PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
الجمهورية الجزائرية الديمقراطية الشعبية
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليم العالي والبحث العلمي
HIGHER SCHOOL OF COMPUTER SCIENCE AND DIGITAL TECHNOLOGIES-BEJAIA
المدرسة العليا في علوم وتكنولوجيات الاعلام الالي بجاية

Dissertation Submitted to the Department Of Computer Science in Partial Fulfillment of the Requirements for Engineer's Degree in Computer Science

Specialty: Artificial Intelligence and Data Sciences

Submitted By:

Adel BELLAHCENE

Computer Vision Based Industrial Inspection System

Supervised by:

Pr. Ahror BELAID
CERIST

Members of jury:

- **Dr.** Ali DJENADI — President — ESTIN
- **Dr.** Chemseddine BERBAGUE — Examiner — ESTIN
- **Mrs.** Nassima SLIMANI — Examiner — ESTIN
- **Mrs.** Sofia CHERGUI — Examiner — ESTIN

Academic year: 2023/2024

Appreciations

As I reach the culmination of my engineering thesis, I find myself reflecting on the journey that brought me here, filled with gratitude for the incredible support I have received along the way.

First and foremost, I would like to express my heartfelt appreciation to the Data Science Research Unit (DSRU). Your belief in my potential and the invaluable opportunity you provided me have been instrumental in shaping this project. The resources, guidance, and expertise offered by DSRU were pivotal in navigating the complexities of implementing a computer vision-based industrial inspection system. Thank you for your unwavering support and for fostering an environment of innovation and excellence.

To Toudja Factory, thank you for opening your doors to us and providing a real-world setting to test and deploy our system. Your collaboration and willingness to integrate new technologies into your production line have been crucial to the success of this project. Your team's openness and cooperation made this journey not only possible but also deeply rewarding.

I am profoundly grateful to my school, which laid the foundation for my engineering education. A special thanks to my mentor, whose guidance, wisdom, and encouragement have been a source of inspiration throughout my academic career. Your unwavering belief in my abilities and your dedicated mentorship have profoundly impacted my journey. I am deeply appreciative of your patience and the countless hours you invested in my growth.

To my family, who have been my rock throughout this endeavor, thank you for your unconditional love and support. To my friends and comrades, your encouragement and camaraderie have been invaluable. A special mention to my wife, whose steadfast support, patience, and understanding have been my anchor. Your belief in me kept me motivated, even during the most challenging times.

Lastly, I extend my gratitude to everyone who contributed to the success of this internship and my career, whether from near or far. Your contributions, big or small, have played a part in this journey. Thank you for being a part of this remarkable experience and for helping me achieve this milestone.

With deep appreciation and heartfelt thanks,

ADEL BELLAHCENE

List of AbbreviationsANN. Artificial Neural Network

AI. Artificial Intelligence

ASM. Attribute Selection Measure

AVI. Automated Visual Inspection

BAFPN. Bidirectional Attention Feature Pyramid Network

CMOS. Complementary Metal Oxide Semiconductor

CNN. Convolutional Neural Network

CoBot. Collaborative Robots

CPU. Central Processing Unit

CV. Computer Vision

DETR. Detection Transformer

DL. Deep Learning

DNN. Deep Neural Network

DPM. Deformable Part Models

DSC. Dice-Sorensen Coefficient

FLOPS. Floating-Point Operations Per Second

FN. False Negatives

FP. False Positives

FPS. Frames Per Second

GIGO. Garbage In Garbage Out

GPU. Graphics Processing Unit

IDS. Imaging Development Systems

IoT. Internet of Things

IoU. Intersection over Union

K-NN. K-Nearest Neighbors

mAP. mean Average Precision

ML. Machine Learning

NDT. Non-Destructive Testing

PCA. Principal Components Analysis

PCB. Printed Circuit Board

QC. Quality Control

R-CNN. Region based Convolutional Neural Network

RGB. Red Green Blue

RoI. Region of Interest

RoI. Return on Investment

SNN. Simulated Neural Network

SSD. Single Shot Detector

SVD. Singular Value Decomposition

SVM. Support Vector Machine

TN. True Negatives

TP. True Positives

VGG. Visual Geometry Group

VI. Visual Inspection

XAI. Explainable AI

YOLO. You Only look Once

Table of Content

List of Tables

List of Figures

General Introduction

Inspection serves as a cornerstone in quality control and constitutes an essential component of the manufacturing process. This procedure is pivotal for the industry as it identifies defects early in the production line, leading to significant savings in time and money, while also enhancing customer satisfaction by ensuring that the product adheres to established standards. Beyond financial savings, industrial inspection is instrumental in safeguarding workers' lives by detecting defective and faulty machinery [1]. The most prevalent form of inspection in industries is Visual Inspection (VI), which involves analyzing the product based on visual aspects such as color, shape, and the presence of cracks during production. VI is applicable not only to products but also to machinery. As a type of NonDestructive Testing (NDT), VI does not alter the product post-inspection, thereby making it the preferred technique in various industries. Ensuring that substandard products do not reach the end customer is imperative in manufacturing [2] [3] [4].

For decades, quality control (QC) has relied on human or mechanical testing. However, numerous factors render human involvement insufficient in contemporary times, including issues such as fatigue, delayed reaction times incompatible with production line speeds, and diminished accuracy. Consequently, the adoption of Automated Visual Inspection (AVI) becomes imperative, particularly in contexts involving hazardous materials like radioactive substances or chemicals. AVI entails the utilization of computer technology and cameras in visual inspection to automate the process. An AVI system can be deconstructed into five key stages: image acquisition, image enhancement, segmentation, feature extraction, and classification, which determines whether to reject or accept the product. In the late 1990s, industries began striving to integrate Artificial Intelligence (AI) into AVI systems. This integration involved implementing rule-based systems such as expert systems, as well as employing machine learning (ML) algorithms and artificial neural networks (ANN) to enhance accuracy and precision [2].

In contemporary times, Artificial Intelligence (AI) pervades various domains, ranging from healthcare to finance, and has seamlessly integrated itself into the fabric of manufacturing. Industries demand systems that are not only efficient and accurate but also cost-effective, precisely the attributes that AI effortlessly provides. Representing a paradigm shift in the field, AI surpasses human capabilities in every aspect, spanning from reaction time to cost efficiency. Industries are leveraging the power of Deep Learning (DL) and its potential for rapid, precise, and economical solutions. The utilization of Machine Learning (ML) algorithms in image and video analysis facilitates the creation of real-time defect detection systems. For instance, INTEL's integration of AI in weld defect detection resulted in a noticeable improvement in productivity rates. Moreover, considering that inspections often occur in external environments, the integration of AI in inspection robots or drones holds tremendous promise. Terra Drone's extensive use of AI in pipeline inspections yielded remarkable outcomes, with a 75% increase in speed and a reduction of approximately 50% in costs [1] [3] [5] [6].

More sophisticated AI approaches can be leveraged to enhance AI-powered AVI, such as Transfer Learning. Transfer learning is an ML technique wherein the knowledge acquired by a model performing one task (task A) is reused to boost or perform a related, albeit different task (task B). This technique offers the advantage of requiring less time, data, and computational resources.[8]

Furthermore, AI can be integrated with the Internet of Things (IoT). The fusion of AI and IoT, often referred to as AIoT, yields exceptional performance. IoT devices and sensors serve as the senses of the system, capturing data from the environment and interacting with it. The AI model acts as the brain, where analysis is conducted, and decisions are made.[9]

Modern manufacturing heavily relies on automated visual inspection (AVI) systems powered by Artificial Intelligence (AI). This review bridges the gap between theoretical advancements and practical application by focusing on the implementation of an AI-based visual inspection system using YOLO models. The review delves into the practical aspects of implementing an AI model, encompassing the entire process from data acquisition and pre-processing to model selection, training, deployment, and post-processing. This knowledge is valuable for researchers, engineers, and industry professionals seeking to optimize their manufacturing operations by deploying AI-driven visual inspection technology.

Chapter 1: Overview of Industrial Inspection

Industrial inspection plays a critical role in ensuring product quality and safety across diverse industries. This chapter delves into the evolution of inspection techniques, tracing their journey from the limitations of manual methods to the cutting-edge advancements in Artificial Intelligence (AI)

1.1 Traditional Inspection Methods

Traditional inspection has long been the backbone of quality control (QC), involving the physical examination of products by trained technicians known as inspectors. These inspectors meticulously examine the object both visually and manually. In the pre-digital era, inspectors were trained to identify defects, sometimes with the naked eye and, in other instances, using simple tools such as lights and magnifying glasses, see Figure 1. The technicians conducted their inspection routines using a checklist on paper, detailing the conditions and standards that the final product must meet. Each product aspect must be scrutinized; for example, in the automobile industry, inspectors must check for defects inside and outside the vehicle, including the engine and safety measures. [10]

Figure 1. Technician Manually Inspecting a Product [115]

1.1.1 Advantages

- Versatility: Manual inspection is adaptable to a wide range of products and defect types, making it suitable for diverse manufacturing environments.

- Flexibility: Human inspectors can exercise judgment and adapt their inspection approach based on product characteristics and specific defect criteria.

1.1.2 Limitations

- Subjectivity: Product assessment is influenced by the inspector's individual experience and skill level, potentially leading to inconsistencies in defect detection.
- Fatigue and Inconsistency: Inspectors' attention and focus can wane over time, increasing the likelihood of errors and missed defects.
- Limited Speed: Manual inspection may not be suitable for high-volume production lines where rapid defect detection is crucial.

1.2 Automated Visual Inspection

As the industry evolved, traditional methods became inadequate for quality control (QC). The limitations of human inspectors prompted industries to develop, elaborate, and transition to automated inspection systems. Initially, computerized tools were employed to assist technicians in performing their tasks.[5]

1.2.1 Definition

Automated Visual Inspection (AVI) is the automated process of quality control for manufactured products, achieved through the use of a camera connected to a computer. AVI is a branch of machine vision, which necessitates proper lighting, a camera, and hardware for processing, typically a computer, see the elaborated schema in Figure 2. During the processing phase, image processing functions such as thresholding, edge detection, and basic segmentation are utilized. In summary, the camera captures an image, sends it to the computer for processing, where a decision is made. Subsequently, the handling system takes action based on the decision made.[2]

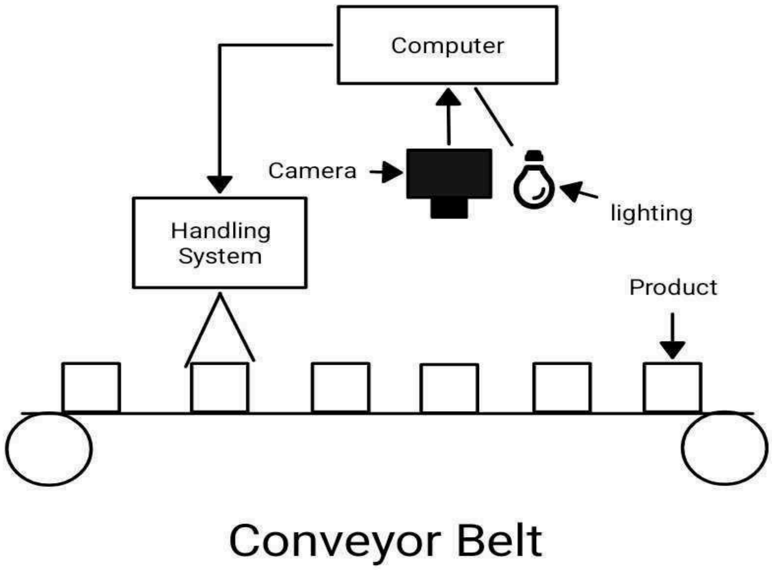


Figure 2. general AVI Schema [primary]

1.2.2 How it Works

AVI systems operate by following a series of stages in the process, utilizing various tools and techniques.

A. Image Acquisition

The acquisition of images of the object to be inspected is achieved using one or more cameras placed at different angles, such as front and top views, while the object moves along the production line.

B. Image Enhancement

Image enhancement, also known as image preprocessing, improves the quality of the acquired image to facilitate later processing stages. Basic functions such as resizing and deblurring are employed to ensure the image matches the required template.[2]

C. Segmentation

Segmentation is the most critical step in the system. During this stage, image processing functions are used to extract features from the preprocessed image, which will be utilized in the classification stage.

- Thresholding: Assigns a binary value to each pixel based on a global threshold value, helping to segment objects from the background.

- Edge Detection: Applies filters to detect edges in the image, aiding in obtaining the contours of the desired object.[2]

D. Feature Extraction

By applying transforms such as Fourier transforms, Hough transforms, or statistical approaches to the segmented image, a set of features describing the image is obtained.[2]

E. Classification

Once a set of features is extracted, classification is performed, which involves assigning a class to the image based on a specific model. Initially, Bayes' Theorem [14] was used to accomplish this task. The theorem states:

$$P(A|B) = P(B|A)P(A)/P(B) \; \ldots \; Eq(2) \; | \; P: \text{probability}, A: \text{class}, B: \text{features}$$

The probability that the object in the image belongs to class A given features B is the product of the probability of observing features B given class A and the probability of class A, divided by the probability of observing features B. This calculation is done for each class, and the class with the highest probability is selected . Once the classification is complete, the system makes the decision to accept or reject the product.[2]

The Figure 3 below shows the System Steps

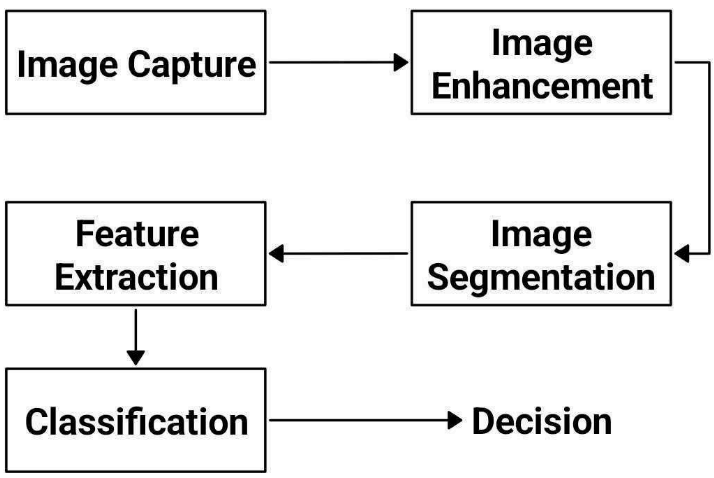


Figure 3. AVI System steps [primary]

1.2.3 Advantages

The advantages are, as shown in Figure 4:

- Speed and Efficiency

- Consistency and Accuracy
- Reduced Costs
- 24/7 Operation
- Data-Driven Insights

Figure 4. Benefits of AVI Systems [12]

1.2.4 Limitations

- Complexity of Inspection Tasks
- Lighting and Environment
- Integration with Existing Processes

1.3 Conclusion

This chapter has charted the course of industrial inspection, tracing its evolution from rudimentary manual methods to the sophisticated automation of today. While manual inspection laid the groundwork, its limitations spurred advancements in automated visual inspection (AVI). The chapter highlighted the growing need for ever-more robust and reliable solutions.

Chapter 2: AI Integration In industrial Inspection Systems

Today, the terms AI, ML, and DL are often used interchangeably, although they have distinct meanings. In this chapter, we will delve into these concepts.

The term "Artificial Intelligence" (AI) embodies the idea of machines replicating human thought processes. The seminal 1956 Dartmouth Summer Research Project proposal offered this foundational definition:

"The conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

The term "Artificial Intelligence" conjures up images of machines tackling challenges once thought to require human intellect and reasoning, see Figure 5.[18]

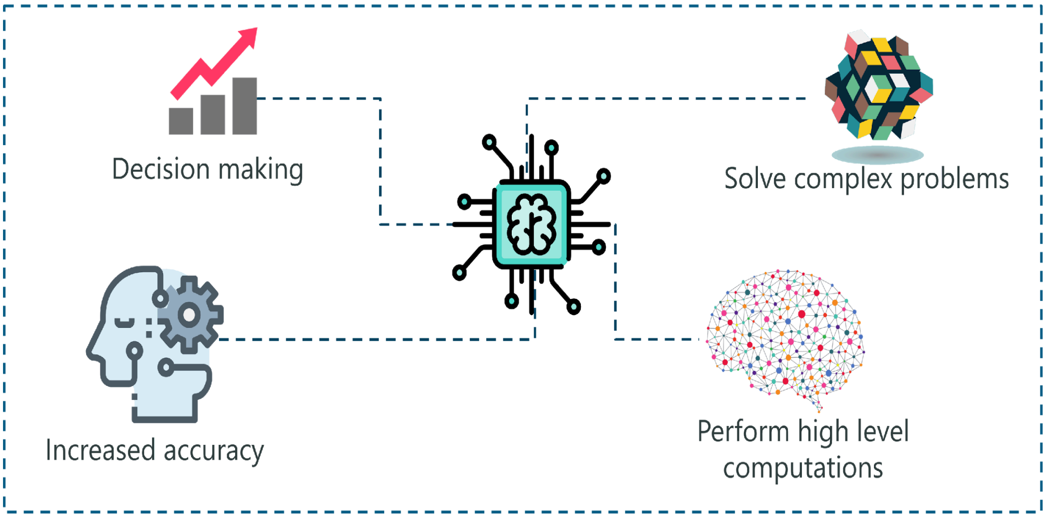


Figure 5. Objectives of AI [114]

We can say that AI is the superset, while ML and DL are subsets, as illustrated in Figure 6

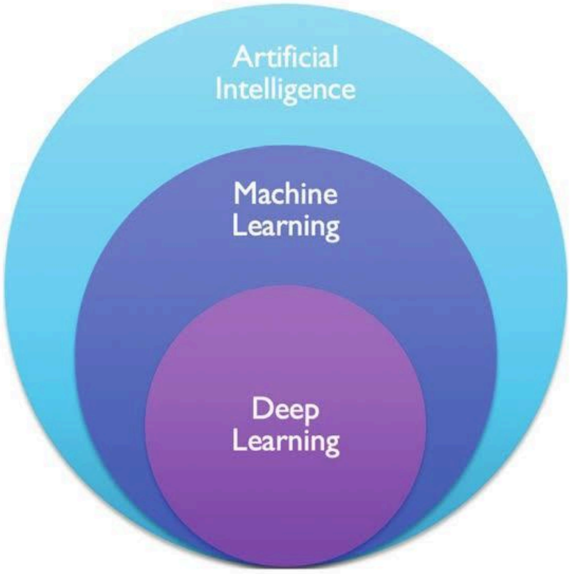


Figure 6. AI Subdivision [117]

2.1 Machine Learning

Machine learning (ML), a branch of AI, allows machines to learn from data and past experiences to make predictions with minimal human intervention. Unlike traditional programming, ML algorithms can learn and adapt on their own when fed new data. Through iterative analysis of large datasets, they identify patterns and gain insights without relying on predefined models. As they process more data, their performance improves [20][107]

2.1.1 Supervised Learning

Supervised learning trains algorithms using labeled data to make accurate classifications or predictions. The model refines itself by adjusting weights during training to avoid overfitting or underfitting the data, as Figure 7 shows. This technique is used in various applications, like spam filtering, and employs algorithms like neural networks and linear regression. This method is used to train a model in order to predict some result as we give new data as input. it may be a class so it's classification or Continuous value then it's a regression. for example, we have a model that takes a picture of a bottle and tells as if it is empty or full, in this case this is classification, but if the model returns the percentage of its emptiness then it's regression. [19][107]

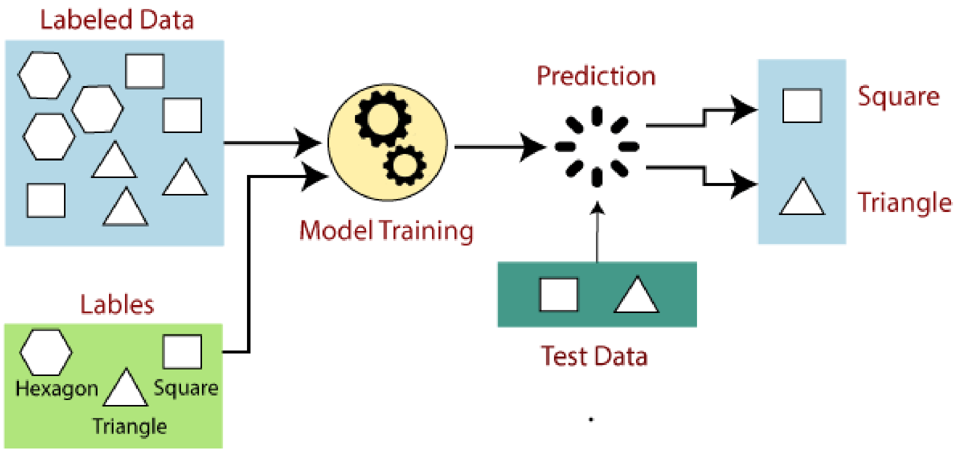


Figure 7. How Supervised learning works[108]

