UAB
Universitat Autònoma
de Barcelona

Bachelor's Degree in Artificial Intelligence

Real-Time Quality Inspection for Laser Marking Systems: Enhancing Accuracy with Synthetic Data

Student:
Júlia Garcia Torné, 1630382

Professor:
David Castells

March 2025

Contents

1 Introduction

In modern manufacturing environments, automation and data-driven technologies play a key role in optimizing production processes and ensuring quality control. One such technology, laser marking, is widely used for engraving essential information—such as serial numbers, manufacturing dates, and identification codes—directly onto products. These permanent markings are vital for traceability, compliance, and supply chain transparency, particularly in industries like pharmaceuticals, electronics, and automotive manufacturing.

While the physical process of laser marking is well established, the development of intelligent systems capable of verifying these markings in real time remains a significant challenge, especially from a data perspective. As production lines become faster and more complex, the need for automated inspection systems increases. However, building robust machine learning models for this task depends heavily on high-quality data. Acquiring large, diverse, and well-labeled datasets of laser-marked products is difficult, particularly when it comes to defective samples. Real-world data often lacks variety and is heavily imbalanced, as errors are rare and inconsistently documented.

2 Problem Definition

The central problem lies not just in detecting marking errors, but in the limitations of the data used to train such systems. In most industrial contexts, collecting and annotating defective examples is time-consuming and can disrupt production. As a result, datasets tend to be small, biased toward correct samples, and lacking the diversity needed for generalization. Variations in lighting, surface textures, materials, and marking styles introduce further complexity, making it difficult for models to learn robust patterns using real data alone.

To address this, the project adopts a data-centric approach to the inspection of laser markings. Instead of relying solely on real-world examples, the system incorporates synthetic data generation to simulate a wide range of marking scenarios, including different types of errors and challenging conditions. This allows for the creation of large, balanced, and customizable datasets tailored to the needs of model training. The goal is to develop an efficient, lightweight inspection pipeline that can operate in real time, even on resource-constrained edge devices. By enhancing the quality and diversity of training data, the system aims to achieve high accuracy in detecting marking defects—such as incorrect placement, partial codes, or unreadable characters—while minimizing computational cost.

3 Background and State of the Art

Industrial Laser Marking and Quality Control Laser marking has become a cornerstone of modern manufacturing, enabling permanent, high-precision engravings for traceability and compliance in industries ranging from aerospace to pharmaceuticals [2]. Traditional inspection methods rely on manual visual checks or rule-based machine vision systems, which are prone to human error and lack adaptability to diverse product geometries [6]. The shift toward automated, data-driven inspection has been accelerated by advances in deep learning, particularly convolutional neural networks (CNNs) for defect detection [4].

Challenges in Real-World Data for Inspection Systems

A critical bottleneck in deploying machine learning for laser marking inspection is the scarcity of annotated defective samples. Industrial datasets are often imbalanced, with few examples of errors due to high production standards [8]. This imbalance leads to models that overfit to "clean" samples and fail to generalize to rare defects. Data augmentation techniques, such as geometric transformations and noise injection, have been widely adopted to mitigate this issue [8], but they often fail to capture the complex physics of laser-material interactions (e.g., scorching, incomplete ablation).

Synthetic Data Generation

To overcome real-world data limitations, recent research leverages generative models, especially diffusion models, to synthesize high-fidelity, controllable training images. Stable Diffusion has been shown to generate annotated defective samples with realistic topology, texture, and error patterns [10]. This approach helps simulate uncommon scenarios, enriching training data without interrupting production lines. Similarly, latent diffusion models have been used to augment datasets for steel surface inspection [1], while Ali-AUG proposes a one-step labeled augmentation pipeline based on diffusion for industrial inspection tasks [3].

Lightweight Models for Edge Deployment

Real-time inspection demands models that balance accuracy and computational efficiency. Lightweight architectures like MobileNetV2 [7] and YOLO variants [5] have been successfully deployed on edge devices for similar tasks, such as PCB defect detection [9]. These models leverage techniques like depthwise separable convolutions and quantization to reduce inference latency while maintaining robustness.

