ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in
Machine Learning for Computer Vision

Image Super-Resolution for Improved 6D Pose Estimation in Industrial Robotic Systems

SUPERVISOR
Prof. Samuele Salti

CANDIDATE
Luca De Dominicis

CO-SUPERVISORS
Dr. Davide Sapienza
Dr. Elena Govi
Prof. Marko Bertogna

Academic year 2023-2024
Session 2nd

To my parents
for their Kindness and Love

Contents

List of Figures

List of Tables

Abstract

Accurate six-dimensional (6D) pose estimation is critical for improving the precision and efficiency of robotic systems in industrial automation. However, existing methods struggle in complex environments with custom objects, variable lighting, and occlusions. This thesis explores the use of super-resolution techniques to enhance image quality and improve pose estimation accuracy.

The research integrates GDR-Net, a state-of-the-art 6D pose estimation network, with DRLN, an advanced super-resolution model. It hypothesizes that higher-resolution images can significantly boost accuracy. A synthetic dataset simulating real industrial conditions was developed to test this approach, demonstrating improved accuracy for small and medium size objects.

The findings highlight the potential of super-resolution to enhance 6D pose estimation robustness and pave the way for more resilient robotic vision systems in challenging industrial settings.

Chapter 1

Introduction

In the rapidly advancing field of industrial automation, robotic systems play a crucial role in increasing the efficiency and precision of manufacturing processes. One critical aspect of these systems is the ability to accurately determine the six-dimensional (6D) pose—position and orientation—of objects in complex, cluttered environments. However, current methods often face challenges with custom-designed objects, intricate geometries, and material properties such as reflectivity, making them less effective in industrial settings.

Recent advances in 6D pose estimation, powered by deep learning, have shown promise but still suffer significant limitations when applied in real-world conditions. Various lighting conditions, occlusions, and diverse object shapes often lead to inaccurate pose estimations. Moreover, the quality of the image data captured in such environments is usually suboptimal, further complicating the process of accurate pose estimation.

To address these challenges, this thesis focuses on testing **GDR-Net**, a state-of-the-art network for 6D pose estimation, under a range of difficult industrial conditions, including varying lighting, occlusions, and complex backgrounds. These tests are important for understanding modern pose estimation models' practical limitations and capabilities in real-world scenarios.

Furthermore, this research introduces a novel integration of super-resolution techniques with 6D pose estimation models to enhance accuracy and robustness in industrial environments. Specifically, we employ the **Densely Residual Laplacian Network (DRLN)**, an advanced super-resolution model, alongside GDR-Net. The hypothesis is that integrating super-resolution will improve the pose estimation performance by enhancing the quality of the image data used for training and inference.

An essential contribution of this thesis is the generation of a highly realistic synthetic dataset designed to emulate the complexities of real-world industrial environments. This dataset includes variations in lighting, occlusions, and a wide range of material properties, providing a robust platform for testing and improving 6D pose estimation methods.

The main objectives of this thesis are to:

1. Generate a comprehensive synthetic dataset that accurately reflects the challenges of industrial environments.

2. Evaluate the performance of GDR-Net under various industrial settings.

3. Demonstrate the improvements in 6D pose estimation accuracy achieved by applying super-resolution techniques.

This thesis is structured as follows:

- **Chapter 2** overviews relevant technologies and methodologies, including object detection, image super-resolution, and 6D pose estimation.
- **Chapter 3** discusses the specific networks chosen for this study, including GDR-Net for pose estimation and DRLN for super-resolution, and details the creation of the synthetic dataset.
- **Chapter 4** presents the experimental setup and results, focusing on how super-resolution improves pose estimation accuracy.
- **Chapter 5** concludes with a discussion of the findings and outlines potential directions for future research.

Chapter 2

Background

2.1 Robotic Picking

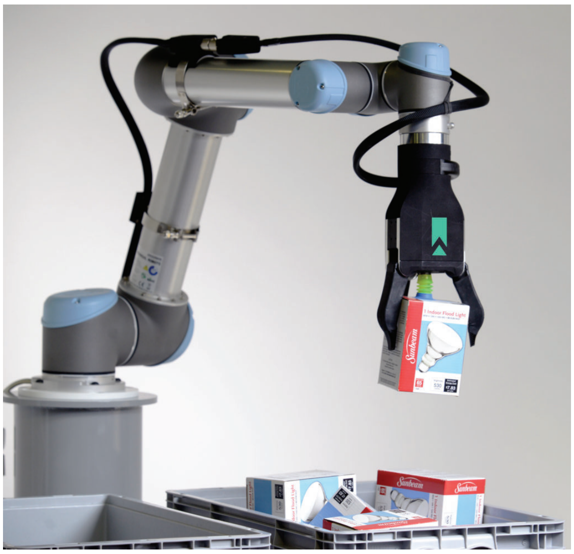


Figure 1: Robotic Picking

Source: [1]

Robotic picking, as shown in Figure 1, refers to automated systems that select and handle items from a storage area or production line, significantly enhancing operational efficiency, accuracy, and safety. These systems are a cornerstone in manufacturing, warehousing, and logistics, where speed and precision are critical. The combination of robotic arms, advanced vision systems, and machine learning algorithms allows robots to identify, grasp, and transport objects autonomously. As a result, robotic picking reduces labour costs, minimizes errors, and enhances safety by operating in environments that may be hazardous to human workers.