2.1.2 Unsupervised Learning

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets, identifying hidden patterns without human intervention, see Figure 8 below. It is useful for exploratory data analysis, cross selling strategies, customer segmentation, image, and pattern recognition. It also aids in dimensionality reduction using methods like principal component analysis (PCA) and singular value decomposition (SVD). Common algorithms in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods.[107]

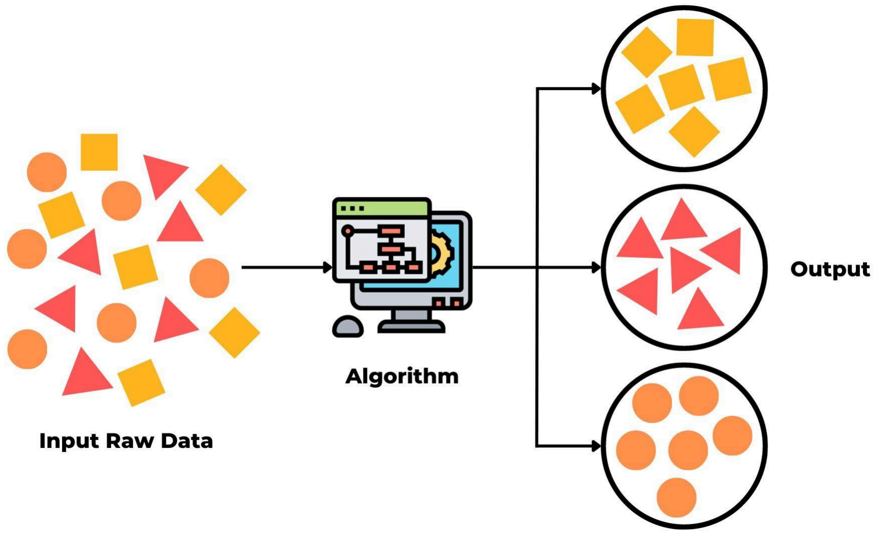


Figure 8. How Unsupervised learning works[109]

2.1.3 How ML work

The process of training and utilizing a machine learning model. Here's a description of the steps involved:

1. Training Data: The process begins with collecting training data, which is the dataset used to train the machine learning (ML) algorithm.
2. Train ML Algorithm: The training data is fed into the ML algorithm to create a model. The model is built by learning patterns from the training data.
3. Model Input Data: Once the model is trained, it can be tested with input data to evaluate its performance.
4. Accuracy: The performance of the model is assessed based on its accuracy. This step determines whether the model's predictions are acceptable or unacceptable.
   - Unacceptable: If the model's accuracy is unacceptable, the process loops back to training the ML algorithm, possibly with adjustments or more data.
   - Acceptable: If the accuracy is acceptable, the model is considered successful.
5. Successful Model: An acceptable model is termed as a successful model and can be used for making predictions.
6. New Input Data: The successful model is then used to make predictions on new input data.
7. ML Algorithm and Prediction: The ML algorithm processes the new input data and makes predictions based on the learned patterns from the training phase.

The Figure 9 summarize the ML process flow

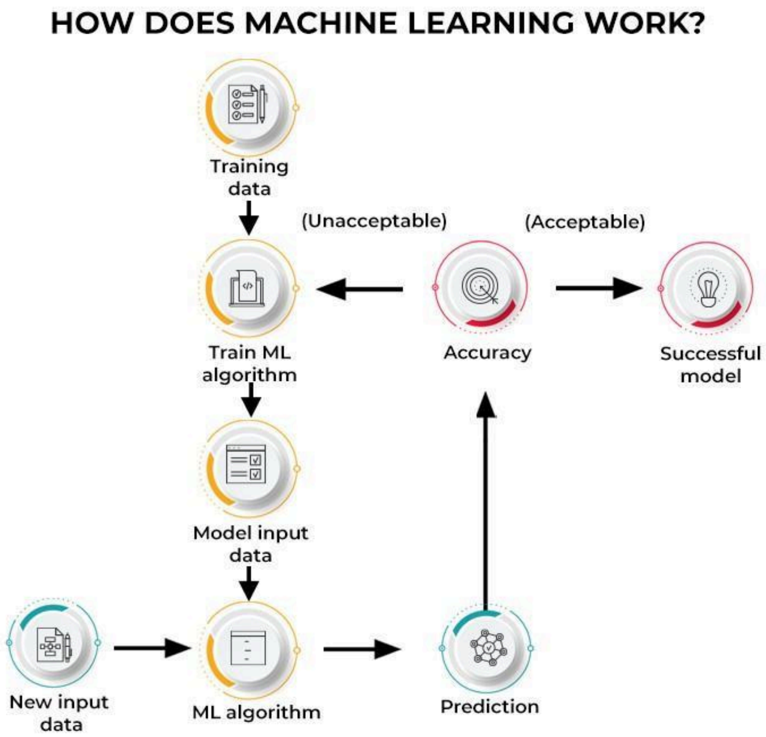


Figure 9. ML Process Flowchart [20]

2.1.4 Metrics

Accuracy, recall, precision, and F1-score are commonly used metrics in machine learning and statistical analysis to evaluate the performance of classification models. Here's a brief explanation of each:

A. Accuracy:

Accuracy measures the proportion of correctly classified instances among all instances evaluated by the model. It is calculated as the ratio of the number of correct predictions to the total number of predictions made by the model.

$Accuracy = (TP + TN) / (TP + TN + FP + FN)$ … Eq(3) | TP: true positive, TN: true negative, FP: false positive, FN: false negative

B. Recall (Sensitivity):

Recall measures the proportion of correctly identified positive instances out of all actual positive instances in the dataset. It is also known as sensitivity or true positive rate.

$Recall = TP / (TP + FN)$ … Eq(4) | TP: true positive, TN: true negative, FP: false positive, FN: false negative

C. Precision:

Precision measures the proportion of correctly identified positive instances out of all instances predicted as positive by the model. It focuses on the accuracy of positive predictions.

Precision = TP / (TP + FP) … Eq(5) | TP: true positive, TN: true negative, FP: false positive, FN: false negative

D. F1-score:

F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance by considering both precision and recall. F1-score is especially useful when dealing with imbalanced datasets where the number of positive instances differs significantly from the number of negative instances.

F1-score = 2 * (Precision * Recall) / (Precision + Recall) … Eq(6)

These metrics are crucial for assessing the effectiveness and reliability of classification models across various domains and applications. They provide valuable insights into a model's ability to correctly classify instances and identify potential areas for improvement.

2.2 Deep Learning

Deep learning is a subset of machine learning that utilizes multi-layered neural networks, known as deep neural networks (DNNs), to replicate the complex decision-making capabilities of the human brain. This advanced form of AI is prevalent in many aspects of modern life.

2.2.1 Key Features of Deep Neural Networks (DNNs):

- Structure: DNNs have three or more layers, often with many more in practice, allowing for intricate processing.
- Training: They are trained on large datasets to recognize patterns, classify data, evaluate possibilities, and make accurate predictions.
- Layers: Additional layers in DNNs enhance the refinement and optimization of predictions compared to single-layer neural networks.

2.2.2 Functionality:

- Forward Propagation: Data progresses through the network's layers, each layer refining and optimizing predictions based on the previous layer's output.
- Input and Output Layers: The input layer receives data for processing, while the output layer provides the final prediction or classification. These layers are called visible layers.
- Backpropagation: This process involves using algorithms like gradient descent to identify and correct prediction errors by adjusting weights and biases, moving backward through the layers. This iterative process improves the model's accuracy over time.

2.2.3 Applications and Impact:

Deep learning models can accurately recognize, classify, and describe objects within data, making them powerful tools for tasks like image and speech recognition, and other AI applications. The combination of forward propagation and backpropagation allows neural networks to learn from data, make predictions, and iteratively improve their accuracy. [25]

2.3 Convolutional Neural Networks CNNs:

Convolutional neural networks, unlike other neural networks, are particularly good at processing images, speech, and audio. They achieve this through three main layers: convolutional layers, pooling layers, and fully-connected layers. Convolutional layers, the first type, identify increasingly complex features in the data as it progresses through the network, Figure 10. Early layers recognize basic elements like colors and edges, while later layers can identify entire objects.[110]

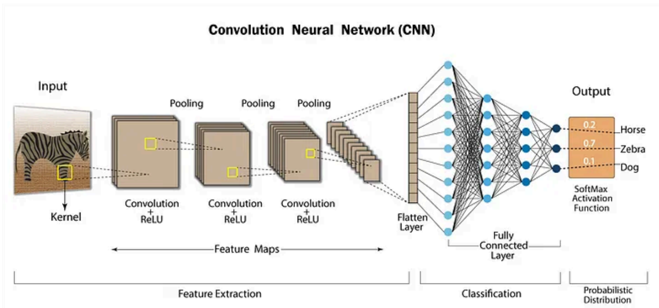


Figure 10. Convolutional Neural Network (CNN) [112]

2.3.1 Famous CNNs Architectures:

Over the years, convolutional neural networks (CNNs) have evolved significantly to address specific needs and challenges in various domains. Some of the renowned architectures that have made a profound impact include, as mentioned in [110]:

A. ResNet:

Residual Networks (ResNets), a pioneering architecture for deep convolutional neural networks, have significantly impacted the field of image recognition since their introduction in 2015 (see Figure 11). ResNets' influence is attributed to several key factors:

1. Depth Capability: ResNets can be extremely deep, exceeding 150 layers. This exceptional depth enables them to learn highly complex patterns within images, which was previously challenging for shallower networks.
2. ImageNet Performance: ResNets dominated the ImageNet classification competition, securing first place. They also excelled in related tasks such as object detection and segmentation, demonstrating their superior performance.
3. Generalization: Beyond accuracy, ResNets are known for their adaptability. Their ability to generalize knowledge across various recognition tasks makes them a highly versatile tool in the field of computer vision. [28][110]

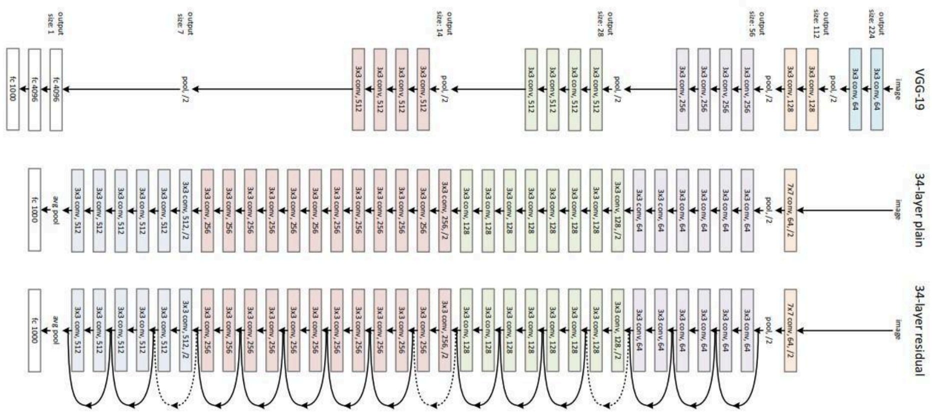


Figure 11. ResNet 34 Architecture[28]

B. VGG:

The VGG architecture, depicted in Figure 12, is renowned as a classic convolutional neural network (CNN) model. Unlike some of its contemporaries, VGG emphasizes depth over complexity. Its distinctive feature lies in the use of numerous simple 3x3 convolutional filters, which are stacked together. This straightforward yet effective approach has demonstrated significant success in image recognition tasks.

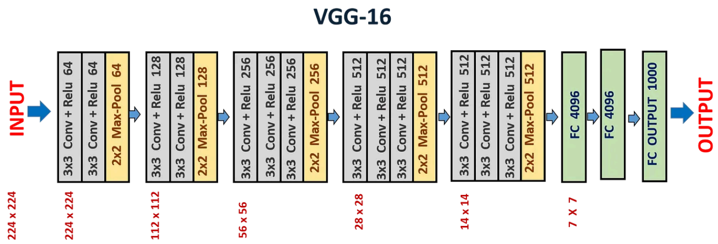


Figure 12. VGG 16 Architecture[29]

C. Inception:

Inception v3 is a powerhouse convolutional neural network (CNN) that builds on the success of the Inception family. This architecture introduces several key improvements:

- Label Smoothing: This trick helps the network make more confident predictions during training.
- Factorized 7x7 Convolutions: These convolutions are like smaller, more efficient building blocks that capture complex features in images.
- Auxiliary Classifier: This nifty addition helps the network learn better by introducing an extra classifier earlier in the process.
- Batch Normalization: This technique stabilizes the training process, making it smoother and faster. By incorporating these refinements, Inception v3 has become a popular choice for image recognition tasks.[31]

2.4 Computer Vision

Before we explore how machines use sight for quality control, let's get familiar with computer vision. This is a branch of AI that lets machines "see" and understand the world around them, just like humans do. The goal? To automate tasks that our eyes handle best. Imagine machines analyzing images, making decisions, and even understanding context – that's computer vision at work! It's used everywhere from healthcare to transportation, making our lives safer and more efficient.

So how does computer vision work? It's a three-step process:

- See It: The system captures an image or video using cameras or other sensors.
- Process It: The image data gets prepped, like resizing or adjusting brightness for better analysis.
- Understand It**:** Machine learning algorithms take over, identifying patterns, objects, and important features. Based on this understanding, the system can make decisions or predictions.

Advancements in machine learning and deep learning have supercharged computer vision, making it a powerful tool across many industries. [33]

2.4.1 Computer Vision Tasks

Seeing Through Images: Different Computer Vision Techniques

Image Classification: This is like recognizing objects in an image. Imagine a social media site automatically flagging inappropriate content.

Object Detection: Here, we not only recognize the object but also pinpoint its location in the image. This helps tasks like finding faulty equipment on a production line.

Object Tracking: This goes beyond detection – it follows an object's movement over time. Self driving cars use this to track pedestrians and other vehicles to avoid accidents. [33]

A. Object Detection

Object detection is a powerful computer vision technique that uses neural networks to *locate and classify objects* within digital images. Imagine it as training a computer to "see" objects like humans do. This technology has a wide range of applications, from medical imaging analysis to self-driving cars through industrial inspection.

There are two key subtasks involved in object detection:

1. Object Localization: This determines the *specific location* of an object in an image. It's like drawing a bounding box around the object to pinpoint its position.
2. Object Classification: This identifies the *category* an object belongs to. Is it a car, a person, or a dog?

By combining these subtasks, object detection can simultaneously *estimate both the location and type* of multiple objects in an image. This makes it a valuable tool for various tasks that require understanding the visual content of images.[34]

B. Image Segmentation

Image segmentation is a computer vision technique that goes beyond object detection. It dives deeper, meticulously dividing an image into distinct "segments" based on pixel characteristics. These segments can be like tiny puzzle pieces, revealing the shapes and boundaries of objects within the image.

This ability to extract detailed information makes image segmentation a powerful tool for various AI applications. From medical imaging analysis (segmenting tumors from healthy tissue) to robotics and self-driving cars (understanding the environment around them), image segmentation plays a crucial role in unlocking the potential of computer vision. [35]

C. Object Detection vs. Image Segmentation

- Goal:
    - Detection: Identify and pinpoint the location of objects in an image.
    - Segmentation: Delineate the exact boundaries of objects in an image.
- Output, see Figure 13:
    - Detection: Bounding boxes around objects with class labels (e.g., car, person).
    - Segmentation: A pixel-wise mask for each object in the image.

D. Choosing the Right Tool

- Object detection is ideal when you need to know what objects are present and where they are generally located. (e.g., Manufacturing products in the production line)
- Image segmentation is better suited for tasks requiring a more granular understanding of object shapes and boundaries. (e.g., medical imaging to segment tumors) [34]

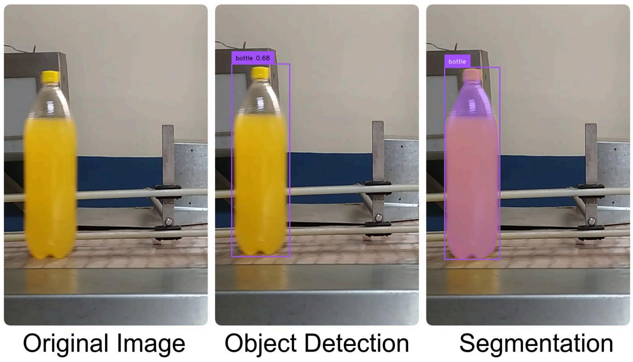


Figure 13. Object Detection vs Segmentation [primary]

2.4.2 Object Detection Models

In the domain of machine learning, object detection tasks leverage various approaches. While traditional methods like the Viola-Jones framework and histogram of oriented gradients were prominent, recent advancements have prioritized convolutional neural networks (CNNs) for object detection. This work focuses on two widely discussed CNN architectures: R-CNN (region-based convolutional neural network) and YOLO (You Only Look Once). Benchmark datasets like Microsoft COCO and ImageNet serve as evaluation tools for these models. [34]

While the focus lies on YOLO, it's important to acknowledge the existence of numerous alternative model architectures. SSD and RetinaNet share a simplified architecture similar to YOLO. DETR, a recent development from Meta (formerly Facebook), leverages a combination of CNN and transformer models, achieving performance comparable to Faster R-CNN.[34][38][36]

A. YOLO:

This family of single-stage detection architectures, as illustrated in Figure 14, prioritizes speed and is built upon Darknet, an open source CNN framework. Introduced in 2016, YOLO prioritizes speed, making it well-suited for real-time object detection applications. Unlike R-CNN's multi-stage approach with separate feature extraction and classification networks, YOLO condenses these processes into a single network, see Figure 15, . Furthermore, YOLO generates significantly fewer bounding box predictions per image (less than 100) compared to R-CNN's 2,000 region proposals. While boasting superior speed and lower background false positives, YOLO exhibits higher localization errors. Since its inception, YOLO has undergone numerous updates, primarily focused on optimizing speed and accuracy. [34][37]

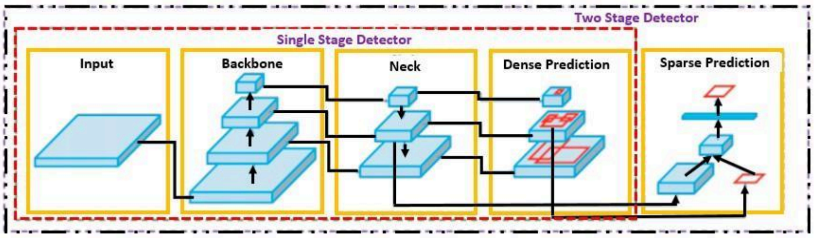


Figure 14. Single-Stage vs Two-Stage Detectors Architecture [41]

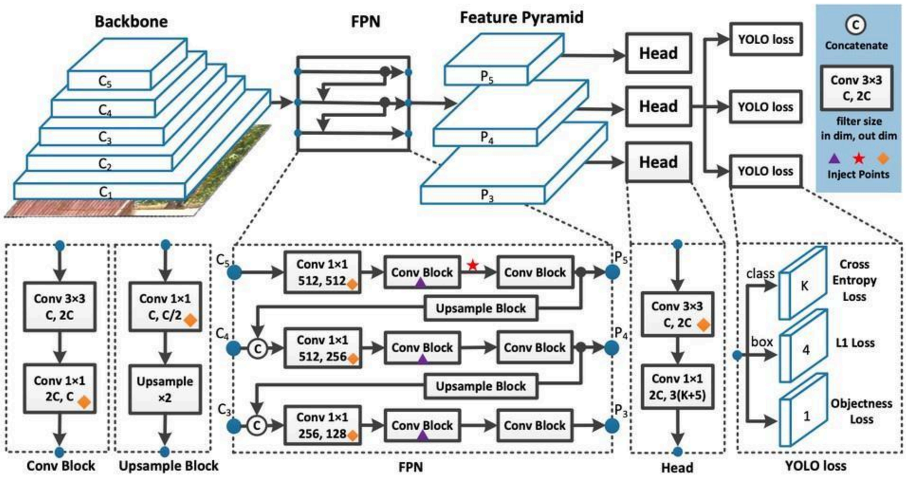


Figure 15. YOLO Architecture [113]

2.4.3 Object Detection Evaluation Metrics

A. Intersection over Union (IoU):

IoU measures the overlap between the predicted bounding box and the ground truth bounding box of an object in an image. It is calculated as the ratio of the area of intersection between the predicted and ground truth bounding boxes to the area of their union. IoU is used to assess the accuracy of object localization.

