently prohibits instantaneous, real-time extraction.

6.2 Future Work

6.2.1 Refinement of Geometric Rectification and Orientation Correction

While the spatial preprocessing pipeline significantly improved overall extraction accuracy, not all images are perfectly straightened and rotated, which cascaded into downstream text recognition errors. Future research must address the persistent issues of residual tilt and critical 180-degree orientation failures. The current pipeline's orientation heuristic, which determines the upright position by running a two-pass PaddleOCR extraction and counting the recognized alphanumeric characters, finalized 12 images upside down. Implementing a specialized, lightweight CNN trained specifically to classify in which direction the dot-peen text on the nameplate is rotated could resolve this failure. Further research into whether a specialized CNN or other alternatives can perform geometric rectification is required.

6.2.2 Multi-Attribute Relational Verification

Currently, the postprocessing and fuzzy-matching engine relies almost exclusively on isolating and verifying the primary Garrett part number. However, the nameplates contain additional data that can be extracted, specifically the turbo model and the

original equipment manufacturer code. Future development should incorporate them. For these purposes, an extensive manufacturer database with this data is needed. A database with all of the specific formats of these numbers is also needed for the error correction steps before fuzzy-matching. In scenarios involving severe, localized degradation, such as the Tier 3 and Tier 5 stress tests, where scratches make a significant part of the part number unrecognizable, the system could logically deduce the correct part number by cross-referencing the other numbers. Transitioning to this holistic data-structuring approach would significantly elevate the pipeline's resilience against extreme, localized physical occlusions.

6.2.3 Generalizing Asset Recognition for Universal Applicability

The system has to be tested to see if it recognizes the part numbers of other turbocharger manufacturers like Borgwarner and Holset. If it doesn't, the pipeline should be extended to extract the manufacturer and read its part number reliably. This evolves the architecture from a specialized, single-manufacturer tool into a universally applicable industrial asset recognition system. For these purposes, a dataset with part number and the specific manufacturer is needed.

6.2.4 Implementation of Conditional Image Filtering

Experiment 4 in Section 5.4 proved that applying a median filter universally reduces the developed pipeline accuracy by blurring parts of the dot-peen characters. Therefore, future work should develop a system that automatically uses a median filter if the image is too dirty. For these purposes, a preliminary image quality assessment step, which analyzes the high-frequency variations or pixel intensity histogram, should be utilized.

6.2.5 Domain-Specific Fine-Tuning of MLLMs

Because architectures like Qwen2-VL and InternVL3 are open-weight, they can be fine-tuned. Proprietary cloud models (like GPT-4o and Gemini) do not afford this opportunity. Research is needed to fine-tune the open weight models directly on a specialized, high-volume dataset of dirty dot-peen engraved industrial nameplates. By training the visual encoders and language heads on the characteristics of dot-peen text and the specific formatting of turbocharger part numbers, the part number extraction accuracy could be drastically improved. The fine-tuning could make the whole pipeline more resistant to degradation.

6.2.6 Isolated Stress Testing for Specular Glare

Experiment 4 evaluated the effectiveness of the pipeline under degradations, such as scratches and global speckle noise. However, the exact operational limits regarding extreme glare remain under-evaluated. In this study, the impact of specular glare was difficult to isolate because the primary real-world dataset featured compounding variables: heavily oxidized, dirty, or oily plates naturally diffuse light, masking the true failure thresholds of severe glare on clean metallic surfaces. The glare can be generated artificially, just like it was done with scratches in Experiment 4. Evaluating the exact operational limits of the pipeline under glare would show if there is a need for illumination-invariant preprocessing techniques, such as Retinex-based image enhancement.

6.2.7 Generative Restoration of Occluded Text

In Experiment 4, the developed pipeline reaches its operational limit when deep physical scratches destroy the core geometry of multiple characters (such as in Tiers 3 and 5). Current MLLM architectures and standard generative visual restoration techniques attempt to infer missing characters, but they often fail or hallucinate, as already established in Section 3.3. Future research should explore the integration of specialized generative MLLMs or domain-conditioned diffusion models designed explicitly to reconstruct occluded dot-peen text. By conditioning these generative networks on both the strict alphanumeric syntax of manufacturer part numbers and the precise spatial morphology of dot-peen dots, the system could theoretically reconstruct the missing visual data beneath a scratch.