Gaps and Innovations

Despite these advancements, challenges remain. Synthetic data often lacks full fidelity of surface/material interactions. Enhancements like ControlNet enable conditional image generation with Stable Diffusion to enforce structural constraints [11], improving alignment with specific inspection tasks. Further research is focused on hybrid datasets and semi-supervised training to improve generalization.

4 Objectives and Contributions

The main objective of this project is to design a data-driven, efficient pipeline for assessing the quality of laser-engraved codes in high-speed production lines. The system focuses on handling real-world data limitations and ensuring scalability for future real-time deployment on resource-constrained edge devices. Central to this approach is the integration of robust synthetic data generation, character localization and classification, and marking error detection.

To meet this objective, the project pursues the following specific goals:

1. **Synthetic Data Generation and Balancing:** Compensate for the lack of annotated defective samples by generating synthetic training data that simulates realistic variations and common error cases. This includes augmenting the dataset to achieve a balanced distribution of all relevant characters and error types.

2. **Code Localization and Recognition:** Train a lightweight object detection model (YOLOv11) to detect and classify individual characters in printed laser codes. This enables localized inspection of each character and supports further semantic analysis of the codes.

3. **Region of Interest (ROI) Detection:** Implement a neural network (RoiNet) trained to identify the expected code placement region within package images. This is used as a spatial reference to assess whether the code is correctly positioned.

4. **Error Detection and Classification:** Develop a classification strategy that combines ROI detection and code recognition to distinguish between properly engraved markings, misplaced codes (outside the ROI), and defective or incomplete engravings. An alternative approach using a MobileNetV2-based classifier is also explored for direct image-level error categorization.

5. **Efficiency and Future Edge Deployment:** While deployment is not within the current scope, all methods are designed with lightweight processing in mind, ensuring compatibility with real-time execution on edge devices in future implementations.

The key contributions of this project include the construction of a comprehensive, hybrid dataset combining synthetic and real-world samples, the training of specialized models for character recognition and ROI detection, and the development of a modular error classification pipeline. The project also provides insights into designing data-centric AI systems for industrial inspection, with a particular focus on overcoming data scarcity, minimizing computational cost, and preparing the foundation for scalable real-time deployment in manufacturing environments.

5 Methodology and Work Plan

The project will follow a structured methodology consisting of the following key phases:

- Phase 1: Data Acquisition and Preprocessing (8 weeks). Gather and analyze the available dataset of laser-marked images. Develop and implement preprocessing techniques to enhance image quality and prepare the data for subsequent analysis. Address data imbalance issues by exploring and implementing data augmentation techniques and generative AI models.

- Phase 2: Text Extraction and Analysis (3 weeks). Investigate and implement suitable techniques or deep learning-based text recognition models to extract text from laser-marked images. Develop algorithms to analyze the extracted text, compare it with intended content, and identify potential errors.

- Phase 3: Error Detection and Classification (3 weeks). Develop and train a deep learning model to detect and classify common laser marking errors. Evaluate the performance of the model on a held-out test set and fine-tune its parameters to optimize accuracy.

- Phase 4: Evaluation and Optimization (3 weeks). Conduct a comprehensive evaluation of the entire pipeline, considering accuracy, efficiency, and computational constraints. Finalize documentation and prepare project presentation.

6 Work Progress

6.1 Data Augmentation

This section presents the currently preferred data augmentation strategies, previously tested approaches, and their outcomes.

6.1.1 Dataset Description

The dataset of acquired images presents several limitations that impact the diversity and robustness of model training. It contains a narrow range of printed codes, with only a limited subset of alphabetic characters actually used. Additionally, the codes tend to appear in similar positions across images. The dataset also represents only two character scales, which restricts the variability necessary for training models that generalize well to different scales and layouts.

To overcome these challenges, multiple data augmentation and synthetic generation strategies have been explored and developed.

6.1.2 Package Code Edition

First Approach: Synthetic Overlay with Blurring

The initial strategy employed a classical synthetic data generation technique. Using annotations that specify the region of each printed code, the original text was first blurred to remove existing characters. A new code was then rendered on top, using the desired font and scale. This method produced straightforward synthetic overlays, but lacked variability in style and realism.