IoU = (Area of Intersection) / (Area of Union) … Eq(7)

B. mAP@50 (mean Average Precision at 50):

mAP@50 is a variant of the average precision (AP) metric, computed by averaging the precision values at different recall levels for multiple classes or categories of objects. In mAP@50, the precision is calculated at a fixed recall threshold of 50%. It measures the model's ability to accurately detect and classify objects while ensuring a balanced trade-off between precision and recall.

C. mAP@50-95 (mean Average Precision at varying IoU thresholds):

mAP@50-95 is an extension of mAP@50 that computes the average precision over a range of IoU thresholds, typically ranging from 0.5 to 0.95 with a certain step size. It measures the model's performance across a spectrum of IoU thresholds, reflecting its ability to accurately localize objects under varying levels of overlap between predicted and ground truth bounding boxes.

These metrics are widely used in benchmarking and comparing the performance of object detection algorithms and models, helping researchers and practitioners assess their effectiveness in real-world scenarios.

2.4.4 YOLO Models

In 2015, the first-ever "You Only Look Once" [37] (YOLO) paper was published, introducing a unified real-time object detection model with fast processing speed that predicts bounding boxes from an image in a single pass. The authors claimed that it achieved a 57.9% mean Average Precision (mAP) at 40 frames per second (fps), which at the time outperformed Deformable Part Models (DPM) and Region based Convolutional Neural Networks (R-CNN) in speed. However, YOLOv1 had limitations in detecting small objects.

The following year, YOLOv2, also known as YOLO9000 [39], was introduced. It was trained on 9000 categories, bridging the gap between detection and classification, and improved both speed and accuracy, achieving 78.6% mAP at 40 fps on standard datasets. In 2018, J. Redmon et al. implemented YOLOv3 [40], which was more accurate, achieving 57.9% mAP with a processing time of 51ms on a Titan X GPU.

YOLOv4 [41], released in 2020, achieved state-of-the-art results on the MS COCO dataset, being faster than other detectors with a 43.5% mAP at 65 fps. In the same year, YOLOv5 [42] was introduced, transitioning to the PyTorch framework from the previous DarkNet framework. YOLOv5 included improved segmentation capabilities, enabling it to detect, classify, and segment images, outperforming other segmentation models and CNNs in terms of time complexity.

Two years later, in 2022, both YOLOv6 [43] and YOLOv7 [44] were released. YOLOv6 achieved notable results with its N variant scoring 35.9% mAP at 1235 fps and its S variant achieving 43.5% mAP at 495 fps. However, one drawback of this version was its high power consumption, making it less suitable for low-powered GPUs. YOLOv7, on the other hand, introduced a new architecture for object detection and a novel scaling method. YOLOv7 E6 surpassed Mask R-CNN by 509% in speed, while the X variant achieved 57.8% mAP at 114 fps on a GPU.

In 2023, YOLOv8 [45] was released, accomplishing unprecedented results with a 99.1% mAP50 and 83.5% mAP, and an average processing speed of 50 fps on flying objects, see Table 1 for a brief results summary, the Figure 16 showcases the benchmark comparison between yolo versions .

YOLO Version	Year	mAP	FPS	Key Features
YOLOv1 [37]	2015	57.9%	40	Real-time focus, fast processing
YOLOv2 (YOLO9000) [39]	2016	78.6%	40	9000 categories, improved speed/accuracy
YOLOv3 [40]	2018	57.9%	51ms	Increased accuracy
YOLOv4 [41]	2020	43.5%	65	State-of-the-art on MS COCO
YOLOv5 [42]	2020	N/A	N/A	Segmentation capability (detect, classify, segment)
YOLOv6 [43]	2022	N (35.9) S (43.5)	N (1235) S (495)	speed-accuracy trade-off
YOLOv7 [44]	2022	57.8%	X (114)	New architecture & scaling method

YOLOv8 [45]	2023	99.1% mAP50 /83.5%	50	Groundbreaking results (exceptional

Table 1. Comparison of YOLO versions

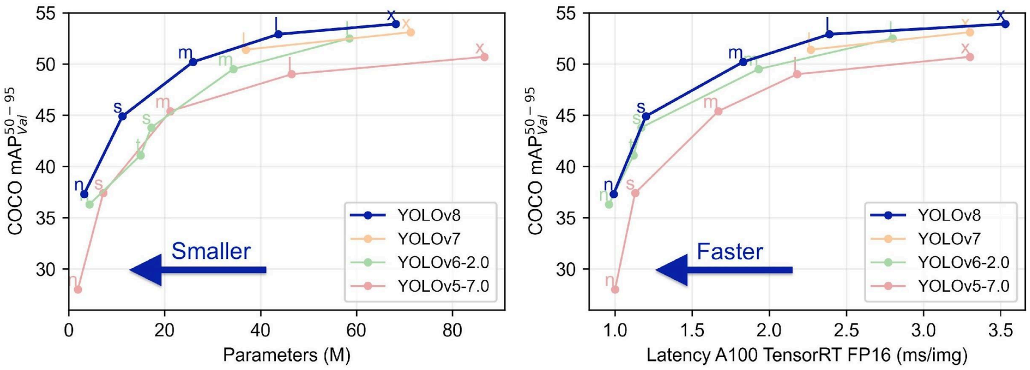


Figure 16. Comparing YOLO versions 8, 7, 6 and 5 both in parameters numbers and latency [119]

2.5 Emergence of AI in AVI

The integration of Artificial Intelligence (AI) into Automated Visual Inspection (AVI) systems is a natural progression in the evolution of visual inspection technologies. One of the key benefits of AVI is the ability to collect vast amounts of data, and modern AI is fundamentally data-driven, learning and improving from the data it processes.[4]

Today, companies collect digital images and videos of machinery, manufactured products, and other aspects of physical operations to conduct visual inspections. These inspections, utilizing video footage and imagery, can be performed in real-time. [4]

The incorporation of AI and machine vision into manufacturing processes falls under the umbrella of Industrial Artificial Intelligence. This involves the application of AI technologies to optimize various aspects of industrial operations, including production, quality control, supply chain management, and predictive maintenance. In the context of quality control, AI enables manufacturers to achieve consistent and reliable product quality while minimizing human error and variability. [16]

2.5.1 How It Works

The functioning of AI-powered AVI follows a similar process to that of conventional AVI, with one notable difference, see Figure 17 below : the last three stages, segmentation, feature extraction, and classification, are consolidated into a single stage referred to as model inference. This stage can be likened to a black box that takes a preprocessed image as input and produces a decision as output. The intricacies of this process will be explored further in the next chapter.

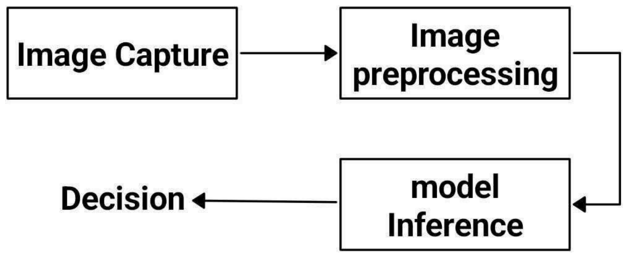


Figure 17. AI model Black box [primary]

2.5.2. Advantages of AI-Based Visual Inspection

- Enhanced Precision
- Consistent Performance
- High-Speed Analysis
- Cost-Efficiency
- Risk Mitigation
- Complex Pattern Recognition
- Data-Driven Insights
- Reduced Error Rates
- Scalability [17]

2.5.3 YOLO Variants for Industrial Defect Detection

A. Challenges of Industrial Defect Detection:

- Stringent Inspection Requirements: Defects can be minute, requiring specialized techniques like spectral imaging for identification.
- Time Constraints: Production line operation demands rapid defect detection for efficient processing.

B. YOLO Variants as a Solution:

YOLO (You Only Look Once) is an undisputed leader in the field of object detection, renowned for its exceptional accuracy and speed.

- Real-Time Capability: Existing research demonstrates YOLO's effectiveness in real-time industrial applications.
- Customization for Specific Needs: YOLO variants allow users to modify internal modules to address specific industrial needs (e.g., attention mechanisms for defect emphasis while maintaining real-time performance.
- Flexible Architecture Selection: Sub-variants within YOLO architectures (e.g., YOLOv5-S/M/L) offer varying computational loads based on parameter count. This enables researchers to choose between lightweight versions (e.g., YOLOv5-small) prioritizing real-time inference over maximum average precision (mAP) or larger versions (e.g., YOLOv5-large) for potentially higher accuracy.[46]

2.6 YOLO in Industrial Inspection, Literature Review

Defect detection in various industrial applications has seen significant advancements with the use of deep learning models, particularly the YOLO (You Only Look Once) family of models. For instance, X. Yin (2020) [63] developed a real-time automated defect detection system for sewer pipes using YOLOv3, achieving an mAP of 85.3% and an F-score over 80%, indicating robust performance in detecting and classifying sewer pipe defects . Similarly, W. Wu et al. (2020) [64] improved YOLOv3 to detect electrical connector defects, achieving a remarkable accuracy rate of 93.5%, with a significantly lower false detection rate of 3.9% and a missed detection rate of 2.6% compared to the original YOLOv3 .

In the realm of pipeline inspection, Z. Hu et al. (2022) [65] designed a pipeline inspection robot system using YOLOv3, demonstrating the feasibility and effectiveness of this automated approach over traditional manual methods . H. Chen et al. (2019) [66] applied YOLOv3 for accurate and real-time detection of electrical components in UAV inspection images, achieving 96.4% accuracy and a 93.6% mAP, showcasing the model's capability in high-stakes inspection tasks .

Further advancements were made by X. Chen et al. (2022) [67], who improved YOLOv3 for real-time detection of surface defects in industrial products, achieving an mAP of 86.96% at an impressive 80.96 FPS . For railroad track component inspection, F. Guo et al. (2021) [68] proposed a YOLOv4-hybrid model, achieving high accuracy (94.4% mAP) and fast processing speed (78.7 FPS) . In agricultural applications, S. Fan et al. (2022) [69] developed a YOLOv4-based method for real-time detection of apple defects using NIR images, achieving a 93.9% accuracy and a 93.74% mAP .

Aircraft maintenance also benefited from YOLO advancements, as Z.-H. Chen et al. (2021) [70] utilized YOLOv4 for defect classification, achieving 72% accuracy, thereby enhancing non-destructive testing (NDT) processes . For solar cell defect detection, M. Zhang et al. (2022) [71] improved YOLOv5, achieving 89.64% accuracy and 36.24 FPS, demonstrating the model's efficiency and effectiveness . R. Jin et al. (2021) [72] enhanced YOLOv5 for fabric defect detection using a teacher-student architecture, achieving an impressive 0.981 AUC and 0.447 mAP .

In petrochemical pipeline inspection, K. Chen et al. (2022) [73] combined Cycle-GAN with YOLOv5, achieving 93.10% precision and 90.96% recall, thereby significantly improving detection accuracy . Z. Li et al. (2022) [74] proposed a two-stage detection framework utilizing YOLOv5 and Inception-ResNetV2, achieving 83.3% mAP on the Enriched-NEU-DET dataset and 91.0% in industrial settings . L. Wang et al. (2023) [75] enhanced YOLOv5 for real-time steel defect detection, achieving a 72% mAP, focusing on various defect scales .

H. F. Le et al. (2022) [76] further advanced YOLOv5 for industrial part defect detection, incorporating BiFPN and Transformer detectors, achieving a 91.6% recall and 95 FPS . J. Xu et al. (2022) [77] improved YOLOv5 for chip pad detection using an OCAM layer, achieving 82.8 mAP on the VOC2007 dataset . Y. Duan et al. (2023) [78] proposed the YOLOv5-CCFE model for railway tunnel equipment recognition, achieving an outstanding 98.6% mAP@0.5 and 68.9% mAP@0.5:0.95 .

In aerospace applications, X. Li et al. (2022) [79] developed an improved YOLOv5s-KEB model for aero-engine surface defect detection, achieving 98.3% mAP with a 2.6ms inference time . J. Zheng et al. (2022) [81] enhanced YOLOv7 for insulator defect detection, achieving 93.8% mAP and 94.9% detection accuracy with the SIoU loss function . Y. Wang et al. (2022) [82] improved YOLOv7 for steel defect detection, incorporating BiFPN and ECA attention mechanisms to achieve 80.2% mAP on the GC10-DET dataset and 81.9% on the NEU-DET dataset .

J. Chen et al. (2023) [83] utilized YOLOv7 for automotive running lights defect detection, achieving 89.7% accuracy with a processing speed of 61 FPS . B. Chen et al. (2023) [84] proposed a YOLOv7-based method for PCB defect detection, achieving 97.5% mAP@0.5 and 54.7% mAP@0.5:0.95 with an enhanced detection speed of 83.3 FPS . H. Huang et al. (2024) [85] developed a YOLOv7-based model for automobile parts defect detection, achieving 75.5% mAP and 76 FPS, outperforming several concurrent models .

In semiconductor defect detection, E. Dehaerne et al. (2023) [86] optimized YOLOv7, achieving an 86% mAP, highlighting the model's superior performance over RetinaNet . Y. Zhao et al. (2024) [87] improved YOLOv7 for oil and gas pipeline inspection, achieving a 90% mAP with enhanced feature extraction and network convergence . E. Bellou et al. (2024) [88] proposed YOLOv8 for real-time power line component detection, achieving an 83.8% accuracy and a 99 F1-score for defective insulators .

M. Liu et al. (2024) [89] enhanced YOLOv8 for bearing defect detection, achieving 86.5% mAP with innovative technologies such as the VanillaNet backbone and Shape-IoU loss . Q. Ling et al. (2023) [90] developed a YOLOv8-based model for dense PCB component detection, achieving 87.7% mAP and 110 FPS, outpacing state-of-the-art models . J. Silva et al. (2024) [91] utilized YOLOv8 for quality control in smart glasses, achieving a 91.6% mAP, demonstrating its application in merging real-world and digital environments .

Finally, M. H. Zubayer et al. (2024) [92] proposed a YOLOv8-based method for jet engine blade defect detection, achieving an impressive 99.5% mAP and 58% mAP50-95 on a custom dataset, further exemplifying the versatility and efficiency of YOLO models in defect detection across various industries. Table 2 outlines the key points of the literature review about YOLO models, Figure 18 highlights the number of studied papers and average accuracy per version.

Reference	Domain of Application	Technique	Accuracy
[63]	Sewer pipes	YOLOv3	mAP: 85.3%, F-score: >80%
[64]	Electrical connector defects	Improved YOLOv3	Accuracy: 93.5%
[65]	Pipeline inspection	YOLOv3	Not specified
[66]	Electrical components in UAV inspection	YOLOv3	Accuracy: 96.4%, mAP: 93.6%
[67]	Surface defects in industrial products	Improved YOLOv3	mAP: 86.96%
[68]	Railroad track components	YOLOv4-hybrid	mAP: 94.4%
[69]	Apple defects	YOLOv4	Accuracy: 93.9%, mAP: 93.74%
[70]	Aircraft maintenance NDT	YOLOv4	Accuracy: 72%
[71]	Solar cell defects	Improved YOLOv5	Accuracy: 89.64%
[72]	Fabric defects	Enhanced YOLOv5	AUC: 0.981, mAP: 0.447
[73]	Petrochemical pipeline inspection	Cycle-GAN and YOLOv5	Precision: 93.10%, Recall: 90.96%
[74]	Industrial defect detection	YOLOv5 and Inception-ResNetV2	mAP: 83.3% (Enriched-NEU-DET), mAP: 91.0% (industrial settings)
[75]	Steel defect detection	Improved YOLOv5	mAP: 72%
[76]	Industrial part defect detection	YOLOv5 with BiFPN and Transformer detectors	Recall: 91.6%
[77]	Chip pad detection	YOLOv5 with OCAM layer	mAP: 82.8
[78]	Railway tunnel equipment recognition	YOLOv5-CCFE	mAP@0.5: 98.6%, mAP@0.5:0.95: 68.9%
[79]	Aero-engine surface defects	YOLOv5s-KEB	mAP: 98.3%
[81]	Insulator defects	YOLOv7	mAP: 93.8%, Accuracy: 94.9%

Reference	Domain of Application	Technique	Accuracy
[82]	Steel defect detection	YOLOv7	mAP: 80.2% (GC10-DET), mAP: 81.9% (NEU-DET)
[83]	Automotive running lights defect detection	YOLOv7	Accuracy: 89.7%
[84]	PCB defect detection	YOLOv7	mAP@0.5: 97.5%, mAP@0.5:0.95: 54.7%
[85]	Automobile parts defect detection	YOLOv7	mAP: 75.5%
[86]	Semiconductor defect detection	Optimized YOLOv7	mAP: 86%
[87]	Oil and gas pipeline inspection	Improved YOLOv7	mAP: 90%
[88]	Power line component detection	YOLOv8	Accuracy: 83.8%, F1-score: 99%
[89]	Bearing defect detection	YOLOv8	mAP: 86.5%
[90]	Dense PCB component detection	YOLOv8	mAP: 87.7%
[91]	Quality control in smart glasses	YOLOv8	mAP: 91.6%
[92]	Jet engine blade defect detection	YOLOv8	mAP: 99.5%, mAP50-95: 58%

Table 2. Table summarizing YOLO versions used in the discussed Papers

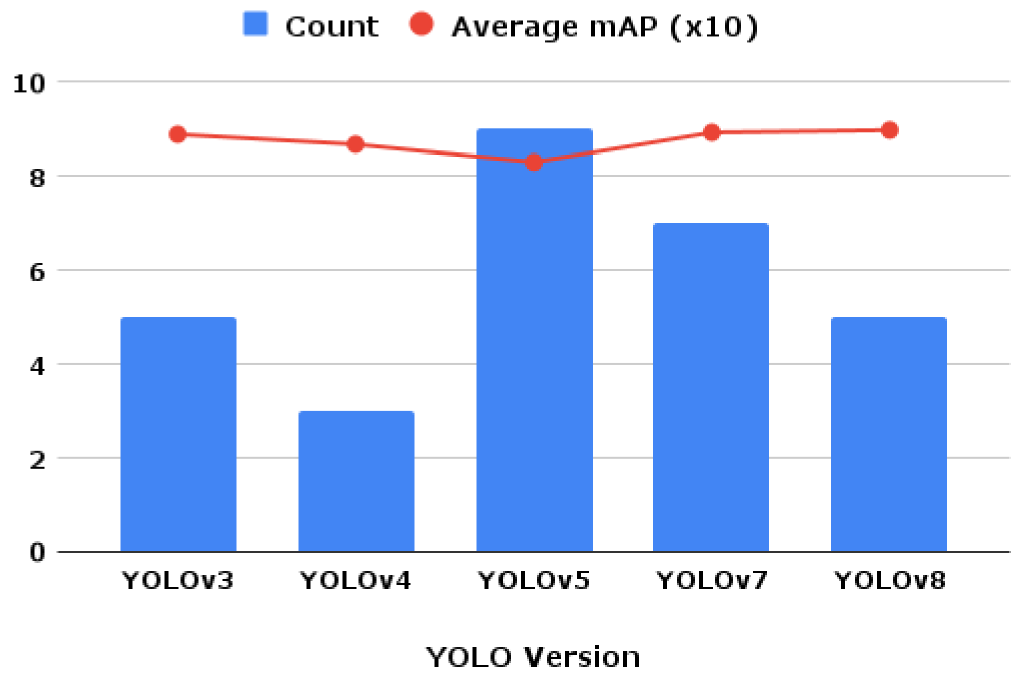


Figure 18. Number of papers showcasing the use of each version of YOLO in industrial inspection [primary]

2.7 Real-World Applications

The industrial sector is at the forefront of adopting cutting-edge technologies to enhance efficiency, reduce costs, and ensure quality. Among these technologies, Artificial Intelligence (AI) has emerged as a pivotal tool, revolutionizing various facets of industrial operations. This chapter delves into the practical implementation and real-world applications of AI in industrial inspection, highlighting its transformative impact on manufacturing processes, quality control, and maintenance strategies.