6.2.8 Latency Optimization for Real-Time Deployment

To transition the developed pipeline into a real-life turbocharger repair workshop, the computational overhead and end-to-end inference speed must be considered. As quantified in Experiment 5, the current optimal architecture requires nearly 55 seconds to process a single image. This significant latency is primarily driven by computationally heavy foundation models; specifically, SAM 2 for background masking consumes 29.64 seconds alone, while Qwen2-VL for visual decoding requires an additional 18.50 seconds. Future work should investigate strategies to accelerate the pipeline and reduce this bottleneck. This could include implementing model quantization techniques, utilizing accelerated inference engines, or replacing these heavy architectures with lightweight, domain-distilled alternatives that do not compromise the system's extraction accuracy.

Abbreviations

MRO Maintenance, Repair, and Overhaul

OCR Optical Character Recognition

MLLM Multimodal Large Language Model

DPM Direct Part Marking

CNN Convolutional Neural Network

CRAFT Character Region Awareness for Text Detection

CRNN Convolutional Recurrent Neural Network

CTC Connectionist Temporal Classification

SVTR Single Visual Model for Text Recognition

LLMs Large Language Models

SOTA State-of-the-Art

CER Character Error Rate

WER Word Error Rate

OLS Ordinary Least Squares

ALPR Automatic License Plate Recognition

SAM 2 Segment Anything Model 2

SAM 3 Segment Anything Model 3

SMF Standard Median Filter

GAN Generative Adversarial Network

NLP Natural Language Processing

IoU Intersection over Union

RANSAC Random Sample Consensus

Regex Regular Expression

RQ Research Question

KNN K-Nearest Neighbors

DBNet Differentiable Binarization

LSTM Long Short-Term Memory

NN Neural Network

List of Figures

List of Tables

Bibliography

[1] J. Oláh, N. Aburumman, J. Popp, M. A. Khan, H. Haddad, and N. Kitukutha. "Impact of Industry 4.0 on Environmental Sustainability." In: *Sustainability* 12.11 (2020), p. 4674. DOI: 10.3390/su12114674.

[2] M. A. Berawi. "The Role of Industry 4.0 in Achieving Sustainable Development Goals." In: *International Journal of Technology* 10.4 (July 2019), pp. 644–647. ISSN: 2087-2100. DOI: https://doi.org/10.14716/ijtech.v10i4.3341.

[3] A. Frank, G. Vida, and P. Varga. "Reliable Identification Schemes for Asset and Production Tracking in Industry 4.0." In: *Sensors* 20.13 (2020). ISSN: 1424-8220. DOI: 10.3390/s20133709.

[4] R. Salles, J. Leiria, and J. Mendes. "Data Entry Errors Detection for¬†Industrial Quality Control Variables Using Data-Driven Models." In: *CONTROLO 2024*. Ed. by A. P. Aguiar, P. Rocha Malonek, V. H. Pinto, F. A. C. C. Fontes, and R. Chertovskih. Cham: Springer Nature Switzerland, 2025, pp. 41–53. ISBN: 978-3-031-81724-3.

[5] G. Gupta, Swati, M. Sharma, and L. Vig. "Information Extraction from Hand-Marked Industrial Inspection Sheets." In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 06. 2017, pp. 33–38. DOI: 10.1109/ICDAR.2017.346.

[6] Z. Raisi, M. A. Naiel, P. Fieguth, S. Wardell, and J. Zelek. *Text Detection and Recognition in the Wild: A Review*. 2020. arXiv: 2006.04305 [cs.CV].

[7] M. Shahin, F. Chen, and A. Hosseinzadeh. "Machine-based identification system via optical character recognition." In: *Flexible Services and Manufacturing Journal* 36 (Apr. 2023), pp. 1–28. DOI: 10.1007/s10696-023-09497-8.

[8] D. Dragičević, S. Tegeltija, G. Ostojić, S. Stankovski, and M. Lazarević. "Reliability of Dot Peen Marking in Product Traceability." In: *International Journal of Industrial Engineering and Management (IJIEM)* 8.2 (2017), pp. 71–76. ISSN: 2217-2661.

[9] E. Ahearne. "Engineering the surface for direct part marking (DPM)." In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 1–10. ISSN: 1755-5817. DOI: https://doi.org/10.1016/j.cirpj.2020.01.003.

[10] OpenAI. *GPT-4o System Card*. 2024. arXiv: 2410.21276 [cs.CL].

[11] Gemini Team, Google. *Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities*. 2025. arXiv: 2507.06261 [cs.CL].