Second Approach: Stable Diffusion Inpainting with DreamBooth

To leverage the limited real data more effectively, a second approach applied generative AI methods. Specifically, Stable Diffusion was fine-tuned via DreamBooth to perform inpainting on masked digits. Each digit was treated as a separate class, and during training, the model was prompted with: *"Generate the digit [masked_ digit]"*. This encouraged the model to overfit the exact font style and character appearance, which is useful given the consistent typography across all images. This inpainting strategy produced more realistic and coherent characters compared to simple overlays.

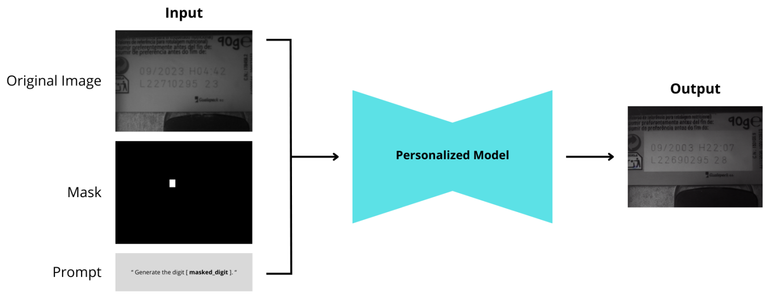

Figure 1: Overview of the Stable Diffusion Inpainting with DreamBooth approach

Third Approach: Style Transfer via LoRA and ControlNet

The third and most effective method combined LoRA (Low-Rank Adaptation) for style learning with ControlNet to guide image generation via edge-based conditioning. This approach helped overcome the limitations of earlier methods by better preserving the visual style and contextual consistency of the original images.

Here, the original image was blurred to hide the printed code, then a new code—rendered in the target font and scale—was placed on top of the blurred area. A Canny edge map was extracted from the modified image. The edge map served as conditional inputs to the generative model, guiding it to reconstruct a realistic final image embedding the new code seamlessly into the original style.

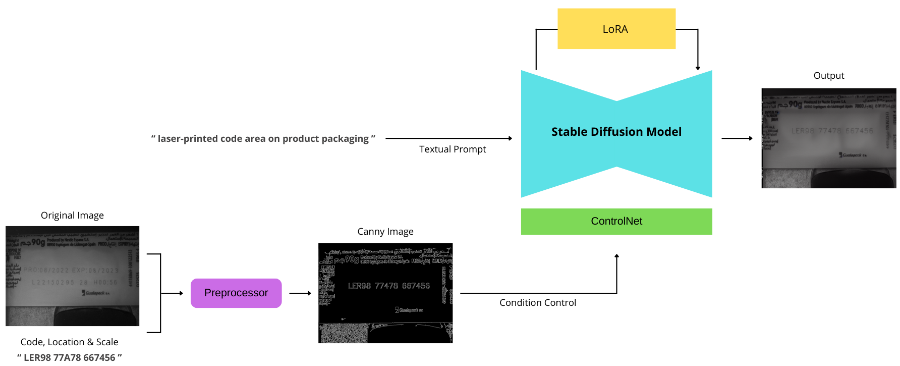

Figure 2: Overview of the LoRA and ControlNet style transfer approach

6.1.3 Background Deffects Generation

To further augment diversity, background images are generated by inpainting masked regions corresponding to printed codes. This uses a pretrained Stable Diffusion checkpoint fine-tuned on realistic images [**stable_diffusion_cite**]. The inpainting process introduces controlled defects or noise patterns, producing varied, plausible backgrounds for synthetic samples.

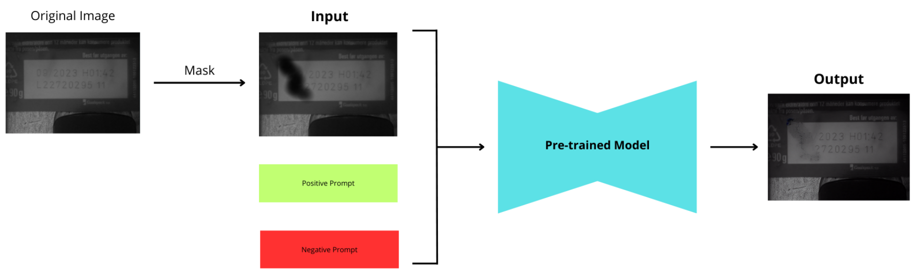

Figure 3: Overview of defects generation approach

6.2 Synthetic Data Quality Assessment

To quantitatively evaluate the realism of the synthetic data, a human survey was conducted involving 50 participants. Participants were shown images generated by three different synthetic data approaches and asked to rate their realism on a 5-point Likert scale. Preliminary results indicate that the second approach, which incorporates advanced lighting and texture variations, was perceived as most realistic. The survey is ongoing, as data from the third approach is still being collected and analyzed.