2.7.1 Real world applications

A. Pharmaceutical: Stevanato Group [93]

Stevanato Group exemplifies the advanced use of AVI systems in pharmaceutical manufacturing. The company develops inspection solutions for products such as ampoules, vials, syringes, and bottles, utilizing high-performance CMOS cameras and advanced illumination techniques.

Their deep learning algorithms minimize false rejection rates, ensuring high accuracy. For example, Stevanato Group has developed AVI machines to inspect ampoules with sodium chloride and vials containing freeze-dried anti-rabies vaccines. Additionally, for a Central American client, they installed a system using trajectory algorithms and line scan cameras to inspect turbid suspensions in vials with children's vaccines.

B. Aerospatial: AutoInspect by 3D.aero [93]

AutoInspect, developed by 3D.aero, is a robot-based optical system that employs an AI-powered inspection mechanism and industrial robotics for intelligent defect classification see Figure 19. This system not only enhances accuracy and reliability but also substantially reduces inspection time. According to 3D.aero, a full aero-engine combustion chamber can be examined in less than four hours, representing an 80% reduction compared to the manual inspection process.

Figure 19. 3D.Aero's AutoInspect aerospatial visual inspection system [93]

C. Automotive: Volvo and UVeye [93]

Since 2020, Volvo Cars has employed the Atlas quality inspection system developed by UVeye, which leverages computer vision technology. This system utilizes over 20 computer vision-powered cameras installed in an aluminum tunnel at the end of the assembly line, see Figure 20. Each camera captures hundreds of images per second, enabling the AI algorithm to meticulously assess the surface quality of each vehicle. This system outperforms traditional manual inspection methods, detecting 10% to 40% more defects, including scratches, dents, and component alignment anomalies. Remarkably, it can identify defects as small as 0.2 millimeters in size.

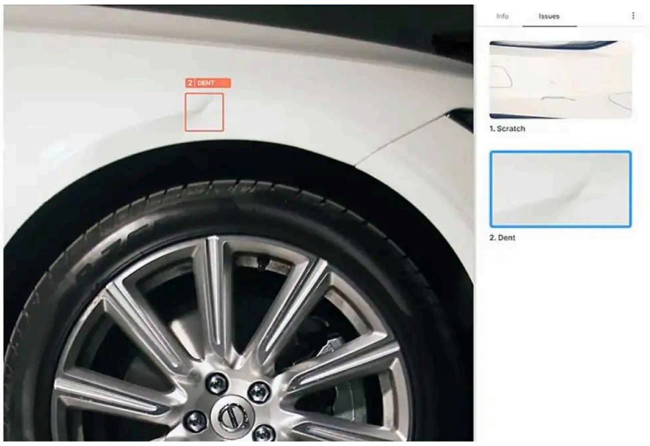


Figure 20. Volvo's cars visual inspection system Atlas [120]

D. Medical Devices: Dovideq Medical Systems [93]

Dovideq Medical Systems developed LightControl, an automated system for inspecting rigid endoscopes. This system utilizes IDS cameras and neural networks to measure key parameters, ensuring flawless endoscope conditions during production and maintenance. Such AVI advancements uphold stringent quality standards in medical device manufacturing, enhancing patient safety and diagnostic accuracy.

E. Electronics: Maddox AI Semi-Automatic PCB Inspection System [94[

Maddox AI has introduced a semi-automatic inspection system designed for printed circuit board (PCB) inspection, leveraging a high-resolution camera. Given the significant variability in product specifications, achieving full automation in inspection proved unfeasible. Nevertheless, the system adeptly detects defects like improperly mounted components, scratches, and solder bumps.

F. Food and Beverage: Detecting Defects on Reusable Glass Bottles [95]

Solomon-3d's SolVision, with minimal sample images, can be trained to distinguish various types of stains and mildew. Leveraging a segmentation tool, the software learns the characteristics and spatial distribution of defects, facilitating automatic real-time identification along the cleaning production line. This comprehensive inspection of glass bottles ensures precise detection of areas requiring disinfection, thereby bolstering the safety and efficiency of the recycling process.

2.7.2 Commercial Solution

A. IBM Maximo [96]

IBM Maximo Visual Inspection empowers quality control and inspection teams by integrating computer vision AI capabilities into their workflows. This user-friendly toolset simplifies the utilization of computer vision, deep learning, and automation for technicians. Through intuitive features for labeling, training, and deploying artificial intelligence vision models, IBM Maximo Visual Inspection facilitates quick and easy deployment. Whether through training via a drag-and-drop visual user interface or importing custom models, activation on mobile and edge devices is seamless. With self-learning machine algorithms, users can customize detect-and-correct solutions, improving efficiency and accuracy in quality control processes.

B. Landing.ai's LandingLens [97]

LandingLens, developed by landing.ai which was founded by Andrew NG, simplifies visual inspection tasks with its intuitive approach to labeling, training, and deploying AI-based computer vision models. Its user-friendly interface requires no prior AI knowledge, enabling rapid model development and deployment across diverse industries such as automotive, electrical, medical, and agricultural sectors. With LandingLens, users can build sophisticated models within minutes and scale projects seamlessly from single production lines to global operations. The platform's standardized workflow streamlines project management and collaboration, making it a valuable tool for businesses seeking efficient and scalable solutions for their computer vision needs.

C. Maddox AI [98]

Maddox AI offers a unique fusion of traditional vision systems' reliability with the capability to accurately inspect intricate components and surfaces. This innovative approach ensures consistent inspection results even for components varying in color, glossiness, or texture across production batches. Maddox AI's visual inspection solution caters to diverse industries, ranging from automotive to food and beverage, providing comprehensive quality control across various manufacturing sectors.

D. Google Cloud Visual Inspection [99]

Google Cloud Visual Inspection is a cutting-edge solution that leverages advanced machine learning and computer vision technologies to enhance quality control processes. This platform offers a comprehensive suite of tools for visual inspection tasks, enabling businesses to efficiently detect defects, anomalies, and irregularities in their products and manufacturing processes. With its scalable infrastructure and robust machine learning capabilities, Google Cloud Visual Inspection empowers organizations to streamline their quality control workflows, improve product quality, and enhance operational efficiency across a wide range of industries.

2.8 Conclusion

In conclusion, the chapter examined the transformative role of AI in industrial inspection, particularly through the use of Machine Learning (ML) and Deep Learning (DL) techniques such as Convolutional Neural Networks (CNNs). These methods have demonstrated superior performance in image analysis and defect detection compared to traditional methods. The review highlighted the emergence of YOLO models, specifically YOLOv8, which excel in both accuracy and processing speed, making them ideal for real-time applications on high-speed production lines. Overall, the integration of AI in industrial inspection signifies a paradigm shift in quality control across various sectors, offering unprecedented accuracy, efficiency, and scalability. This not only enhances manufacturing processes but also contributes to significant cost savings, reduced downtime, and improved regulatory compliance, with the promise of ongoing advancements in quality assurance and operational excellence

Chapter 3: AI based Automated Visual Inspection System Implementation

3.1 Introduction

As discussed in previous chapters, You Only Look Once (YOLO) models have proven highly effective for real-time defect detection in industrial inspection systems. Building upon this knowledge, this chapter delves into an internship project undertaken at DSRU in collaboration with Toudja Factory. The project's objective was to implement a computer vision-based industrial inspection system utilizing YOLO models. This end-to-end system encompassed the entire process, from data collection of soda bottle images (full, empty, labeled, and unlabeled) to deployment on a Jetson Nano and seamless integration into the production line. The model's core function was to identify these bottle characteristics for real-time inspection, allowing for the rejection of empty or unlabeled bottles before packaging. This early-stage detection offers significant benefits to factories, including reduced material waste and enhanced customer satisfaction by ensuring only properly filled and labeled products reach consumers.

3.2 Development Environment

This section details the hardware and software resources utilized throughout the project.

3.2.1 Hardware

- Personal Computer (PC):
    - Processor: Intel Core i5 (6th Generation)
    - Memory (RAM): 8 GB
- Kaggle Notebooks Platform:
    - Graphics Processing Unit (GPU): NVIDIA P100 (16 GB)
    - Memory (RAM): 30 GB
    - Storage: 20 GB
- Jetson Nano Development Kit:
    - Graphics Processing Unit (GPU): NVIDIA Tegra (4 GB)
    - System on a Chip (SoC): NVIDIA Tegra X1+ (J1020)
- Camera: Arducam Industrial Gray Level Camera (up to 120 fps)

3.2.2 Software

- Personal Computer (PC):
  - Operating System: Ubuntu 20.04 LTS
  - Integrated Development Environment (IDE): Visual Studio Code
  - Programming Language: Python (version 3.8)
- Kaggle Notebooks Platform:
  - Operating System: Debian-based
  - Programming Language: Python (version 3.10)
- Jetson Nano Development Kit:
  - Operating System: Ubuntu 20.04 LTS
  - Integrated Development Environment (IDE): Visual Studio Code
  - Programming Language: Python (version 3.8)
  - NVIDIA JetPack (version 4.6)
- Applications, Frameworks, and Libraries:
  - Annotation: Roboflow
  - Training and Experiment Tracking: MLflow
  - Image Processing: OpenCV
  - Object Detection Models: YOLOv5 and YOLOv8 (from Ultralytics)
  - Deep Learning Framework: PyTorch
  - Other Libraries: NumPy, Matplotlib, GPIO, ONNX, TensorRT

3.2.3 Hardware Utilization

The personal computer served as the primary development platform for coding, solution development, and basic tasks like image processing and annotation.

Kaggle notebooks provided a GPU-powered environment for model training and experiment tracking, leveraging the platform's NVIDIA T4 GPU to accelerate computationally intensive tasks unfeasible on the local PC due to its lack of a dedicated GPU.

The Jetson Nano functioned as the final deployment environment for the model. Code and model underwent necessary adaptations for deployment, including building a TensorRT engine for optimized inference performance.

3.2.4 Testing Environment Setup

Following the establishment of the development environment, the camera was strategically positioned to capture optimal bottle images. The Jetson Nano interacted with a "boxer" or “actuator” (presumably a system for bottle ejection) upon model detection of empty or unlabeled bottles.

3.3 Contributions

This section details my contributions throughout the iterative process of data collection, model training, validation, and deployment undertaken during the project. This work was time-consuming and required significant effort, particularly in implementing a complete machine learning workflow pipeline independently.

3.3.1 Data Collection and Preparation:

Data acquisition involved capturing images and videos from various sources:

- Production Line: Utilizing smartphones and an industrial camera to collect data directly from the production line.

- Supermarket Shelves: Capturing images of bottled beverages at "Superette Bekkouche" to represent real-world variations.

Following data collection, a meticulous selection process identified images showcasing optimal bottle features:

- Clear label visibility
- Cap presence
- Liquid level representation

Empty bottles were obtained for inclusion in the dataset.

The dataset development process involved creating several variations:

- Binary Classification: Categorization into "empty" and "full" classes.
- Multi-Class Classification: Classification of distinct bottle components like bottle, liquid, label, and cap.
- Mixed Dataset: Combining full and empty bottles with individual components for comprehensive representation.

Each dataset variation further comprised sub-versions incorporating data augmentation techniques:

- Grayscale conversion
- Blurring
- Translation (image shifting)
- Brightness adjustments

This approach yielded datasets ranging from 100 images (without augmentation) to over 900 labeled images in augmented versions. However, the focus shifted towards quality over quantity in selecting the most suitable data for training.

The final stage involved data partitioning into training (70%), validation (20%), and testing (10%) sets to facilitate model evaluation.

A. Version 0: Baseline Dataset

The initial dataset, referred to as the baseline dataset, was pre-existing at the start of the project. This version consists of 73 RGB square images, each with a width of 320 pixels, captured in various outdoor and random environments. The dataset includes two classes: "empty" and "full," which describe the state of a bottle. The images were divided into training and validation sets with ratios of 93% and 7%, respectively. There are 256 annotations in total, with an average of 3.6 annotations per image across the two classes. The Figure 21 below illustrates the class balance between the sets. The images exclusively feature small bottles, as depicted in the accompanying Figure 22

Figure 21. Dataset V0 class balance

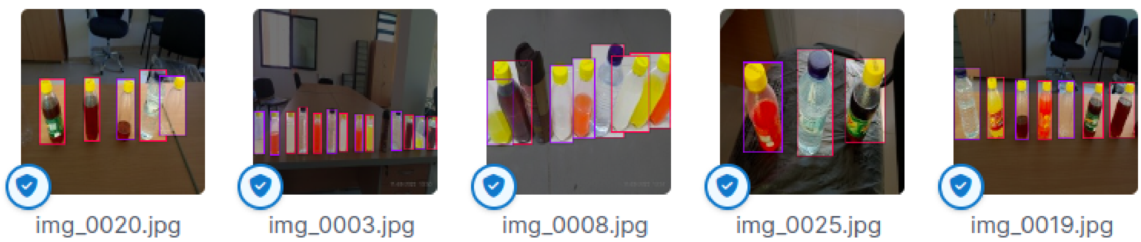


Figure 22. Dataset V0 sample

B. Version 1: Enhanced Dataset

This version is an enhancement of the baseline dataset, incorporating additional images of large bottles captured from the production line and at home using a smartphone. Additionally, some grayscale images were introduced. The resulting dataset consists of 384 images with over 1,200 annotations, averaging 3.2 annotations per image across the two classes: "empty" and "full." The image size remains 320 pixels squared. The dataset is divided into training, validation, and test sets with proportions of 70%, 20%, and 10%, respectively. The class balance is illustrated in the Figure 23 below.

Figure 23. Dataset V1 class balance

We applied several augmentations to the training set, resulting in an increased number of images. The augmentations applied include:

- Rotation: Between -15° and +15°
- Brightness: Between -24% and +24%
- Exposure: Between -15% and +15%
- Blur: Up to 0.6 pixels

The Figure 24 below shows some samples of the images.

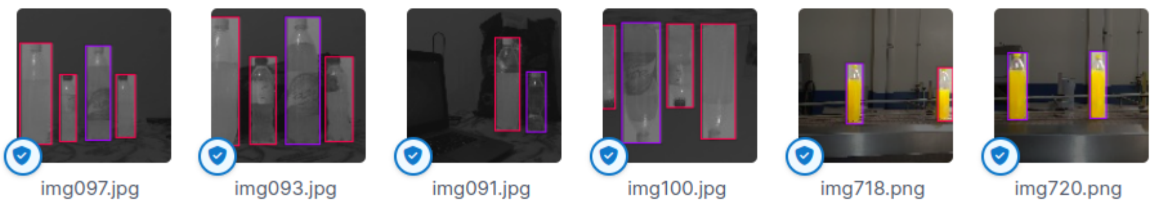


Figure 24. Dataset V1 sample

C. Version 2: New Classes

Given the subjectivity involved in annotating whether a bottle is empty or full, we decided to introduce a new set of classes: bottle, cap, date, label, and liquid. This dataset facilitates a multi-task model that can simultaneously detect the presence of the cap and label while helping to calculate the fill level of the bottle using the liquid class.

The dataset contains 172 images with over 2,000 annotations, averaging 12 annotations per image across the five classes. Several sub-versions were created by varying the image size (either 320 pixels or 640 pixels) and by applying or not applying augmentations. The dataset was split into training, validation, and test sets using the usual ratios of 70%, 20%, and 10%, respectively. The class balance is illustrated in the Figure 25 below. The images were captured using a smartphone, an RGB webcam, and a grayscale industrial camera.

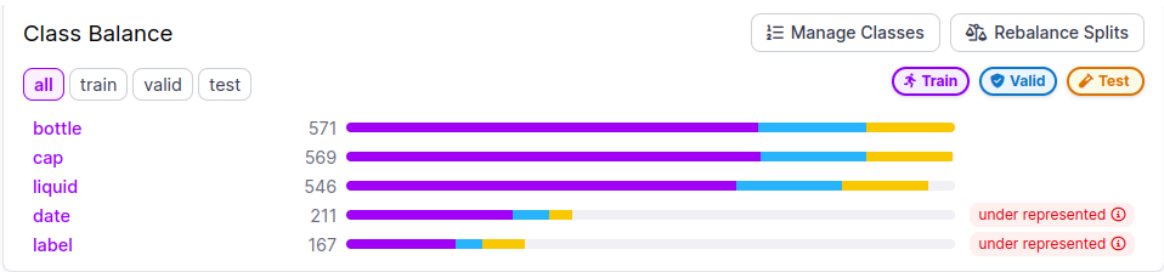


Figure 25. Dataset V2 class balance

The augmentations applied include:

- Rotation: Between -15° and +15°
- Brightness: Between -24% and +24%
- Exposure: Between -15% and +15%
- Blur: Up to 0.6 pixels

By varying image sizes and augmentation techniques, we developed the following dataset V2 variants:

- 320_sm, 320_md, 320_lg
- 640_sm, 640_md, 640_lg

In this context:

- "sm" (small) indicates that no augmentation was applied.
- "md" (medium) indicates that some augmentation was applied.
- "lg" (large) indicates that all the aforementioned augmentation techniques were applied.

Figure 26 below illustrates some samples of the dataset.

Figure 26. DataSet V2 Samples.

D. Version 3: Last Version

- 100 images
- Mostly grayscale: Captured directly from the production line
- Resolution: 320 and 640 pixels wide
- Custom resizing function
- Augmentations: Brightness and exposure adjustments

These are just some key features of the last dataset version, for more information, please refer to section 3.4.1.

3.3.2 Model Training and Validation

This project employed two YOLO model versions: YOLOv5 and YOLOv8. Prioritizing consistency and robustness, the "nano" and "small" model variants were chosen. This selection aimed to balance inference speed (favored by smaller models) with accuracy. The most commonly used configurations were 320x320 input image size and the "nano" variant for both YOLO versions.

The iterative training and validation process encompassed the following steps:

1. Training the chosen model variant (small/nano) with a specific input size (320/640 pixels)

2. Model validation using the validation set, analyzing metrics like loss and inference time.
3. Deployment of the trained model to the Jetson Nano for real-world testing.
4. Implementation of optimization techniques:
   - Exporting the model to ONNX format for broader compatibility.
   - Converting the model to a TensorRT engine for enhanced inference speed on NVIDIA GPUs.
   - Quantization and pruning (techniques to reduce model size and improve efficiency) when necessary.
5. Evaluation of speed and accuracy after optimization.
6. Iteration restarts if performance fails to meet expectations. This might involve:
   - Trying a different model variant or input size.
   - Rebuilding the dataset to address identified issues.

Building an AI project inherently involves iterative refinement. Each iteration strives to enhance model performance by:

- Identifying and rectifying sources of errors.
- Formulating hypotheses to address performance shortcomings.

The primary challenges encountered during each iteration were accuracy and speed:

- Accuracy: Mitigated by:
  - Utilizing a more representative real-world dataset.
  - Employing larger model variants (small/medium) and larger input sizes (640 pixels).
- Speed: Addressed by:
  - Deploying the model in TensorRT engine format.
  - Quantization and pruning techniques.
  - Selecting smaller model variants (nano) and smaller input sizes (320 pixels) to reduce computational demands.

It is important to acknowledge that these solutions can sometimes be contradictory. Careful consideration is required to achieve an optimal balance between accuracy and speed in a given scenario.

A. Version 0: Baseline Model

The baseline model utilized a YOLO version 5 nano variant trained with dataset V0. The evaluation results are depicted in the Figure 27 below.

```
val: Scanning /kaggle/working/datasets/labels/val.cache... 15 images, 0 backgrou
                 Class     Images  Instances          P          R      mAP50
                   all         15        105      0.885      0.819      0.901       0.76
                 empty         15         45      0.947      0.689      0.852      0.709
                  full         15         60      0.822       0.95      0.949      0.811
Speed: 0.1ms pre-process, 7.0ms inference, 7.4ms NMS per image at shape (32, 3, 320, 320)
```

Figure 27. Model V0 eval results

Issues identified with this version primarily pertain to inference speed, indicating that the model operates slowly and does not align with the production line's speed.

B. Version 1: Enhanced V0

Version 1 of the model addressed the issues encountered with Model V0 by implementing several techniques:

1. Exporting to ONNX format using the YOLOv5 export module.
2. Attempting int8 Quantization, which was unsuccessful due to a lack of data samples for calibration.
3. Using the sparseML library for pruning, which faced challenges due to hardware issues.
4. Building a TensorRT engine using the YOLOv5-TensorRT library on the Jetson Nano. This approach performed well, achieving 50 frames per second (fps) while maintaining accuracy.