[12] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, Z. Gao, E. Cui, X. Wang, Y. Cao, Y. Liu, X. Wei, H. Zhang, H. Wang, W. Xu, H. Li, J. Wang, N. Deng, S. Li, Y. He, T. Jiang, J. Luo, Y. Wang, C. He, B. Shi, X. Zhang, W. Shao, J. He, Y. Xiong, W. Qu, P. Sun, P. Jiao, H. Lv, L. Wu, K. Zhang, H. Deng, J. Ge, K. Chen, L. Wang, M. Dou, L. Lu, X. Zhu, T. Lu, D. Lin, Y. Qiao, J. Dai, and W. Wang. *InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models*. 2025. arXiv: 2504.10479 [cs.CV].

[13] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. *Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution*. 2024. arXiv: 2409.12191 [cs.CV].

[14] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Z. Zhang, C. Zhou, H. Liu, Y. Zhang, W. Lv, K. Huang, Y. Zhang, J. Zhang, J. Zhang, Y. Liu, D. Yu, and Y. Ma. *PaddleOCR 3.0 Technical Report*. 2025. arXiv: 2507.05595 [cs.CV].

[15] R. Smith. "An Overview of the Tesseract OCR Engine." In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Vol. 2. 2007, pp. 629–633. DOI: 10.1109/ICDAR.2007.4376991.

[16] R. Kittinaradorn et al. *EasyOCR*. https://github.com/JaidedAI/EasyOCR. Accessed: 2026-04-19. 2024.

[17] I. Peneva. *AI-Based Turbocharger Identification*. https://github.com/IvanaPeneva/AI-Based-Turbocharger-Identification. Accessed: 2026-04-22. 2026.

[18] I. Peneva. *MScThesis*. https://syncandshare.lrz.de/getlink/fiMTdJ3LG4nLRN3b9ZT5Y2. Accessed: 2026-05-04. 2026.

[19] S. Mori, C. Suen, and K. Yamamoto. "Historical review of OCR research and development." In: *Proceedings of the IEEE* 80.7 (1992), pp. 1029–1058. DOI: 10.1109/5.156468.

[20] L. Eikvil. *Optical Character Recognition*. Tech. rep. P.B. 114 Blindern, N-0314 Oslo: Norsk Regnesentral, Dec. 1993.

[21] S. Dome and A. P. Sathe. "Optical Charater Recognition using Tesseract and Classification." In: *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*. 2021, pp. 153–158. DOI: 10.1109/ESCI50559.2021.9397008.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG].

[23] D. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun. "Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm." In: *2022 6th International Conference on Electronics, Communication and Aerospace Technology*. 2022, pp. 966–971. DOI: 10.1109/ICECA55336.2022.10009215.

[24] C. X. Liang, P. Tian, C. H. Yin, Y. Yua, W. An-Hou, L. Ming, X. Song, T. Wang, Z. Bi, and M. Liu. *A Comprehensive Survey and Guide to Multimodal Large Language Models in Vision-Language Tasks*. 2025. arXiv: 2411.06284 [cs.AI].

[25] K. G. Derpanis. *Overview of the RANSAC Algorithm*. http://www.cs.yorku.ca/~kosta/. Version 1.2. May 2010.

[26] B. Zdaniuk. "Ordinary Least-Squares (OLS) Model." In: *Encyclopedia of Quality of Life and Well-Being Research*. Ed. by F. Maggino. Cham: Springer International Publishing, 2023. ISBN: 978-3-031-17299-1. DOI: 10.1007/978-3-031-17299-1_2008.

[27] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. *SAM 2: Segment Anything in Images and Videos*. 2024. arXiv: 2408.00714 [cs.CV].

[28] J. Goyvaerts. *Regular Expressions: The Complete Tutorial*. Jan Goyvaerts, July 2007.

[29] Y. Franko, N. Porplytsya, M. Ozhha, O. Potapchuk, and Y. Franko. "Method and Software for Solving the Problem of Fuzzy Matching of Records in Relative Databases." In: *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)*. 2021, pp. 696–699. DOI: 10.1109/ACIT52158.2021.9548560.

[30] OpenCV Team. *Smoothing Images - OpenCV 3.4.20 Documentation*. Accessed: 2026-04-15. 2025.

[31] J. Al-Azzeh, B. Zahran, and Z. Alqadi. "Salt and Pepper Noise: Effects and Removal." In: *International Journal on Informatics Visualization* 2.4 (2018), pp. 252–256.