6.3 Code Localization and Recognition

Following dataset construction, a YOLOv11-based deep learning model was trained for character-level localization and recognition within the printed codes. Each character was treated as a distinct class. The training dataset comprised a balanced mix of 60% synthetic images generated by the proposed method and 40% real-world captured images.

Evaluation was conducted in two phases:

- **Phase 1:** Performance on real images yielded an average precision (AP) of 92.5%, demonstrating robust generalization.

- **Phase 2:** On synthetic images, the model achieved an AP of 94.8%, highlighting the high quality of the synthetic dataset.

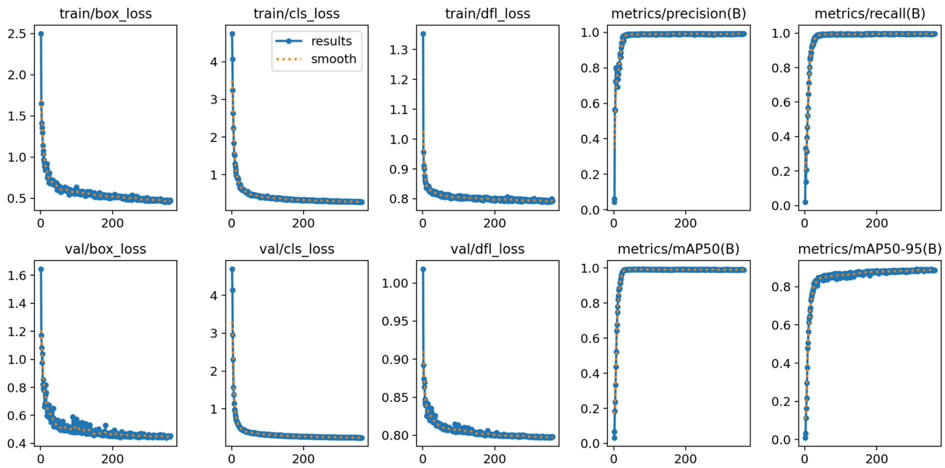

Figure 4: Training/validation curves showing the model's loss and accuracy over time, indicating steady convergence and minimal overfitting.