Ultimately, the TensorRT engine successfully addressed the speed and inference time issues. However, during deployment in the production environment, it became evident that the model, trained only on small bottles, did not perform as intended when faced with large bottles (2 liters) on the production line during a visit to Toudja's factory.

C. Version 2: Enhanced DataSet Version Model

To address the issues encountered in previous iterations, we developed a new dataset, V1, specifically including images of big bottles. We trained the same version and variant of YOLO and created a TensorRT engine for efficient inference. However, during deployment, we encountered a problem with the web-camera used, which does not match the required speed and introduces motion blur.

D. Version 3: New Dataset New Model

The solution chosen was to replace the current camera with an Arducam industrial grayscale camera, offering a resolution of 640@120fps and ensuring minimal motion blur. We proceeded to create a new dataset, V2, using images captured with this camera along with additional photos. For this iteration, we selected YOLOv8, encompassing both nano and small variants of the model. Training involved all variations of dataset V2, resulting in 12 experiments, each tracked using MLFlow. The experiment names follow this template: imgsz_dssz_v, where:

- imgsz: denotes the image size (320 or 640 pixels)
- dssz: indicates the dataset size (sm for small, md for medium, or lg for large)
- v: represents the YOLO variant (n for nano, s for small)

The Figures illustrate different aspects of the training progress for all experiments:

- Figure 28: Evolution of mAP50-95 through epochs for all experiments.
- Figure 29: Evolution of mAP50-95 through epochs grouped by dataset variants.
- Figure 30: Evolution of mAP50-95 through epochs grouped by image size.
- Figure 31: Evolution of mAP50-95 through epochs grouped by model variant.

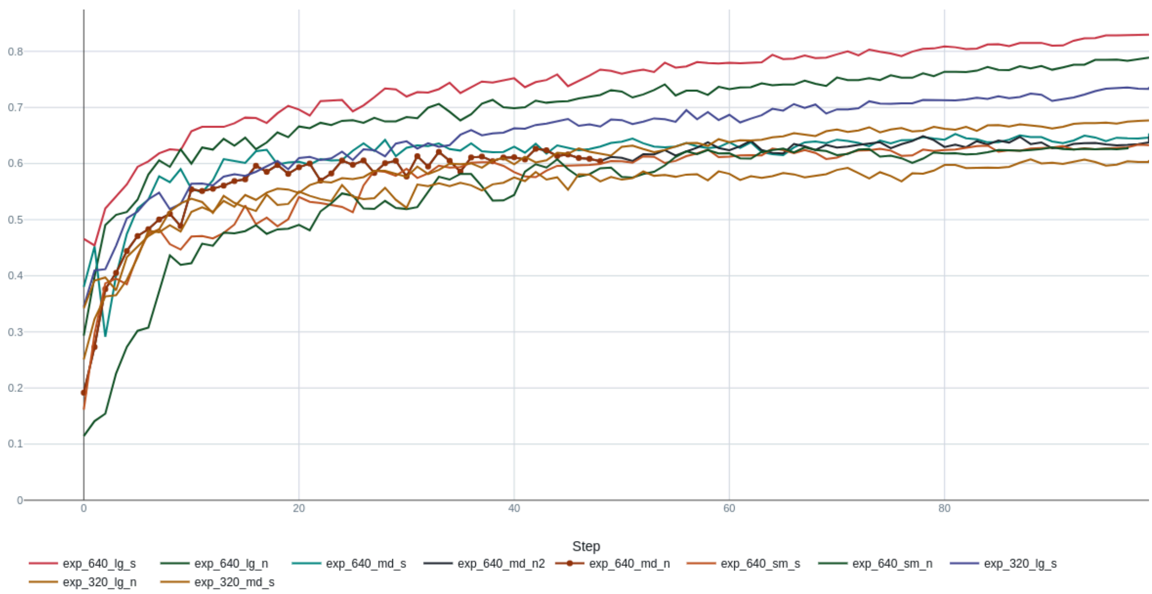


Figure 28. Evolution of mAP50-95 through epochs for all experiments.

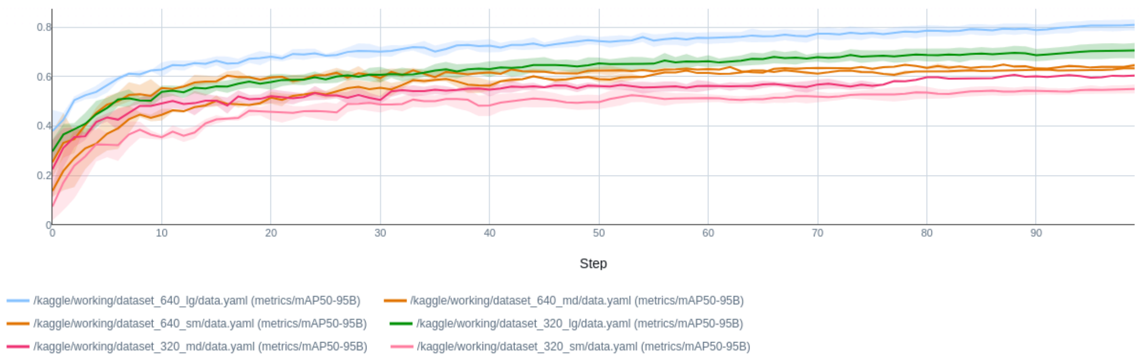


Figure 29. Evolution of mAP50-95 through epochs grouped by dataset variants.

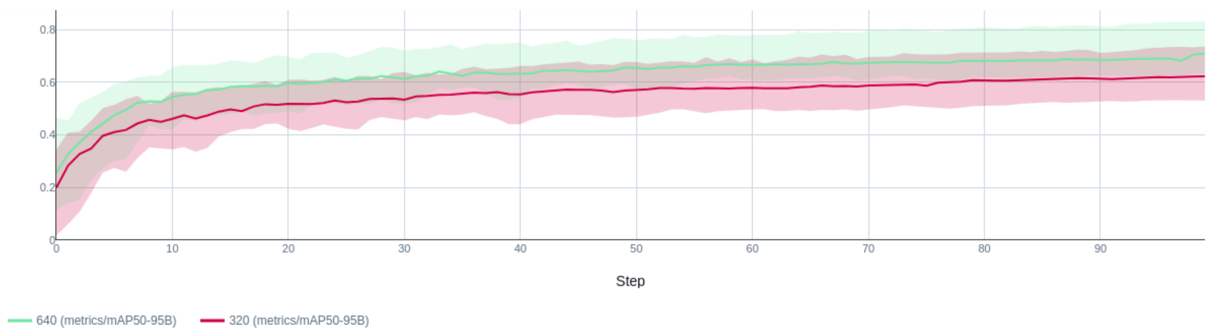


Figure 30. Evolution of mAP50-95 through epochs grouped by image size

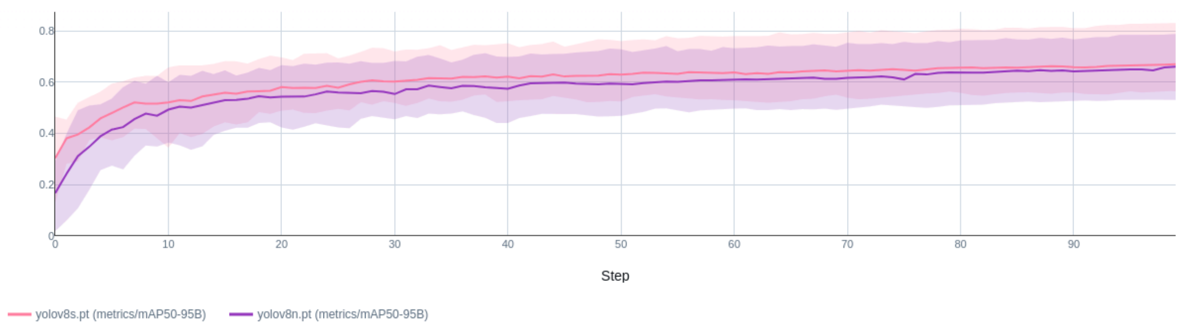


Figure 31. Evolution of mAP50-95 through epochs grouped by model variant.

Based on the analysis of the training phase, it's clear that the model achieves superior performance when trained with a larger dataset containing 640px wide images, especially when using the small YOLO variant. To validate this observation, we conducted a benchmark on the test set, which theoretically represents new cases unseen by the model.

The evaluation scores for model variants, including both speed and mAP50-95 metric, are illustrated in Figure 32 This evaluation was conducted on Kaggle.

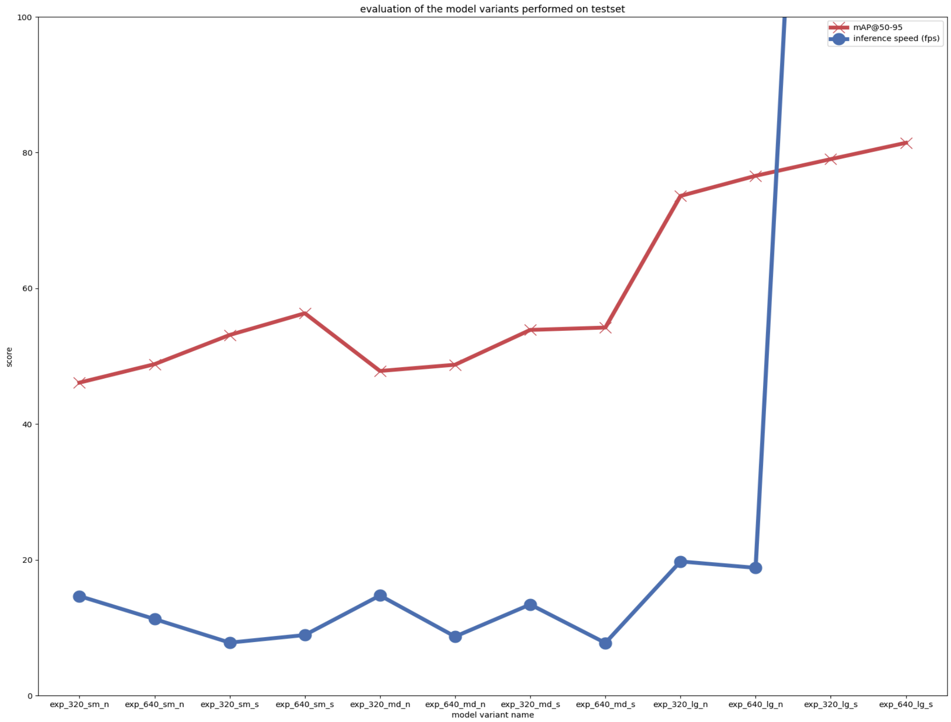


Figure 32. Evaluation of the model variants

It appears that the models did not perform as anticipated during the testing phase, suggesting that certain model variants may have overfitted during training.

E. Version 4: Last but not Least

For the latest iteration, we utilized the final version of the dataset that best represents the deployment environment. This latest version comprises four variants. We trained two variants of YOLOv8 specifically, the nano and small variants. For more details on why this version of YOLO was chosen, please refer to section 3.4.2.

During training, we prioritized online augmentation over offline augmentation to enhance generalization and mitigate the risk of overfitting. Section 3.4.3 provides further information about the fine-tuning process employed.

- Figure 33: Evolution of mAP50-95 through epochs for all experiments.
- Figure 34: Evolution of mAP50-95 through epochs grouped by dataset variants.

- Figure 35: Evolution of mAP50-95 through epochs grouped by image size.
- Figure 36: Evolution of mAP50-95 through epochs grouped by model variant.

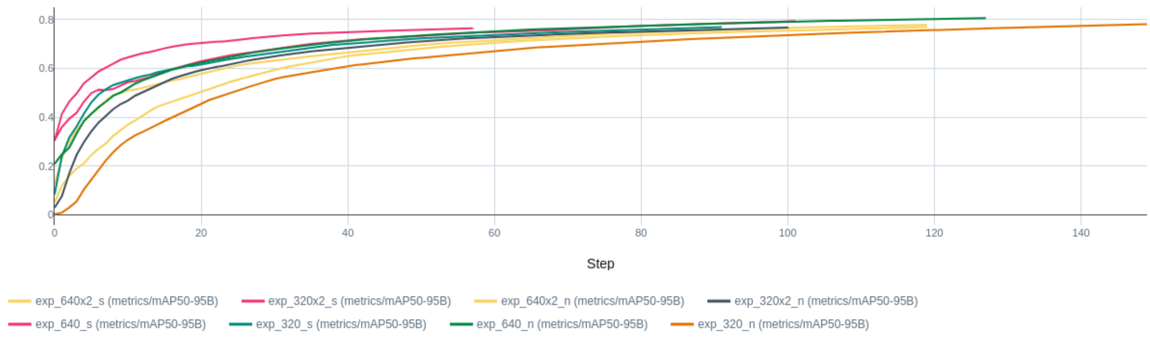


Figure 33. Evolution of mAP50-95 through epochs for all experiments.

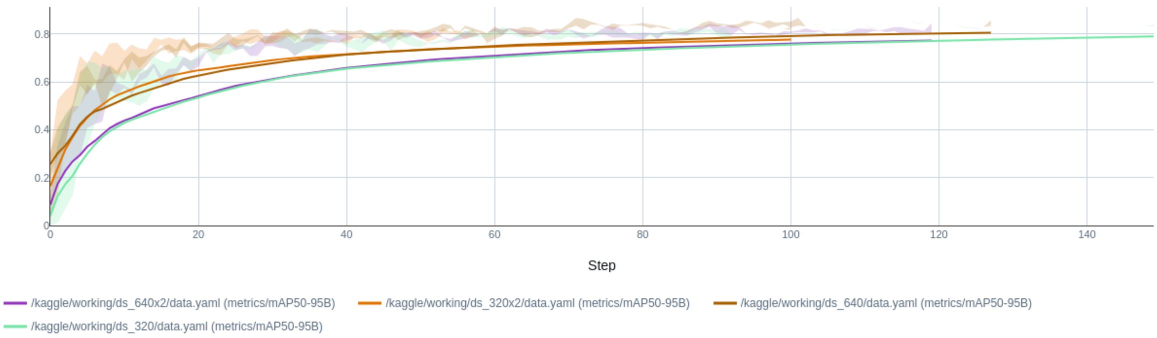


Figure 34. Evolution of mAP50-95 through epochs grouped by dataset variants.

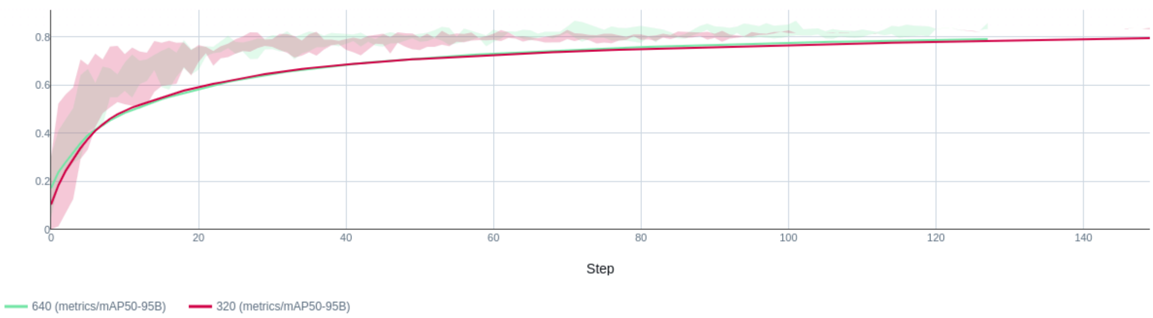


Figure 35. Evolution of mAP50-95 through epochs grouped by image size.

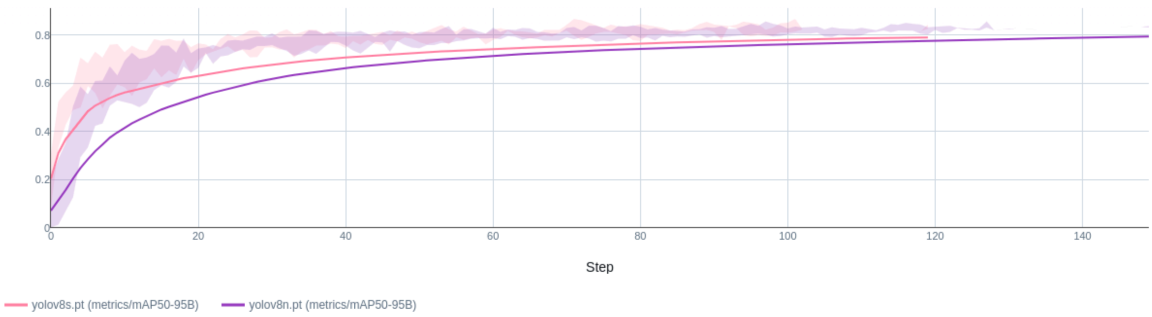


Figure 36. Evolution of mAP50-95 through epochs grouped by model variants.

From the observations above, it's notable that some of the model training experiments ended prematurely due to the implementation of early stopping techniques aimed at mitigating overfitting. This tendency to stop earlier is more pronounced with larger models and larger datasets, which aligns with expectations.

The following Figure 37 underscores the significance of employing such techniques and optimizing hyperparameters effectively. It is evident that the mean Average Precision (mAP) has converged for all models derived from the experiments, and the inference speed is notably high when evaluated on Kaggle's GPU infrastructure.

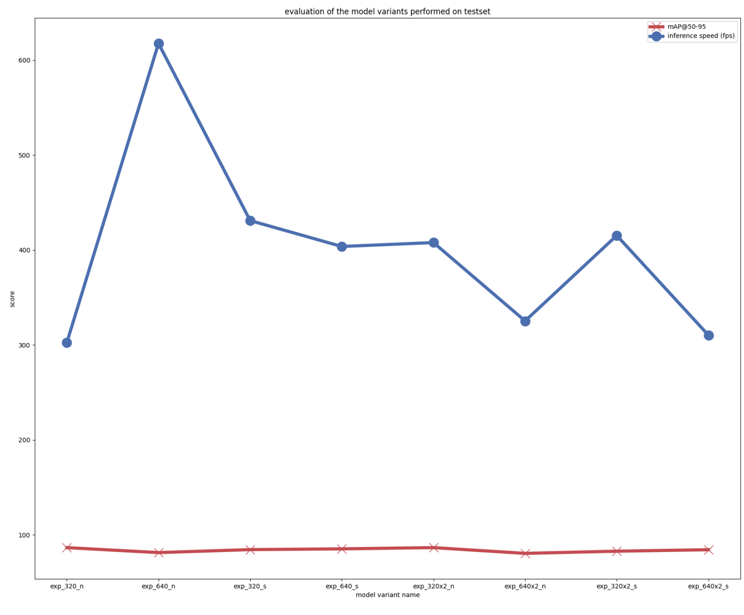


Figure 37. Evaluation of last model version variants

To understand how we built the best performing model, please refer to section 3.4 for detailed insights and methodologies used in the process.

3.3.3 Deployment and Testing

Upon achieving a satisfactory model performance, deployment to the Jetson Nano occurred. Real-time camera input from the production line served as the testing environment, a significant change from the controlled conditions of static images or captured videos used previously.

This stage introduced a new set of challenges related to camera input:

- Lighting Variations: Fluctuating lighting conditions in the production environment.
- Exposure and Brightness: Inconsistent exposure and brightness levels affecting image quality.
- Camera Frame Rate vs. Conveyor Belt Speed: Slower camera frame rate compared to conveyor belt speed, resulting in image blur and negatively impacting model performance.

To overcome these challenges, efforts focused on:

- Positioning the camera for an optimal view that closely matches the dataset image perspective.
- Identifying the ideal delay between object detection and the execution of corrective actions (e.g., bottle ejection).

These refinements aimed to ensure the model's effectiveness in a real-world production line setting.

3.4 Best Performing Model

3.4.1 Dataset Construction

A. Data Collection Strategy

The data collection process involved iterative visits to the Toudja factory to test and refine the models. This iterative approach ensured the dataset captured real-world variations encountered in the production environment.

Industrial Camera Integration: An industrial gray-level camera served as the primary data acquisition tool, mimicking the actual input source for the final AI system.

Multiple Viewpoints: Images and videos were captured from various angles to identify the optimal camera placement for efficient detection and inspection on the production line. This included capturing data at different locations, such as the output of the filling machine and the output of the labeling machine.

Real-World Representation: The focus was on capturing images that closely reflected real-world scenarios, as opposed to relying solely on randomly captured images from supermarkets or personal settings.