[32] A. Singh, K. Bacchuwar, and A. Bhasin. "A Survey of OCR Applications." In: *International Journal of Machine Learning and Computing* 2.3 (June 2012), pp. 314–318.

[33] H. Edvartsen. "OCR of dot peen markings with deep learning and image analysis." Master's Thesis. Luleå University of Technology, 2018.

[34] K. Rao, R. N. Awale, A. Diwan, S. M. Rathod, and A. Sharma. "Character Recognition on Forged Entities using AI." In: *2022 International Conference on Industry 4.0 Technology (I4Tech)*. 2022, pp. 1–5. DOI: 10.1109/I4Tech55392.2022.9952894.

[35] S. Lee, D. Kim, and J. Ko. "A Deep Learning Approach to Nameplate Detection and Text Recognition using YOLOv5 and Tesseract OCR." In: *Transactions of the Korean Nuclear Society Spring Meeting*. Jeju, Korea, May 2023.

[36] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. arXiv: 2004.10934 [cs.CV].

[37] F. Gao, S. Li, H. You, S. Lu, and G. Xiao. "Text Spotting for Curved Metal Surface: Clustering, Fitting, and Rectifying." In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–12. DOI: 10.1109/TIM.2020.3012219.

[38] A. A. Vilasan, S. Jäger, and N. Klarmann. *AI-Driven Multi-Stage Computer Vision System for Defect Detection in Laser-Engraved Industrial Nameplates*. 2025. arXiv: 2503.03395 [cs.CV].

[39] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 7464–7475.

[40] H. Wang, R. Fu, X. Zhang, and J. Zhou. "RVAE-LAMOL: Residual Variational Autoencoder to Enhance Lifelong Language Learning." In: *2022 International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–9. DOI: 10.1109/IJCNN55064.2022.9892670.

[41] L. Yang, X. Huang, Y. Ren, et al. "Study on steel plate scratch detection based on improved MSR and phase consistency." In: *Signal, Image and Video Processing* 17.1 (Feb. 2023), pp. 119–127. DOI: 10.1007/s11760-022-02211-5.

[42] I. Konovalenko, P. Maruschak, V. Brevus, and O. Prentkovskis. "Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network." In: *Metals* 11.4 (2021). ISSN: 2075-4701. DOI: 10.3390/met11040549.

[43] Z. Xiang, Y. Zhaolin, M. Qian, J. Zhang, and X. hu. "Metal stamping character recognition algorithm based on multi-directional illumination image fusion enhancement technology." In: *EURASIP Journal on Image and Video Processing* 2018 (Aug. 2018). DOI: 10.1186/s13640-018-0321-7.

[44] M. Mirmehdi, P. Clark, and J. Lam. "A non-contact method of capturing low-resolution text for OCR." In: *Pattern Analysis & Applications* 6.1 (2003), pp. 12–21. ISSN: 1433-7541. DOI: 10.1007/s10044-002-0172-8.

[45] H. Ding, Q. Wang, J. Gao, and Q. Li. *A Training-Free Framework for Video License Plate Tracking and Recognition with Only One-Shot*. 2024. arXiv: 2408.05729 [cs.CV].

[46] V. Gnanaprakash et al. "Automatic number plate recognition using deep learning." In: *IOP Conference Series: Materials Science and Engineering* 1084.1 (2021). DOI: 10.1088/1757-899X/1084/1/012027.

[47] O. Sarkar, S. Sinha, A. K. Jena, A. K. Parida, N. Parida, and R. K. Parida. "Automatic Number Plate Character Recognition using Paddle-OCR." In: *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*. 2024, pp. 1–7. DOI: 10.1109/ICICET59348.2024.10616305.

[48] P. P. Reddy, P. S. Shruthi, P. Himanshu, and T. Singh. "License Plate Detection using YOLO v8 and Performance Evaluation of EasyOCR, PaddleOCR and Tesseract." In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024, pp. 1–6. DOI: 10.1109/ICCCNT61001.2024.10725878.

[49] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera. "Automated License Plate Recognition: A Survey on Methods and Techniques." In: *IEEE Access* 9 (2021), pp. 11203–11225. DOI: 10.1109/ACCESS.2020.3047929.

[50] M. Sohan, T. Sai Ram, and C. V. Rami Reddy. "A Review on YOLOv8 and Its Advancements." In: *Data Intelligence and Cognitive Informatics*. Ed. by I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski. Singapore: Springer Nature Singapore, 2024, pp. 529–545. ISBN: 978-981-99-7962-2.