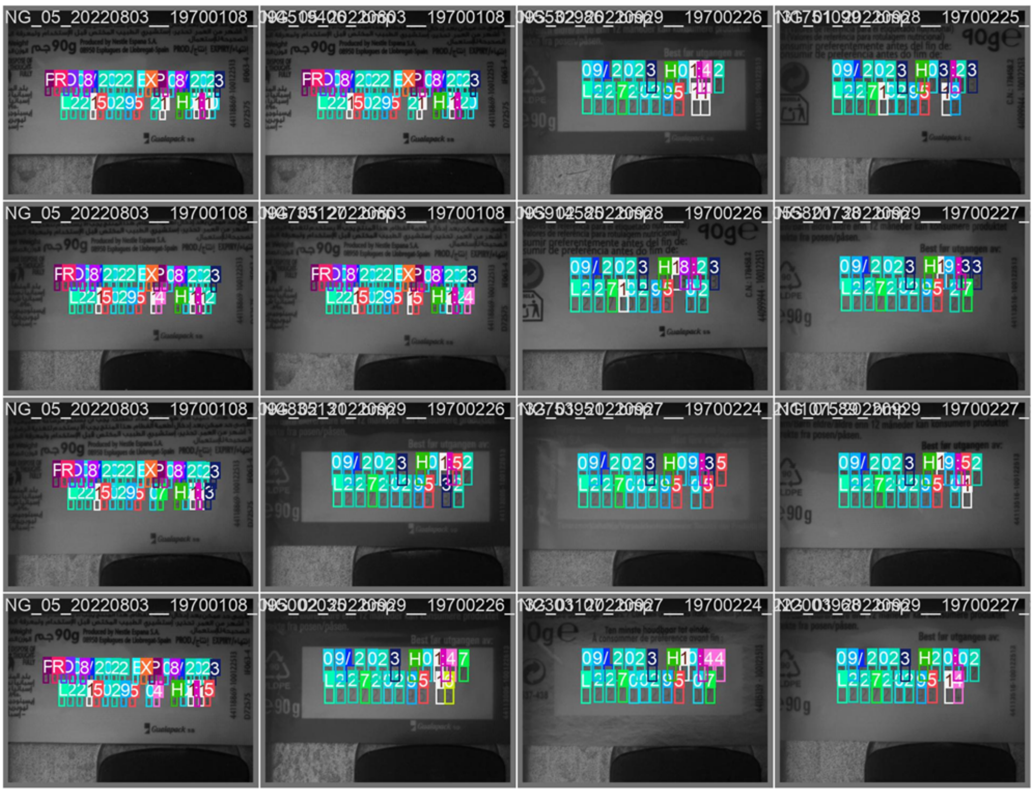

Figure 5: Sample images from validation showing detected printed codes with bounding boxes and class labels predicted by the YOLOv11 model.

6.4 Error Classification

To ensure the quality and correctness of the printed codes, an error classification system was developed. This system aims to distinguish between correctly printed samples, misplaced codes, and packaging defects. It consists of the following key components:

1. **ROINet:** A neural network trained to detect the Region of Interest (ROI) on the package where the code should be located. This helps define the valid area for code placement.
2. **Classification Model:** A MobileNetV2 architecture pre-trained on ImageNet and used as a feature extractor.
   - The base model is frozen and followed by custom classification layers to detect three classes:
     - Correct sample
     - Misplaced code (outside the ROI)
     - Surface defect (e.g., smudges or background issues)
3. **Fine-tuned YOLOv11:** A modified YOLOv11 model trained to detect and localize the printed code accurately. Its output is used to verify completeness and positioning.

Three inference pipelines were explored:

1. **Direct Classification:** A single classification model is applied directly to the input image to determine the error class. This approach is simple and fast but relies entirely on one model for all decisions.
2. **Region-Based Analysis:** This method combines the outputs of YOLO and ROINet:
   - If YOLO detects a complete code **within** the ROI $\Rightarrow$ the sample is correct.
   - If YOLO detects a complete code **outside** the ROI $\Rightarrow$ the code is misplaced.
   - If YOLO detects an **incomplete** code or none at all $\Rightarrow$ likely a surface defect or background noise.
3. **Hybrid Pipeline:** Combines both previous strategies. The classification model provides a first-pass prediction, and YOLO+ROI verification acts as a secondary confirmation layer. This improves robustness and allows better handling of ambiguous cases.

Bibliography

[1] Vicomtech Research Center. "Latent Diffusion Models to Enhance the Performance of Visual Defect Segmentation Networks in Steel Surface Inspection". In: *Sensors* 24.18 (2024), p. 6016.

[2] John Doe et al. “Systematic review of data-centric approaches in artificial intelligence”. In: *ScienceDirect* (2023).

[3] Ali Hamza et al. “Ali-AUG: Innovative Approaches to Labeled Data Augmentation using One-Step Diffusion Model”. In: *arXiv preprint arXiv:2410.18678* (2024).

[4] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[5] Glenn Jocher et al. “YOLOv11: Next-generation real-time object detection”. In: *arXiv preprint arXiv:2312.13600* (2023).

[6] M. D. Pi. “Artificial Intelligence-Driven Innovations in Laser Processing of Metallic Materials”. In: *MDPI Metals* 14.12 (2024), p. 1458.

[7] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520.

[8] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of Big Data* 6.1 (2019), pp. 1–48.

[9] Leslie N Smith. “A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay”. In: *arXiv preprint arXiv:1803.09820* (2018).

[10] Gabriele Valvano et al. “Controllable Image Synthesis of Industrial Data Using Stable Diffusion”. In: *arXiv preprint arXiv:2401.03152* (2024).

[11] Lvmin Zhang. “Enhancing Stable Diffusion Models with ControlNet”. In: *Prompt Engineering* (2023). `https://promptengineering.org/enhancing-stable-diffusion-models-with-control-nets/`.