Smartphone Integration: Smartphone cameras were utilized as a supplementary data source to enhance the model's generalizability.

B. Data Selection Process

Given the initial video format of the dataset, a meticulous selection process was undertaken:

Screenshot Extraction: Representative screenshots were extracted from the videos to capture the most relevant aspects of the actual production setting.

Repetitive Image Removal: Redundant and highly similar photos were excluded to ensure a more diverse and informative dataset.

This rigorous selection process resulted in a final dataset of 130 high-quality images that effectively represented the complexities of the real-world production environment, as sample is shown in the Figure 38 below.

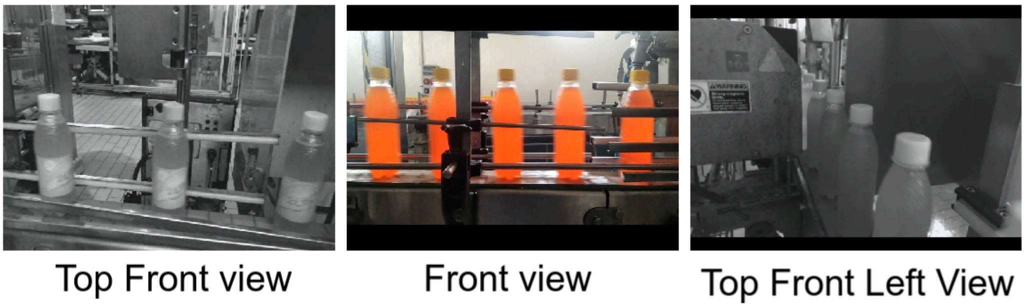


Figure 38. Different point of views of images

C. Image Size Standardization

A crucial step in the data preprocessing stage involved ensuring consistent image size and quality across the dataset. This was particularly important due to the significant size discrepancies between images captured using smartphones (3024 x 4032 pixels) and the industrial camera (640 x 480 pixels).

The adopted approach involved resizing all images to a standard dimension of 640 x 640 pixels while preserving the original aspect ratio. To achieve this, a padding technique was employed, see Figure 39 and Figure 40 below. Padding adds blank pixels around the image to maintain the original ratio while conforming to the desired dimensions. This standardization step aimed to:

Enhance Model Generalizability: By maintaining the aspect ratio of the bottles, the model's ability to recognize bottles regardless of orientation within the image is improved.

Reduce Image Size: Resizing the images to a smaller dimension facilitated efficient upload to Roboflow, the chosen platform for image annotation.

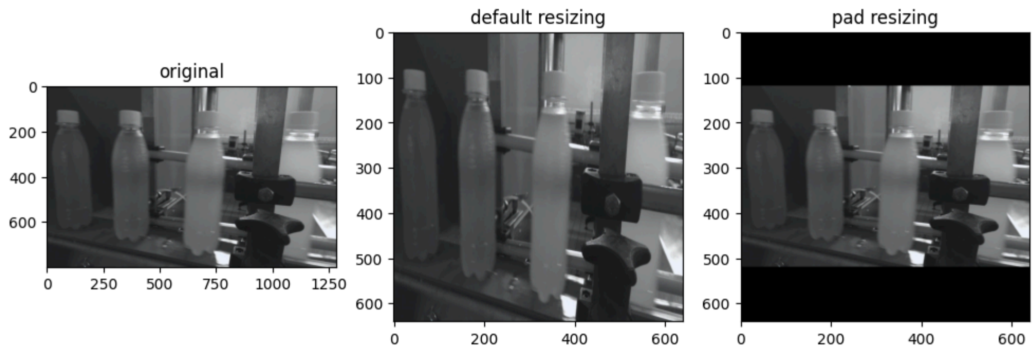


Figure 39. Comparison between standard resizing function and our custom function on horizontal image

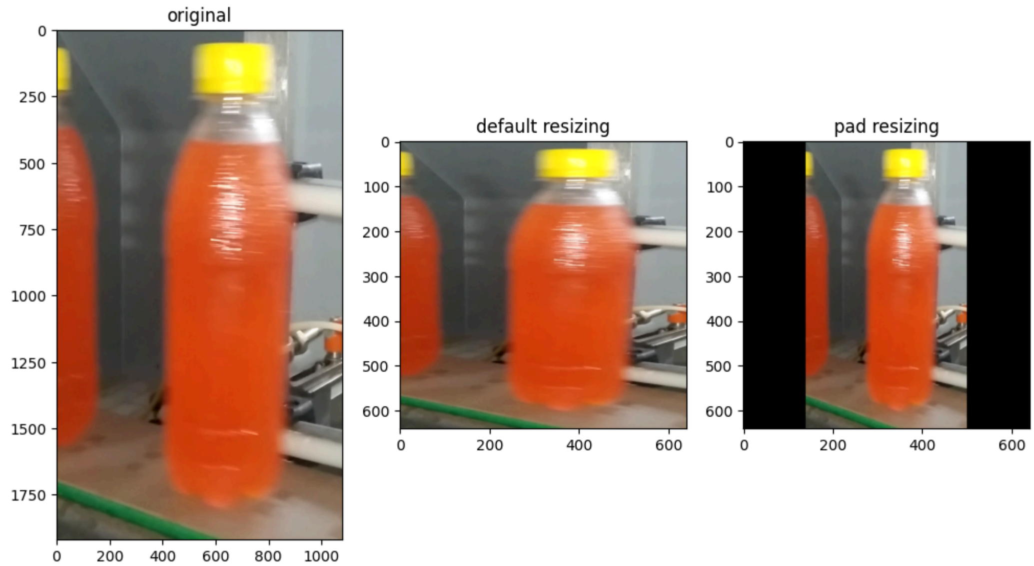


Figure 40. parison between standard resizing function and our custom function on vertical image

D. Data Annotation with Roboflow

Roboflow Platform: Roboflow's web application served as the primary tool for image annotation in this project. This user-friendly platform streamlines image dataset creation for various tasks, including object detection. Notably, it allows exporting the annotated dataset in formats compatible with YOLOv8, the chosen object detection model.

Project Setup and Collaboration: A Roboflow project was established to manage the image annotation process collaboratively. This project facilitated the inclusion of multiple contributors for efficient teamwork.

Class Label Definition: Distinct object labels were defined to represent the elements within the images: cap, emptyBottle, fullBottle, label, and liquid.

Image Upload and Annotation: The 130 selected images were uploaded, and the annotation task was assigned to designated personnel.

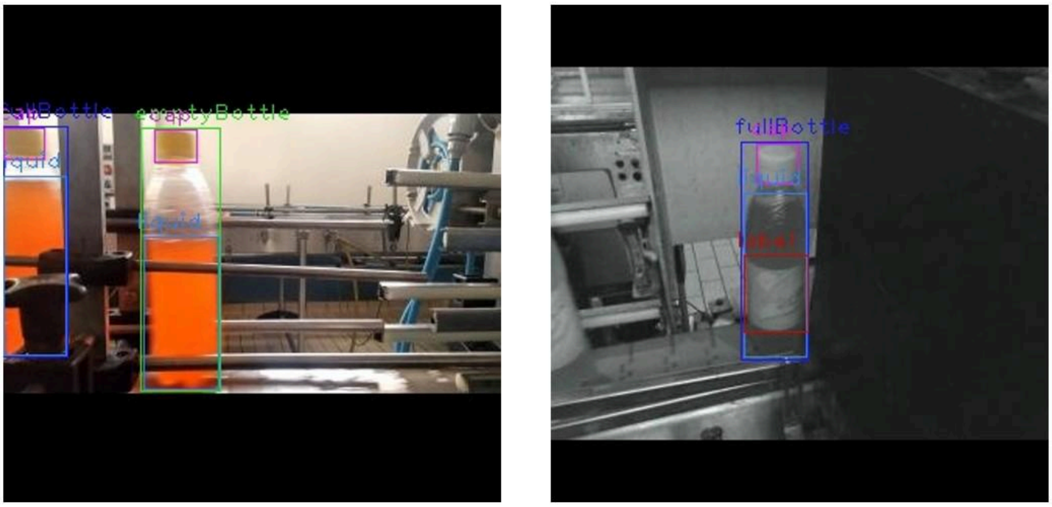
Annotation Criteria: To guide the model towards extracting relevant features, a specific annotation criterion was established. Bottles were only annotated if a human observer could definitively classify them as empty or full.

Dataset Splitting and Augmentation: Following annotation, the images were incorporated into the Roboflow dataset. Best practices were followed by splitting the dataset into 78% training, 14% validation, and 8% testing sets to prevent overfitting during model training.

Two dataset versions were generated:

- v1: No data augmentation (original images).
- v2: Brightness and exposure augmentations for increased model robustness.

The final stage involved exporting the dataset in the YOLOv8 format for compatibility with the chosen object detection model.



A B

Figure 41. Examples of dataset images

For illustration purposes (A vs B) as in Figure 41 above, both bottles were annotated in A because their fill level was readily distinguishable . Conversely, the left bottle in B presented an ambiguous fill level, making definitive classification challenging. Consequently, this bottle was excluded from annotation.

E. Class Distribution Analysis

Images	annotations	image ratio
130 annotated images 2 null examples as bg	876 annotations Avg: 6.7 per image	square 640 * 640

Table 3. Dataset Health

From Table 3 above and manual inspection of the Figures revealed the following class distribution within the dataset:

- Full Bottles: 240 (88.6%)
- Empty Bottles: 31 (11.4%)

- Liquid: 270 annotations (1 extra compared to total bottles) - This suggests a potential mislabeling of one image.
- Caps: 267 annotations (4 bottles missing caps)
- Labels: 68 annotations (203 bottles missing labels)

The dataset was split into training, validation, and testing sets, maintaining the predefined ratios (78%, 14%, and 8%, respectively) to ensure balanced representation within each subset, see Figure 42

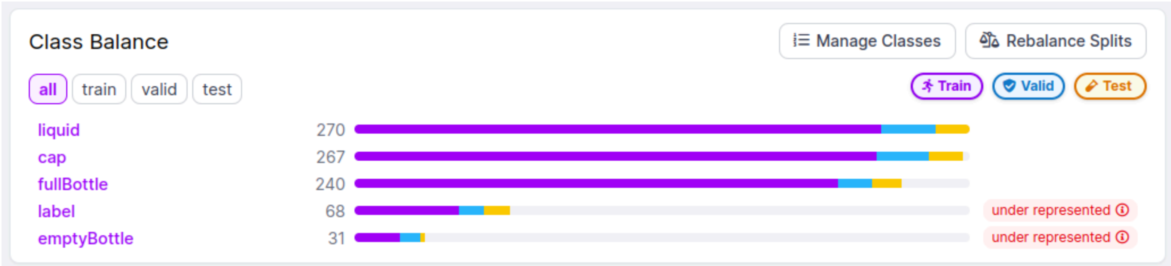


Figure 42. Class balance through dataset splits

Class Imbalance

It is noteworthy that the "empty bottle" and "label" classes are under-represented compared to the "full bottle" class, as in Figure 43, this reflects a potential real-world scenario where empty bottles and unlabeled bottles might be less frequent on the production line. This imbalance could be addressed in future iterations by employing specific data augmentation techniques to generate additional examples for these under-represented classes.

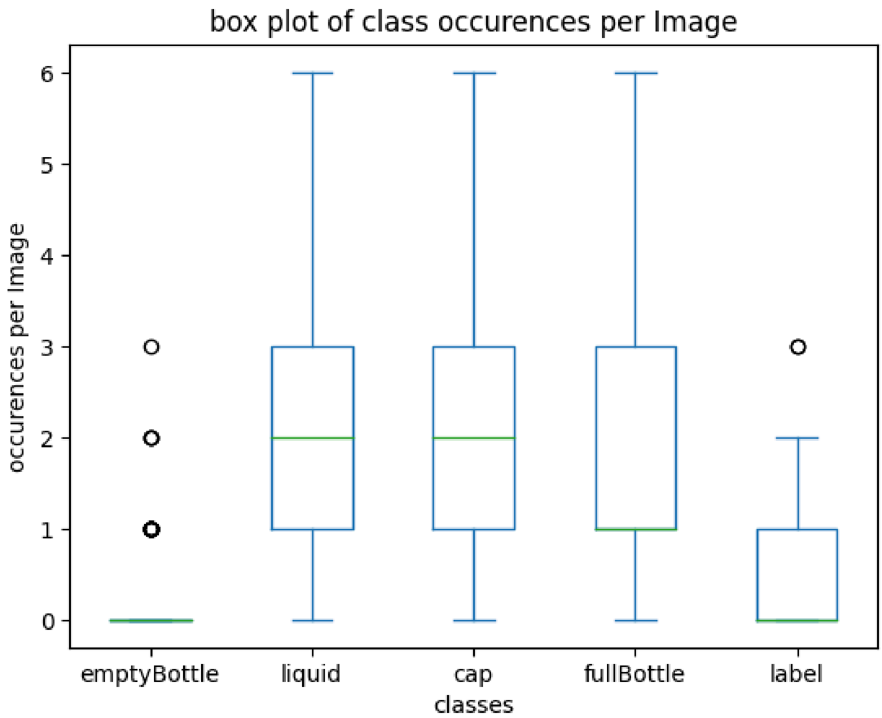


Figure 43. Box plot of instances per image

From the box plot we can conclude that Table 4 :

class	emptyBottle	liquid	cap	fullBottle	label
1st quartile	0	1	1	1	0
mediane	0	2	2	1	0
3rd quartile	0	3	3	3	1
max	0, rarely 1,2,3	6	6	6	2, rarely 3

Table 4. Class instances distribution per image

3.4.2 Justification for YOLOv8 Selection

The YOLOv8 object detection model was chosen for this project due to its compelling strengths:

Balanced Performance: YOLOv8 demonstrably achieves a desirable equilibrium between accuracy and inference speed, critical for real-time production line applications.

Robustness: This model exhibits robustness in handling outliers and under-represented classes within the dataset, potentially encountered in the production environment (e.g., empty bottles) as observed in our data analysis.

Integrated Tracking: A significant advantage of YOLOv8 is its built-in multi-object tracker. This functionality is particularly valuable for scenarios with potential object occlusion or motion blur across frames. The tracker enhances detection reliability by maintaining the persistence of previously identified objects, even during temporary occlusions. This improved reliability facilitates more informed decision-making based on tracked objects.

While acknowledging that no model is infallible, YOLOv8's combination of these strengths makes it a well-suited choice for the project's requirements in a real-world production line setting.

3.4.3 Fine-Tuning YOLOv8 with Online Augmentation

The Kaggle platform provided a GPU-powered environment for training the YOLOv8 model. To facilitate experiment tracking and comparison, MLflow was employed.

The "nano" variant of YOLOv8 was chosen, prioritizing both efficiency and accuracy for real-time deployment on the Jetson Nano. An input image size of 320 x 320 pixels was used, balancing model performance with computational demands.

Online data augmentation was the preferred approach to avoid redundancy with any potential offline pre-processing steps. This strategy dynamically augments images during training, enriching the dataset and potentially improving model generalizability. Here's a breakdown of the specific augmentation techniques used:

- Translation (translate: 0.1): Introduces slight random shifts in the image position, simulating potential variations in object placement within the scene.
- Color Jitter (BGR): Applies random variations to the image's color channels, mimicking lighting inconsistencies that might occur in real-world settings.
- Mosaic Augmentation (mosaic: 0): (Disabled in this experiment) This technique combines four training images into a single image, enhancing object diversity within a single training instance.

- Mixup Augmentation (mixup: 0.0): (Disabled in this experiment) This technique blends elements from two different images, promoting model robustness to potential variations.
- Copy-Paste Augmentation (copy_paste: 0.2): Randomly copies and pastes image regions, potentially introducing occlusions or creating new object arrangements.
- Scale Augmentation (scale: 0.5): Randomly scales the image size, simulating potential variations in object distances from the camera.

The training process employed the following hyperparameters:

- Epochs: 150
- Patience: 30:
- Pre Trained: True

For other hyperparameters, the default values recommended by the official YOLOv8 Ultralytics documentation were utilized [109]. This approach ensured a well-established foundation for fine-tuning while allowing for customization of specific parameters based on the project's requirements.

3.4.4 Performance and metrics

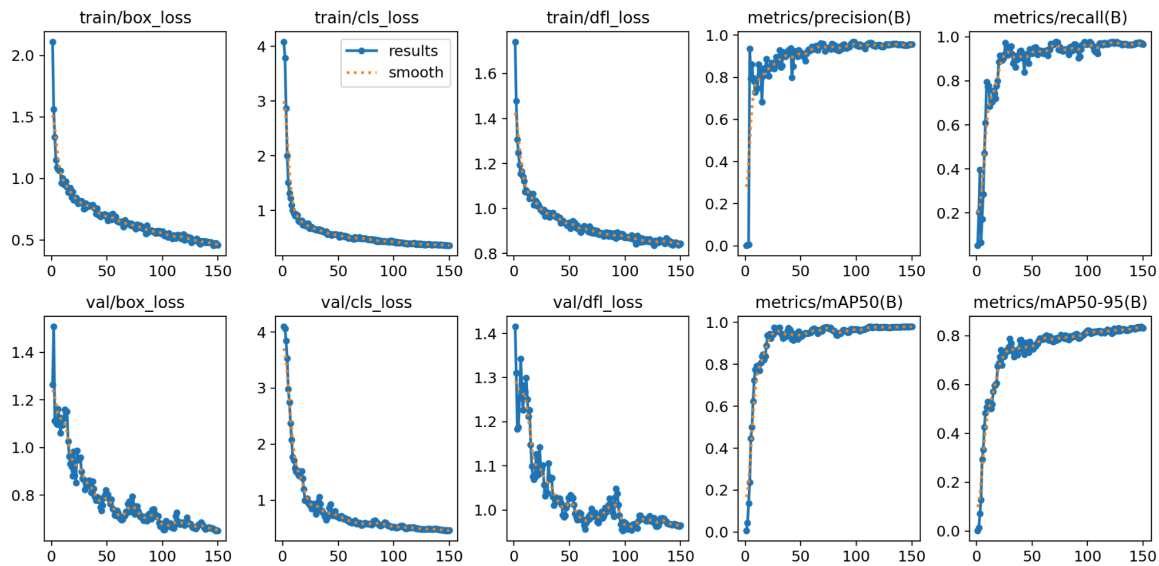


Figure 44. Loss and metrics curves of train and val of YOLOv8

The training process exhibited successful model convergence. The visual similarity between the training and validation loss curves (Figure 44 above) suggests that the model learned effectively from the data without experiencing overfitting or underfitting. This indicates the model's ability to generalize well to unseen data, a crucial characteristic for real-world deployment.

The model achieved a promising performance metric of 0.97 mAP50 (mean Average Precision at an Intersection over Union (IoU) threshold of 0.5) across all classes (illustration below). This metric signifies the model's ability to accurately detect and localize objects within the images. A high mAP50 value indicates strong overall performance, see Figure 45.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	18	82	0.953	0.962	0.974	0.834
cap	18	23	0.956	1	0.981	0.767
emptyBottle	8	9	1	0.808	0.926	0.874
fullBottle	12	15	0.865	1	0.974	0.848
label	9	11	0.991	1	0.995	0.9
liquid	18	24	0.955	1	0.995	0.779

Figure 45. Validation metrics

Benchmarks on the Kaggle platform utilizing a P100 GPU yielded promising speed results, with the model achieving 96 FPS, as shown in Figure 46

Format	Status❔	Size (MB)	metrics/mAP50-95(B)	Inference time (ms/im)	FPS
PyTorch	✅	5.9	0.8336	10.42	95.98

Figure 46. Benchmark of YOLOv8 model on Kaggle GPU

Encouragingly, the model achieved 20 FPS on your personal computer during video processing. This frame rate is sufficient for real-time applications, suggesting its potential suitability for deployment based on your hardware capabilities.