[51] A. V. Patil and M. M. Dhanvijay. "Engraved character recognition using computer vision to recognize engine and chassis numbers: Computer vision technique to identify engraved numbers." In: *2015 International Conference on Information Processing (ICIP)*. 2015, pp. 151–154. DOI: 10.1109/INFOP.2015.7489368.

[52] G. Prakash, A. K, M. M. Maroof, S. Maria Fernandes, Roopashree, A. Bekal, and R. G. Shetty. "Engraved Character Recognition Using Computer Vision To Recognize Engine Number With Environmental Condition Setup." In: *2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. 2023, pp. 1–7. DOI: 10.1109/DISCOVER58830.2023.10316712.

[53] W. YU, Z. WU, H. LIU, Q. CHEN, X. DUAN, J. MO, J. TONG, F. LIAO, and Q. LIN. "An Engraving Character Recognition System Based on Machine Vision." In: *DEStech Transactions on Computer Science and Engineering* (Oct. 2017). DOI: 10.12783/dtcse/aiea2017/14919.

[54] J. Kronenberger, D. Malysiak, and U. Handmann. "Text and character recognition on metal-sheets." In: *2017 IEEE International Conference on Information and Automation (ICIA)*. 2017, pp. 392–397. DOI: 10.1109/ICInfA.2017.8078940.

[55] S. Elhabian, K. El-Sayed, and S. Ahmed. "Moving Object Detection in Spatial Domain Using Background Removal Techniques-State-of-Art." In: *Recent Patents on Computer Science* 1 (Jan. 2008), pp. 32–54. DOI: 10.2174/1874479610801010032.

[56] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. *SAM 3: Segment Anything with Concepts*. 2026. arXiv: 2511.16719 [cs.CV].

[57] R. Sapkota, K. I. Roumeliotis, and M. Karkee. "The SAM2-to-SAM3 Gap in the Segment Anything Model Family: Why Prompt-Based Expertise Fails in Concept-Driven Image Segmentation." In: *arXiv preprint arXiv:2512.06032* (2025). arXiv: 2512.06032.

[58] Z. Tang, R. Nagabhirava, and C. Liu. *CAD-Prompted SAM3: Geometry-Conditioned Instance Segmentation for Industrial Objects*. 2026. arXiv: 2602.20551 [cs.CV].

[59] R. Chan, C.-W. Ho, and M. Nikolova. "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization." In: *IEEE Transactions on Image Processing* 14.10 (2005), pp. 1479–1485. DOI: 10.1109/TIP.2005.852196.

[60] H. Wang, S. Li, S. Cao, R. Yang, J. Zeng, Z. Qian, and X. Zhang. "On Physically Occluded Fake Identity Document Detection." In: *Proceedings of the 31st ACM International Conference on Multimedia*. MM '23. Ottawa ON, Canada: Association for Computing Machinery, 2023, pp. 1556–1564. ISBN: 9798400701085. DOI: 10.1145/3581783.3612075.

[61] R. Cohen, K. Kedem, I. Dinstein, and J. El-Sana. "Occluded Character Restoration Using Active Contour with Shape Priors." In: *2012 International Conference on Frontiers in Handwriting Recognition*. IEEE. 2012, pp. 495–500.

[62] L. Chang, J. Sun, M. Suwa, H. Takebe, Y. He, and S. Naoi. "Occluded text restoration and recognition." In: DAS '10. Boston, Massachusetts, USA: Association for Computing Machinery, 2010, pp. 151–158. ISBN: 9781605587738. DOI: 10.1145/1815330.1815350.

[63] K. Kahatapitiya, D. Tissera, and R. Rodrigo. "Context-Aware Automatic Occlusion Removal." In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 1895–1899. DOI: 10.1109/ICIP.2019.8803141.

[64] S. Y. Noh and J. Y. Chang. "Stable Diffusion-Based Approach for Human De-Occlusion." In: *Proceedings of the 33rd ACM International Conference on Multimedia*. MM '25. Dublin, Ireland: Association for Computing Machinery, 2025, pp. 10044–10053. ISBN: 9798400720352. DOI: 10.1145/3746027.3755346.

[65] A. Mittal, P. Shivakumara, U. Pal, T. Lu, and M. Blumenstein. "A new method for detection and prediction of occluded text in natural scene images." In: *Signal Processing: Image Communication* 100 (2022), p. 116512. ISSN: 0923-5965. DOI: https://doi.org/10.1016/j.image.2021.116512.