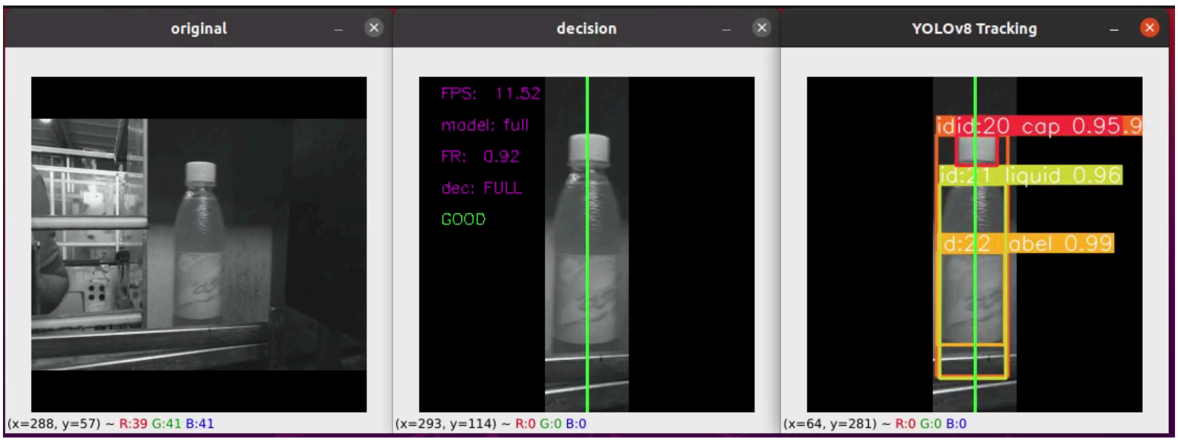


Figure 47. Screenshot of YOLOv8 model running on CPU achieving 11.52 fps

Note: In Figure 47 above, the low FPS (frames per second) observed is due to three video screens playing simultaneously, which requires additional time to render. Additionally, a video recorder was enabled to capture the video from which this screenshot was taken.

3.4.5 Post-Processing and Decision-Making

The post-processing stage serves as the final step before real-world action is taken by the system.

Empty Bottle Detection: Upon identifying a bottle centered within the frame, the system classifies it as empty or full using the model's predictions. An empty bottle classification triggers a signal to the sorting mechanism (e.g., boxer) to eject the bottle.

Additional Class Roles: While the primary decision relies on the empty/full classification, the other detected classes play supporting roles:

- Cap Detection: The "cap" class verifies the presence or absence of a cap on the bottle.
- Label Detection: The "label" class confirms the presence or absence of a label on the bottle.

Liquid Level for Empty Bottle Verification: The "liquid" class provides a secondary verification for emptiness. Visual emptiness can sometimes be subjective. By analyzing the liquid level relative to the bottle height and applying a predefined threshold (e.g., thresh = 0.85), the system can objectively determine emptiness, see Figure 48 Furthermore, this data can be potentially utilized for future model retraining to improve automatic threshold detection.

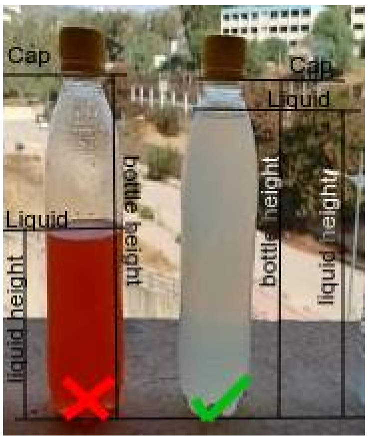


Figure 48. How Liquid class is used to classify emptiness of a bottle

Addressing Independent Object Detection: It's important to acknowledge that YOLO detectors treat objects independently. For instance, if two bottles and two caps are detected, the model doesn't inherently associate each cap with its corresponding bottle. To address this, a distance function was implemented based on the intersection area between bounding boxes of bottles and parts (caps and labels) relative to the area of the part's bounding box. If this distance exceeds a certain threshold, the part is assigned to the specific bottle. distance = Intersection_area(bottle, part) / area(part)

Part Association Example: Based on the calculated distances, the table below came as output Table 5:

parts	Bottle 1	Bottle 2
cap 1	0.95	0
cap 2	0	1
liquid 1	0	0.99
label 1	0	0.9
liquid 2	1	0

Table 5. predictions examples

Now the system can assign each part to its corresponding bottle:

Bottle 1: cap 1, liquid 2

Bottle 2: cap 2, liquid 1, label 1

3.5 Conclusion

This chapter detailed the development of an AI system for production line empty bottle detection. Data pre-processing, annotation, model selection, fine-tuning, and post-processing for real-world decision-making were addressed.

The chosen YOLOv8 model exhibited promising training performance (high mAP50: 0.97) and potential for real-time processing on powerful hardware. However, deployment on the target Jetson Nano platform revealed challenges: YOLOv8's lack of optimization resulted in dependency issues and a low frame rate (7 FPS), hindering real-time applicability.

These findings emphasize the crucial role of hardware considerations during model selection and the potential need for platform-specific optimization strategies. Future work could explore alternative lightweight models optimized for Jetson Nano or delve deeper into YOLOv8 optimization for this platform to achieve real-time performance.

Overall, the project demonstrates the feasibility of AI for empty bottle detection in production lines. The experience gained regarding model selection, optimization, and hardware considerations offers valuable insights for future development efforts.

Chapter 4: Challenges and Future Insights of AI in Industrial Inspection

While AI holds tremendous promise in enhancing manufacturing processes, supply chain leaders must navigate potential challenges associated with its implementation. As companies seek to leverage AI for industrial inspection, understanding and addressing these challenges is paramount. This chapter explores the complexities and limitations inherent in integrating AI into manufacturing, equipping supply chain leaders with insights to mitigate risks and optimize the effectiveness of AI-driven inspection systems. From data quality issues to algorithmic biases, it delves into the multifaceted landscape of challenges and offers strategies for overcoming them to ensure successful adoption and integration of AI in industrial inspection processes. Despite the challenges, the future of AI in industrial inspection is promising, particularly with the integration of Internet of Things (IoT) technology and drones, offering new avenues for innovation and advancement in quality control processes.

4.1 Challenges and Limitations

In the realm of AI, there are always some challenges that may face you during your ML system implementation, especially when it comes to AI based industrial inspection systems. As follows, some of the issues we faced trying to integrate our implemented model into Toudja's production line.

4.1.1 Data Quality

Data quality is a fundamental pillar in AI. Although it may seem redundant and tedious to emphasize, the fact remains that modern AI is data-driven or data-centric. Throughout the development of this project, we faced significant challenges in creating a high-quality dataset. This dataset needed to be minimally redundant, representative of real-world use cases, and sufficiently large.

We addressed this issue by capturing images and videos directly at the deployment site using the same camera that will be utilized as the video source input for the system. This approach ensured that our dataset was highly relevant and reflective of the actual conditions under which the system would operate.

4.1.2 Deployment

Albert Einstein famously said, “In theory, theory and practice are the same. In practice, they are not.” This quote aptly captures the challenges encountered when implementing an end-to-end ML pipeline. Machine learning models taught in courses and used in Kaggle competitions are valuable for learning, but real-world applications are far more complex. They involve dealing with numerous variations, unforeseen input data, and environmental variables.

Another critical realization is that machine learning is highly iterative and extends beyond deployment through continuous learning, monitoring, and feedback loops—hence the emergence of MLOps.

Regarding the latest iteration of our project, we encountered deployment issues because the Jetson Nano is outdated for newer models like YOLOv8 and presents significant dependency challenges. The solutions are either to downgrade to an older version of YOLO, sacrificing some speed and accuracy while implementing a tracking algorithm, or to purchase a newer Jetson with the latest version of JetPack, which is considerably more expensive.

4.1.3 Speed and Accuracy

As you may know, achieving 100% accuracy is impossible with current technologies. However, it is crucial to develop a model that is as accurate as possible without overfitting. This can be achieved through fine-tuning, hyperparameter testing, and refining data until convergence.

Modern production lines operate at high speeds to meet demand, so our model must keep pace to avoid reducing the production rate. Slowing down production would counteract the purpose of implementing our AI Visual Inspection (AIVI) system, which is to increase efficiency and gains. This issue can be addressed by selecting the appropriate model architecture and necessary hardware.

4.1.4 Delay

Choosing the optimal location for your camera can result in a specific distance between the camera and the actuator that handles defective products—in our case, the actuator that ejects the bottles. To ensure that the product is ejected at the precise moment it reaches the actuator, we must introduce a delay between decision-making and action-taking. This delay must not impact the system's speed, the conveyor belt, or the program.

The solution lies in parallel programming using multithreading or multiprocessing. Each time the system detects a defective product, it launches a thread that waits for a predetermined amount of time (parameter t) before taking action and then terminates the thread. This approach ensures the continuity of our model in detecting products without disrupting the overall process.

4.1.5 Lack of System Integration

The integration of AI manufacturing systems with existing technology is imperative for enhancing manufacturing processes. However, legacy systems prevalent in manufacturing companies pose integration challenges due to various factors such as unclear return on investment (ROI) for upgrades and the complexity of implementing newer technology. It is essential for manufacturing companies seeking to integrate AI with their current technology to undertake the following actions:

1. Conduct a comprehensive review of the areas within the organization that require integration with AI-powered manufacturing systems.
2. Engage in discussions with AI vendors to understand the integration capabilities of the technology.
3. Explore the necessary upgrades or modifications required for existing manufacturing systems to facilitate seamless integration with AI technology

4.2 Future Trends and Research Direction

4.2.1 Explainable AI XAI

Explainable AI (XAI) has emerged as a crucial development in the manufacturing and visual inspection context, aiming to enhance the transparency and interpretability of AI systems. By providing insights into the decision-making processes of AI models, XAI enables humans to understand and trust AI-generated outcomes. Through techniques such as decision trees and rule-based systems, XAI elucidates the rationale behind model predictions, fostering acceptance and adoption of AI technologies. The global XAI market is expected to grow significantly, driven by increasing demand in manufacturing and visual inspection applications, with an estimated value of $1.2 billion by 2027. As XAI continues to evolve, it is poised to address challenges and democratize access to AI technologies, shaping a more transparent and accessible future for manufacturing and visual inspection industries. [103][122]

4.2.2 AI Collaborative Robots (CoBots)

AI-powered cobots represent a transformative force in the manufacturing landscape, offering advanced capabilities for efficiency and safety. With adaptive features, safe collaboration, intuitive interaction, and predictive maintenance, these cobots are revolutionizing industries. Functioning as versatile entities, cobots can be customized to suit various manufacturing requirements. While akin to humans in needing training, they can acquire skills rapidly through recorded data and pre-programmed patterns.

The integration of AI-powered cobots is expected to deepen collaboration between humans and machines, enhancing both productivity and job satisfaction. Fundamentally, their role is to undertake tasks beyond human capability or those that are hazardous, thereby optimizing operations. [104][125]

4.2.3 Industry 5.0

While Industry 4.0 was characterized by its technological orientation, Industry 5.0 adopts a value-driven approach.

Despite the considerable strides in efficiency and productivity facilitated by Industry 4.0, the paradigm also introduced challenges such as workforce displacement, rigidity stemming from high specialization, and substantial investments required for the production of intricate products. In contrast, Industry 5.0 emphasizes collaboration between skilled human labor and robotics to enhance value-adding processes and elevate product personalization. This conceptual framework underscores the utilization of advanced technologies to empower human workers, ensuring heightened speed, safety, and efficiency in operations, see Figure 49. [105][124]

Constructed upon three fundamental pillars human-centricity, resilience, and sustainability Industry 5.0 carries profound implications for manufacturers.

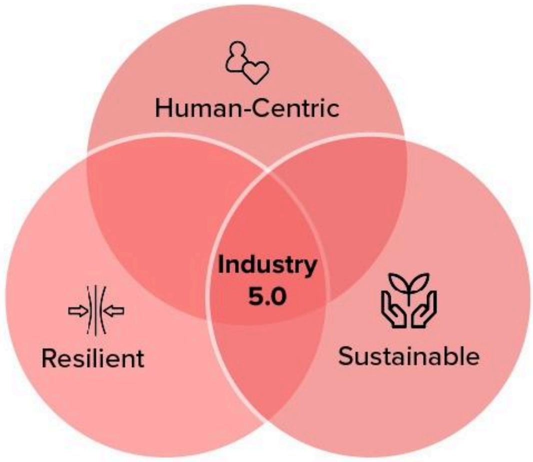


Figure 49. The three fundamental pillars of Industry 5.0 [121]

4.2.4 BlockChain, Iot and AI Synergy

The amalgamation of Blockchain, IoT, and AI constructs a resilient framework that extends the capabilities of each component. Blockchain functions as a secure platform for IoT devices, preserving an unalterable ledger of device data and facilitating intricate AI analyses while guaranteeing protection against tampering. AI complements this arrangement by delivering sophisticated data analysis, pattern recognition, and predictive maintenance, enabling proactive issue detection and instantaneous decision-making.

This synergy not only enhances data handling efficiency and security across diverse sectors but also introduces elevated levels of automation and innovation. Through the integration of these technologies, enterprises can capitalize on the potency of secure, self-governing, and intelligent systems to unlock unparalleled value and opportunities, surpassing the potential of individual technologies.

In a quality control scenario, AI aids in defect identification and ensures product quality throughout the manufacturing process. Blockchain furnishes a secure and unalterable log of all quality inspections and maintenance activities, vital for regulatory compliance and auditability. [106][123]

4.3 Conclusion

In conclusion, the challenges and future insights regarding AI in industrial inspection underscore the need for proactive approaches to address existing hurdles and harness emerging opportunities. Despite challenges such as data bias and integration complexities, the potential for AI to revolutionize industrial inspection is immense. By leveraging techniques like Explainable AI and fostering collaboration between humans and machines, industries can enhance transparency, efficiency, and safety in inspection processes. Looking ahead, the continued evolution of AI holds promise for overcoming current limitations and driving further advancements in industrial inspection. Embracing these innovations with a forward-thinking mindset will be pivotal in realizing the full potential of AI in transforming the industrial inspection landscape.

General Conclusion

This thesis has explored the transformative journey of industrial inspection, highlighting the limitations of traditional manual methods and the promise of AI-powered visual inspection (VI). The integration of various Machine Learning (ML) techniques, particularly Support Vector Machines and decision trees, laid the groundwork for more sophisticated systems. However, the true revolution arrived with Deep Learning, specifically Convolutional Neural Networks (CNNs).

CNNs excel at feature extraction and pattern recognition, enabling superior defect detection in complex industrial images. They have become the workhorse of AI-based VI due to their ability to learn intricate relationships within vast image datasets.

The recent emergence of YOLO models marks a significant leap towards real-time applications. Unlike traditional object detection methods, YOLO offers a single-stage approach, significantly reducing processing time while maintaining high accuracy. This real-time capability makes YOLO ideal for high-speed production lines, enabling immediate identification and rectification of defects.

The chapter on the YOLO-based inspection system exemplified these advantages. While the chosen YOLOv8 model demonstrated promising results in training and potential for real-time processing on powerful hardware, challenges arose during deployment on the target Jetson Nano platform. This underscores the importance of considering hardware constraints during model selection and the potential need for platform-specific optimization strategies.

Looking forward, even more sophisticated VI techniques are on the horizon. Integration with robotics and sensor fusion holds immense potential for creating intelligent systems capable of real-time, comprehensive analysis across various industrial sectors. By embracing these advancements, industries can achieve unparalleled levels of efficiency, accuracy, and safety within their production processes.

In conclusion, the future of industrial inspection is undeniably intelligent and powered by AI. By leveraging powerful Deep Learning architectures like CNNs, the real-time efficiency of YOLO models, and ongoing advancements in AI, we can forge a path towards a future where quality is assured, safety is prioritized, and industrial processes are optimized for success.

References

[1] “Industrial Inspection: An In-depth Analysis”, Energy-Robotics, Nov. 16, 2023. [Online]. Available: https://www.energy-robotics.com/post/industrial-inspection. [Accessed: Jun. 03, 2024]
[2] D. T. Pham and R. J. Alcock, Smart Inspection Systems: Techniques and Applications of Intelligent Vision. 2002.
[3] J. Khan, “Automated Visual Inspection Systems & how do they work?”, nNanoNets, May 19, 2021. [Online]. Available: https://nanonets.com/blog/ai-visual-inspection/. [Accessed: Jun. 03, 2024]
[4] “What Is Visual Inspection?”, IBM. [Online]. Available: https://www.ibm.com/topics/visual-inspection. [Accessed: May 28, 2024]
[5] R. Hryniewicz, “AI Visual Inspection: Defect Detection in Manufacturing”, NeuroSYS, Apr. 22, 2024. [Online]. Available: https://neurosys.com/blog/ai-defect-detection-in-manufacturing. [Accessed: Jun. 03, 2024]
[6] C. Yildiz, “The role of Artificial Intelligence in industrial inspections”, LinkedIn, Jan. 02, 2023. [Online]. Available: https://www.linkedin.com/pulse/role-artificial-intelligence-industrial-inspections-cenk-yildiz. [Accessed: Jun. 03, 2024]
[8] West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer". Brigham Young University, College of Physical and Mathematical Sciences
[9] T. Nolle, “AI and IoT: How do the internet of things and AI work together?”, TechTarget, Jun. 27, 2023. [Online]. Available: https://www.techtarget.com/iotagenda/tip/AI-and-IoT-How-do-the-internet-of-things-and-AI-work-together. [Accessed: Jun. 03, 2024]
[10] “Traditional Inspection Methods Vs Digital Inspection Software: A Comparative Analysis”, Intellinet Systems. [Online]. Available: https://www.intellinetsystem.com/blogs/traditional-inspection-methods-vs-digital-inspection-software, https://www.intellinetsystem.com/traditional-inspection-methods-vs-digital-inspection-software. [Accessed: Jun. 03, 2024]
[12] “Automated Visual Inspection Applications & Examples”, MoviTherm, Mar. 21, 2024. [Online]. Available: https://movitherm.com/2024/03/21/blog/automated-visual-inspection-applications-examples/. [Accessed: May 28, 2024]
[14] Stuart, A.; Ord, K. (1994), Kendall's Advanced Theory of Statistics: Volume I – Distribution Theory, Edward Arnold, §8.7
[16] “How to Use AI for Quality Control”, Sciotex. [Online]. Available: https://sciotex.com/how-to-use-ai-for-quality-control/. [Accessed: Jun. 03, 2024]
[17] “What is AI-based Visual Inspection and its Use cases?”, TagX, Aug. 18, 2023. [Online]. Available: https://www.linkedin.com/pulse/what-ai-based-visual-inspection-its-use-cases-tagx. [Accessed: Jun. 03, 2024]
[18] “AI, ML, AL & DL: What's the Difference?”, F eight federal [Online]. Available: https://f8federal.com/differences-ai-ml-al-dl/. [Accessed: Jun. 03, 2024]
[19] “What Is Machine Learning (ML)?”, IBM. [Online]. Available: https://www.ibm.com/topics/machine-learning. [Accessed: Jun. 03, 2024]
[20] V. Kanade, “What Is Machine Learning? Definition, Types, Applications, and Trends”, Spiceworks, Apr. 04, 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/. [Accessed: Jun. 03, 2024]

[25] “What is Deep Learning? | IBM”, IBM. [Online]. Available: https://www.ibm.com/topics/deep-learning. [Accessed: Jun. 03, 2024]
[28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” arXiv (Cornell University), Jan. 2015, doi: 10.48550/arxiv.1512.03385.

References

[29] Simonyan, “Very deep convolutional networks for large-scale image recognition,” (No Title), Jan. 2015, [Online]. Available: https://cir.nii.ac.jp/ja/crid/1371977243859268618
[31] C. Szegedy et al., “Going Deeper with Convolutions,” arXiv (Cornell University), Jan. 2014, doi: 10.48550/arxiv.1409.4842.
[33] “What is Computer Vision?”, IBM. [Online]. Available: https://www.ibm.com/topics/computer-vision. [Accessed: Jun. 03, 2024]
[34] J. Murel, “What is Object Detection?”, IBM, Jan. 03, 2024. [Online]. Available: https://www.ibm.com/topics/object-detection. [Accessed: Jun. 04, 2024]
[35] “What Is Image Segmentation?”, IBM, Jun. 04, 2024. [Online]. Available: https://www.ibm.com/topics/image-segmentation. [Accessed: Jun. 04, 2024]
[36] Ross Girschick, "Fast R-CNN," Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448, https://arxiv.org/abs/1504.08083
[37] J. Redmon, S. Divvala, R. Girshickand A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, May 2016, doi: 10.48550/arXiv.1506.02640.
[38] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: Single Shot MultiBox Detector," Proceedings of the European Conference of Computer Vision (ECCV), 2016, pp. 21-37, https://arxiv.org/abs/1512.02325 .
[39] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, stronger,” arXiv (Cornell University), Jan. 2016, doi: 10.48550/arxiv.1612.08242.
[40] J. Redmon and A. Farhadi, “YOLOV3: an incremental improvement,” arXiv (Cornell University), Jan. 2018, doi: 10.48550/arxiv.1804.02767.
[41] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOV4: Optimal speed and accuracy of object detection,” arXiv (Cornell University), Jan. 2020, doi: 10.48550/arxiv.2004.10934.
[42] M. S, M. S. S, K. T, and S. P, “Image Detection and Segmentation using YOLO v5 for surveillance,” Applied and Computational Engineering, vol. 8, no. 1, pp. 142–147, Aug. 2023, doi: 10.54254/2755-2721/8/20230109.
[43] C. Li et al., “YOLOV6: A Single-Stage Object Detection Framework for Industrial Applications,” arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2209.02976.
[44] 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,” arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2207.02696.
[45] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8,” arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2305.09972.
[46] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,” Machines, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.
[63] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, “A deep learning-based framework for an automated defect detection system for sewer pipes,” *Automation in Construction*, vol. 109, p. 102967, Jan. 2020, doi: 10.1016/j.autcon.2019.102967.
[64] W. Wu and Q. Li, “Machine Vision Inspection of Electrical Connectors Based on Improved Yolo v3,” *IEEE Access*, vol. 8, pp. 166184–166196, Jan. 2020, doi: 10.1109/access.2020.3022405.
[65] Z. Hu, J. Zhou, B. Yang, and A. Chen, “Design of pipe-inspection robot based on YOLOV3,” *Journal of Physics. Conference Series*, vol. 2284, no. 1, p. 012023, Jun. 2022, doi: 10.1088/1742-6596/2284/1/012023.
[66] H. Chen, Z. He, B. Shi, and T. Zhong, “Research on Recognition Method of Electrical components based on YOLO V3,” *IEEE Access*, vol. 7, pp. 157818–157829, Jan. 2019, doi: 10.1109/access.2019.2950053.
[67] X. Chen, J. Lv, Y. Fang, and S. Du, “Online detection of surface defects based on improved YOLOV3,” *Sensors*, vol. 22, no. 3, p. 817, Jan. 2022, doi: 10.3390/s22030817.
[68] F. Guo, Y. Qian, and Y. Shi, “Real-time railroad track components inspection based on the improved YOLOv4 framework,” *Automation in Construction*, vol. 125, p. 103596, May 2021, doi: 10.1016/j.autcon.2021.103596.
[69] S. Fan *et al.*, “Real-time defects detection for apple sorting using NIR cameras with pruning-based YOLOV4 network,” *Computers and Electronics in Agriculture*, vol. 193, p. 106715, Feb. 2022, doi: 10.1016/j.compag.2022.106715.

References

[70] Z.-H. Chen and J.-C. Juang, “YOLOV4 Object Detection Model for nondestructive radiographic testing in aviation maintenance tasks,” *AIAA Journal/AIAA Journal on Disc*, pp. 1–6, Oct. 2021, doi: 10.2514/1.j060860.
[71] M. Zhang and L. Yin, “Solar Cell Surface Defect Detection Based on Improved YOLO v5,” *IEEE Access*, vol. 10, pp. 80804–80815, Jan. 2022, doi: 10.1109/access.2022.3195901.
[72] R. Jin and Q. Niu, “Automatic fabric defect detection based on an improved YOLOV5,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–13, Sep. 2021, doi: 10.1155/2021/7321394.
[73] K. Chen *et al.*, “An Automatic Defect Detection System for Petrochemical Pipeline Based on Cycle-GAN and YOLO v5,” *Sensors*, vol. 22, no. 20, p. 7907, Oct. 2022, doi: 10.3390/s22207907.
[74] Z. Li, X. Tian, X. Liu, Y. Liu, and X. Shi, “A Two-Stage industrial defect detection framework based on Improved-YOLOV5 and Optimized-Inception-ResnetV2 models,” *Applied Sciences*, vol. 12, no. 2, p. 834, Jan. 2022, doi: 10.3390/app12020834.
[75] L. Wang, X. Liu, J. Ma, W. Su, and H. Li, “Real-Time Steel Surface Defect Detection with Improved Multi-Scale YOLO-v5,” *Processes*, vol. 11, no. 5, p. 1357, Apr. 2023, doi: 10.3390/pr11051357.
[76] H. F. Le, L. J. Zhang, and Y. X. Liu, “Surface defect detection of industrial parts based on YOLOV5,” *IEEE Access*, vol. 10, pp. 130784–130794, Jan. 2022, doi: 10.1109/access.2022.3228687.
[77] J. Xu, Y. Zou, Y. Tan, and Z. Yu, “Chip pad inspection method based on an improved YOLOV5 algorithm,” *Sensors*, vol. 22, no. 17, p. 6685, Sep. 2022, doi: 10.3390/s22176685.
[78] Y. Duan, S. Qiu, W. Jin, T. Lu, and X. Li, “High-Speed Rail tunnel panoramic inspection image recognition technology based on improved YOLOV5,” *Sensors*, vol. 23, no. 13, p. 5986, Jun. 2023, doi: 10.3390/s23135986.
[79] X. Li, C. Wang, H. Ju, and Z. Li, “Surface defect detection model for Aero-Engine components based on improved YOLOV5,” *Applied Sciences*, vol. 12, no. 14, p. 7235, Jul. 2022, doi: 10.3390/app12147235.

[81] J. Zheng, H. Wu, H. Zhang, Z. Wang, and W. Xu, “Insulator-Defect detection algorithm based on improved YOLOV7,” *Sensors*, vol. 22, no. 22, p. 8801, Nov. 2022, doi: 10.3390/s22228801.
[82] Y. Wang, H. Wang, and Z. Xin, “Efficient Detection model of steel strip surface defects based on YOLO-V7,” *IEEE Access*, vol. 10, pp. 133936–133944, Jan. 2022, doi: 10.1109/access.2022.3230894.
[83] J. Chen, S. Bai, G. Wan, and Y. Li, “Research on YOLOv7-based defect detection method for automotive running lights,” *Systems Science & Control Engineering*, vol. 11, no. 1, Mar. 2023, doi: 10.1080/21642583.2023.2185916.
[84] B. Chen and Z. Dang, “Fast PCB defect detection method based on FasterNet backbone network and CBAM attention mechanism integrated with feature fusion module in improved YOLOV7,” *IEEE Access*, vol. 11, pp. 95092–95103, Jan. 2023, doi: 10.1109/access.2023.3311260.
[85] H. Huang and K. Zhu, “Automotive parts defect detection based on YOLOV7,” *Electronics*, vol. 13, no. 10, p. 1817, May 2024, doi: 10.3390/electronics13101817.
[86] E. Dehaerne, B. Dey, S. Halder, and D. G. Stefan, “Optimizing YOLOV7 for semiconductor defect detection,” *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2302.09565.
[87] Y. Zhao *et al.*, “An oil and gas pipeline inspection UAV based on improved YOLOv7,” *Measurement + Control/Measurement and Control*, Feb. 2024, doi: 10.1177/00202940241230426.
[88] E. Bellou, I. Pisica, and K. Banitsas, “Aerial inspection of High-Voltage power lines using YOLOV8 Real-Time Object Detector,” *Energies*, vol. 17, no. 11, p. 2535, May 2024, doi: 10.3390/en17112535.
[89] M. Liu, M. Zhang, X. Chen, C. Zheng, and H. Wang, “YOLOV8-LMG: An improved bearing defect detection algorithm based on YOLOV8,” *Processes*, vol. 12, no. 5, p. 930, May 2024, doi: 10.3390/pr12050930.
[90] Q. Ling, N. A. M. Isa, and M. S. M. Asaari, “Precise detection for dense PCB components based on modified YOLOV8,” *IEEE Access*, vol. 11, pp. 116545–116560, Oct. 2023, doi: 10.1109/access.2023.3325885.

References

[91] J. Silva, P. Coelho, L. Saraiva, P. Vaz, P. Martins, and A. López-Rivero, “Validating the use of smart glasses in industrial quality control: a case study,” *Applied Sciences*, vol. 14, no. 5, p. 1850, Feb. 2024, doi: 10.3390/app14051850.
[92] M. H. Zubayer, C. Zhang, W. Liu, Y. Wang, and H. M. Imdadul, “Automatic defect detection of jet engine turbine and compressor blade surface coatings using a Deep Learning-Based algorithm,” *Coatings*, vol. 14, no. 4, p. 501, Apr. 2024, doi: 10.3390/coatings14040501.
[93] “Automated Visual Inspection: Use Cases and Implementation Tips”, *Itransition*, Apr. 11, 2024. [Online]. Available: https://www.itransition.com/computer-vision/automated-visual-inspection. [Accessed: May 28, 2024]
[94] “Case Study: Inspection of Printed Circuit Boards”, *Maddox AI*. [Online]. Available: https://www.maddox.ai/en/our-case-studies/inspection-of-printed-circuit-boards/. [Accessed: Jun. 08, 2024]
[95] “AI Visual Inspection of Glass Bottles | SOLOMON 3D”, *Solomon 3D*. [Online]. Available: https://www.solomon-3d.com/ai-visual-inspection-for-glass-bottles/. [Accessed: Jun. 08, 2024]
[96] “Maximo Visual Inspection”, *IBM*. [Online]. Available: https://www.ibm.com/products/maximo/visual-inspection. [Accessed: Jun. 08, 2024]
[97] “Product”, *Landing AI*, Jun. 08, 2024. [Online]. Available: https://landing.ai/platform. [Accessed: Jun. 08, 2024]
[98] “AI-Based Visual Quality Control”, *Maddox Ai*. [Online]. Available: https://www.maddox.ai/en/. [Accessed: Jun. 08, 2024]
[99] “Visual Inspection AI Google Cloud”, *Google Cloud*. [Online]. Available: https://cloud.google.com/solutions/visual-inspection-ai. [Accessed: Jun. 08, 2024]
[103]“11 New Technologies in AI: All Trends of 2023-2024”, *Devabit*. [Online]. Available: https://devabit.com/blog/top-11-new-technologies-in-ai-exploring-the-latest-trends/. [Accessed: Jun. 08, 2024]
[104]“AI and Collaborative Robots: The Future of Manufacturing”, *Cobots Online*, Mar. 22, 2024. [Online]. Available: https://cobotsonline.co.uk/blog/ai-and-collaborative-robots-the-future-of-manufacturing. [Accessed: Jun. 08, 2024]
[105]“What is Industry 5.0 and How is it Shaping Manufacturing?”, *Cobots Online*, Jan. 11, 2024. [Online]. Available: https://cobotsonline.co.uk/blog/what-is-industry-5-0. [Accessed: Jun. 08, 2024]
[106] “How Can Blockchain, IoT, and AI Collaborate to Drive Technological Innovation?”, *Block Convey*, Apr. 25, 2024. [Online]. Available: https://www.linkedin.com/pulse/how-can-blockchain-iot-ai-collaborate-drive-technological-ruwbc. [Accessed: Jun. 08, 2024]
[107] M. Bansal, A. Goyal, and A. Choudhary, “A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,” Decision Analytics Journal, vol. 3, p. 100071, Jun. 2022, doi: 10.1016/j.dajour.2022.100071.
[108]“Supervised Machine learning - Javatpoint,” *www.javatpoint.com*. Available: https://www.javatpoint.com/supervised-machine-learning
[109] N. Quan, “What is unsupervised learning?,” *Eastgate Software*, Jan. 26, 2024. Available: https://eastgate-software.com/what-is-unsupervised-learning/
[110] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, “Basics of supervised deep learning,” in *Studies in big data*, 2019, pp. 13–29. doi: 10.1007/978-981-13-6794-6_2.
[111] Intelligence artificielle,” *Dossiers Thématiques - Commission Nationale Pour La Protection Des Données - Luxembourg*. Available: https://cnpd.public.lu/fr/dossiers-thematiques/intelligence-artificielle.html
[112] A. Naebi and Z. Feng, “The performance of a Lip-Sync imagery model, new combinations of signals, a supplemental bond graph classifier, and deep formula detection as an extraction and root classifier for electroencephalograms and Brain–Computer interfaces,” *Applied Sciences*, vol. 13, no. 21, p. 11787, Oct. 2023, doi: 10.3390/app132111787.

References

[113] J. Torres, “What is YOLOv8? Exploring its Cutting-Edge Features - YOLOv8,” *YOLOv8*, Jan. 13, 2024. Available: https://yolov8.org/what-is-yolov8/
[114] Shet, Srijana. (2020). Artificial Intelligence for Every Individual? It's Easy if You Do it Smart. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS070354.
[115] OnestopNDT, “Types of visual inspection,” *OnestopNDT*, Jun. 29, 2024. Available: https://www.onestopndt.com/ndt-articles/visual-inspection-types
[116] S. Inspection, “The different types of sampling plans for QC inspections – Sunshine Inspection Service,” Jan. 03, 2019. Available: https://www.sunchineinspection.com/the-different-types-of-sampling-plans-for-qc-inspections-2/
[117] WikiMemoires, “Apprentissage automatique Vs intelligence artificielle ?,” *WikiMemoires*, Dec. 01, 2022. Available: https://wikimemoires.net/2020/12/apprentissage-automatique-vs-intelligence-artificielle/
[118] M. Ahmed, R. Seraj, and S. M. S. Islam, “The k-means Algorithm: A Comprehensive Survey and Performance Evaluation,” *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.
[119] Ultralytics, “GitHub - ultralytics/ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite,” *GitHub*. Available: https://github.com/ultralytics/ultralytics
[120] J. Camillo, “AI-Based Vision Technology AIDS vehicle inspection,” *2021-01-07 | ASSEMBLY*, Jan. 05, 2021. Available: https://www.assemblymag.com/articles/96075-ai-based-vision-technology-aids-vehicle-inspection
[121] S. Srivastava, “Industry 5.0 – Revolutionizing the Factory Floor with Human-Centric Manufacturing,” *Appinventiv*, Jun. 26, 2024. Available: https://appinventiv.com/blog/industry-5-0-manufacturing/
[122] P. Gohel, P. Singh, and M. Mohanty, “Explainable AI: current status and future directions,” *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arxiv.2107.07045.
[123] S. Guergov and N. Radwan, “Blockchain Convergence: Analysis of issues affecting IoT, AI and blockchain,” *International Journal of Computations, Information and Manufacturing*, vol. 1, no. 1, Dec. 2021, doi: 10.54489/ijcim.v1i1.48.
[124] P. K. R. Maddikunta *et al.*, “Industry 5.0: A survey on enabling technologies and potential applications,” *Journal of Industrial Information Integration*, vol. 26, p. 100257, Mar. 2022, doi: 10.1016/j.jii.2021.100257.
[125] Z. M. Bi, C. Luo, Z. Miao, B. Zhang, W. J. Zhang, and L. Wang, “Safety assurance mechanisms of collaborative robotic systems in manufacturing,” *Robotics and Computer-integrated Manufacturing*, vol. 67, p. 102022, Feb. 2021, doi: 10.1016/j.rcim.2020.102022.

Appendix A: Data Science Research Unit DSRU

Affiliation and Mission

The DSRU, affiliated with the Research Center for Scientific and Technical Information (CERIST), is a research unit dedicated to advancing the field of data science. Its core mission encompasses:

- Comprehensive Research: Conducting in-depth research and studies in the realm of data science.
- Knowledge Advancement: Contributing to the progress of scientific and technological knowledge and expertise in data science.
- Innovation Development: Fostering the development of innovative products and services leveraging data science principles.

The unit achieves these goals by:

- Valorization and Dissemination: Transforming research outcomes into practical applications and disseminating knowledge through various channels.
- Implementation and Education: Integrating research findings into training programs and facilitating real-world implementation of data science solutions.

Emphasis on Applied Research

The DSRU prioritizes a research approach that bridges the gap between theory and practice. This is evidenced by:

- Practical Foundations: Developing and applying theoretical frameworks that are firmly grounded in real-world experience.
- Industry Collaboration: Establishing objectives and structures through collaboration with industry partners, ensuring the unit's research addresses current industry needs.
- Dynamic Interplay: Promoting a dynamic interaction between theory and practice. The unit recognizes the importance of hands-on experience in identifying contemporary data science challenges while acknowledging that theoretical knowledge safeguards against professional obsolescence in this fast-paced field.

Role in the Internship

The DSRU played a pivotal role in the internship by providing essential resources and support:

- Material Resources: Supplying hardware necessary for the project, such as a Jetson Nano and an industrial camera.
- Technical Guidance: Offering invaluable mentorship and technical guidance throughout the internship.

Appendix B: SARL SPC GB, Toudja Factory

Company Background

SARL SPC GB, established in 1936, is a Limited Liability Company (LLC) headquartered in Bejaia, Algeria, specializing in the production of confectionery, carbonated beverages, fruit drinks, and mineral water.

The company emphasizes its commitment to quality and consumer health through investments in "latest generation water treatment stations" and rigorous quality control procedures. This dedication has demonstrably garnered customer satisfaction.

Internship Collaboration

- SPC GB served as the primary field testing location for the internship project.
- The company facilitated data collection through filming and descriptions of their machinery and production line, highlighting the limitations of traditional inspection methods in soda bottle production.

Soda Production Line Description

The production unit incorporates a series of interconnected machines:

1. Revopack: This semi-automatic machine efficiently positions preforms (preliminary bottle shapes) for subsequent blowing machines.
2. Blowing Machine: This high-speed machine manufactures PET bottles at a rate of up to 12,000 bottles per hour.
3. Filling Machine: This automated system fills and caps bottles, ensuring smooth flow through the production line.
4. Labeling Machine and Dater: This combined unit applies wraparound labels and prints production and expiration dates on the bottles.
5. Shrink-Wrapper: This machine applies a layer of shrink film for secure packaging.
6. Palletizer: This automated system efficiently stacks and arranges filled and wrapped bottles onto pallets.
7. Roller Wrapper: This final step provides a secure and stable outer wrapping for the palletized products.

System Integration:

The computer vision system designed during the internship is intended to integrate at two key points in the production line:

Post-Filling Inspection: This first integration point allows for real-time detection of empty bottles before unnecessary labeling and packaging.

Post-Labeling Inspection: The second integration point verifies the presence of labels, ensuring complete product information before final packaging.

Future developments aim to expand the system's functionality to include date and code verification on the soda bottles.

Abstract

This review explores the cutting-edge applications of computer vision (CV) in industrial inspection. We highlight the limitations of traditional methods and showcase how CV, coupled with Machine Learning (ML) and Deep Learning (DL) techniques like Convolutional Neural Networks (CNNs), is revolutionizing defect detection and quality control. The review explores the recent advancements in real-time object detection with YOLO models, emphasizing their potential for high-speed production lines. We conclude by discussing the future of CV in industrial inspection, including integration with robotics and sensor fusion for intelligent and comprehensive inspection systems.

Résumé

Ce mémoire explore les applications de pointe de la vision par ordinateur (CV) dans l'inspection industrielle. Nous mettons en évidence les limites des méthodes traditionnelles et montrons comment la CV, associée aux techniques de Machine Learning (ML) et de Deep Learning (DL) telles que les réseaux de neurones convolutifs (CNN), révolutionne la détection des défauts et le contrôle de la qualité. Le résumé explore les progrès récents en matière de détection d'objets en temps réel avec les modèles YOLO, en soulignant leur potentiel pour les lignes de production à grande vitesse. Nous concluons en discutant de l'avenir de la CV dans l'inspection industrielle, y compris l'intégration avec la robotique et la fusion de capteurs pour des systèmes d'inspection intelligents et complets.

ملخص

يستعرض هذا الملخص التطبيقات المتطورة للرؤية الحاسوبية (CV) في الفحص الصناعي. نسلط الضوء على قيود الطرق التقليدية ونوضح كيف أن الرؤية الحاسوبية، إلى جانب تقنيات التعلم الآلي (ML) والتعلم العميق (DL) مثل الشبكات العصبونية الترشيحية (CNNs)، تحدث ثورة في اكتشاف العيوب ومراقبة الجودة. يستكشف الملخص التطورات الحديثة في مجال كشف الأشياء بزمن حقيقي باستخدام نماذج YOLO، مع التأكيد على إمكاناتها لسلاسل الإنتاج عالية السرعة. نستنتج من خلال مناقشة مستقبل الرؤية الحاسوبية في الفحص الصناعي، بما في ذلك التكامل مع الروبوتات ودمج المستشعرات لأنظمة الفحص الذكية والشاملة. يقدم هذا الملخص الموجز معلومات قيمة حول الدور التحولي للرؤية الحاسوبية في ضمان الجودة والكفاءة داخل القطاع الصناعي.