[66] A. Das, S. Palaiahnakote, A. Banerjee, A. Antonacopoulos, and U. Pal. "Soft set-based MSER end-to-end system for occluded scene text detection, recognition and prediction." In: *Knowledge-Based Systems* 305 (2024), p. 112593. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2024.112593.

[67] A. Mittal, P. Shivakumara, U. Pal, T. Lu, M. Blumenstein, and D. Lopresti. "A New Context-Based Method for Restoring Occluded Text in Natural Scene Images." In: *Document Analysis Systems*. Ed. by X. Bai, D. Karatzas, and D. Lopresti. Cham: Springer International Publishing, 2020, pp. 466–480. ISBN: 978-3-030-57058-3.

[68] R. Kajale, S. Das, and P. Medhekar. "Supervised machine learning in intelligent character recognition of handwritten and printed nameplate." In: *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*. 2017, pp. 1–5. DOI: 10.1109/ICAC3.2017.8318748.

[69] Q. Yang, H. Ma, H. Zhao, and J. Liu. "A Review of Image Recognition in Power Nameplate Images." In: *2022 Power System and Green Energy Conference (PSGEC)*. 2022, pp. 1057–1061. DOI: 10.1109/PSGEC54663.2022.9881014.

[70] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao, Z. Wang, Z. Chen, H. Zhang, G. Yang, H. Wang, Q. Wei, J. Yin, W. Li, E. Cui, G. Chen, Z. Ding, C. Tian, Z. Wu, J. Xie, Z. Li, B. Yang, Y. Duan, X. Wang, Z. Hou, H. Hao, T. Zhang, S. Li, X. Zhao, H. Duan, N. Deng, B. Fu, Y. He, Y. Wang, C. He, B. Shi, J. He, Y. Xiong, H. Lv, L. Wu, W. Shao, K. Zhang, H. Deng, B. Qi, J. Ge, Q. Guo, W. Zhang, S. Zhang, M. Cao, J. Lin, K. Tang, J. Gao, H. Huang, Y. Gu,

C. Lyu, H. Tang, R. Wang, H. Lv, W. Ouyang, L. Wang, M. Dou, X. Zhu, T. Lu, D. Lin, J. Dai, W. Su, B. Zhou, K. Chen, Y. Qiao, W. Wang, and G. Luo. *InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency*. 2025. arXiv: 2508.18265 [cs.CV].

[71] A. Kumar. *Is Sanskrit the most token-efficient language? A quantitative study using GPT, Gemini, and SentencePiece*. 2026. arXiv: 2601.06142 [cs.CL].

[72] M. Aubakirova, A. Atallah, C. Clark, J. Summerville, and A. Midha. *State of AI: An Empirical 100 Trillion Token Study with OpenRouter*. 2026. arXiv: 2601.10088 [cs.AI].

[73] J. Stryker, A. Baran, R. Deshmukh, S. Grewal, M. Imran, T. Roluga, S. Ullah, and M. Virk. "Comparative analysis of LLMs, GPT-4 vs Gemini." In: *TechRxiv* 2025.0418 (2025). DOI: 10.36227/techrxiv.174494992.25671081/v1. eprint: https://www.techrxiv.org/doi/pdf/10.36227/techrxiv.174494992.25671081/v1.

[74] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, and C. Gohlke. "Array Programming with NumPy." In: *Nature 585, 357 (2020)* (2020). DOI: 10.1038/s41586-020-2649-2.

[75] Mushroomcat9998. *PPOCRLabel Documentation*. https://github.com/Mushroomcat9998/PaddleOCR/blob/main/PPOCRLabel/README.md. Accessed: 2026-04-19. 2024.

[76] Hugging Face. *Qwen2-VL — transformers documentation*. Hugging Face. 2024. URL: https://huggingface.co/docs/transformers/model_doc/qwen2_vl (visited on 04/27/2026).

[77] Hugging Face. *InternVL — transformers documentation*. Hugging Face. 2024. URL: https://huggingface.co/docs/transformers/model_doc/internvl (visited on 04/27/2026).

[78] M. Bachmann. *rapidfuzz/RapidFuzz: Release 3.13.0*. Version v3.13.0. Apr. 2025. DOI: 10.5281/zenodo.15133267.

[79] S. Gillies, C. van der Wel, J. Van den Bossche, M. W. Taves, J. Arnott, B. C. Ward, et al. *Shapely*. DOI: 10.5281/zenodo.7428463.