Various types of robotic picking systems are used in different industries, including [7]:

[1] Source: `https://www.therobotreport.com/grasp-sight-picking-evolve-robots/`

- **Cartesian Robots**: These robots move along linear paths in the X, Y, and Z directions, offering high positional accuracy and coverage over large workspaces. Their straightforward motion control makes them suitable for precise, repetitive tasks.
- **Delta Robots**: Known for their high speed and precision, delta robots are ideal for applications requiring rapid movements, such as assembly lines. However, they are typically limited to handling lighter loads.
- **Robotic Arms**: Versatile and capable of complex movements, robotic arms often have 5 or 6 axes of freedom, allowing for highly flexible operations. They are widely used across industries due to their adaptability to various tasks, from welding to packaging.
- **Collaborative Robots (Cobots)**: Designed to work alongside human operators, cobots are equipped with safety features that allow them to interact safely with people, making them ideal for environments where human-robot collaboration is necessary.

One of the key drivers behind the success of robotic picking systems is the integration of computer vision. Robots with vision systems can interpret their surroundings using cameras and sensors to identify, localize, and interact with objects in a dynamic environment. Computer vision improves the adaptability of robotic systems, allowing them to respond to changes in object appearance, positioning, or lighting conditions. The main benefits of integrating computer vision into robotic picking systems include:

- **Increased Accuracy**: Vision-guided robots can recognise and verify objects with high precision, reducing picking and handling errors.
- **Flexibility**: Robots can handle a variety of objects without requiring pre-programmed coordinates, making them more versatile in dynamic environments.
- **Efficiency**: Robots can quickly adapt to new products or changes in the layout of items, minimizing downtime and maximizing throughput.
- **Safety**: By accurately identifying and handling items, vision-equipped robots reduce the risk of damaging products or the robot itself.

Robotic picking is foundational in industrial automation, where precision, speed, and adaptability are crucial. Since this research focuses on enhancing object pose estimation for robotic systems, it is essential to understand the importance of robotic picking in modern applications. Practical 6D pose estimation directly impacts the success of robotic picking, especially in environments with clutter and occlusion, making it a critical factor in developing efficient and reliable automation systems.

2.2 Object Detection

Object detection is a fundamental task in computer vision, involving identifying objects within an image and localizing them through bounding boxes. Figure 2 illustrates the process of object detection, where the system detects objects in an image and marks them with bounding boxes for localization. Object detection is critical in applications such as autonomous driving, surveillance, and robotics, allowing systems to recognize and interact with the objects around them.

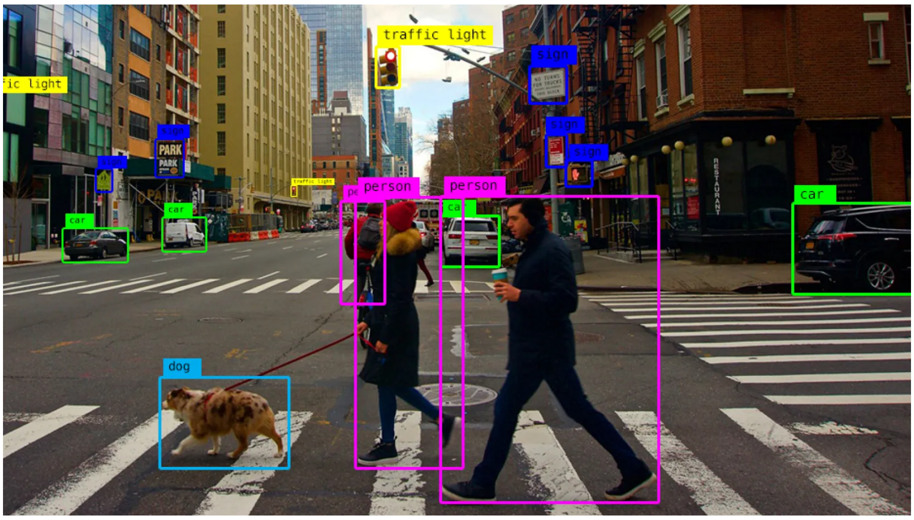


Figure 2: Object detection

Source: [2]

Core Concepts of Object Detection

- **Bounding Boxes**: These define the spatial location and size of objects within an image.
- **Class Labels**: The detected objects are classified into specific categories, such as 'person', 'car', or 'dog'.
- **Confidence Scores**: The confidence score represents the likelihood that the detected object belongs to a particular class.

Object Detection Methods

Object detection algorithms can be broadly divided into two categories:

- **Two-Stage Detectors**: These models, like R-CNN and its variants (Fast R-CNN, Faster R-CNN), generate region proposals for objects and then classify each region. While accurate, these methods can be computationally expensive.
- **Single-Stage Detectors**: Models like YOLO (You Only Look Once) [6] and SSD (Single Shot Multibox Detector) [5] predict bounding boxes and class probabilities in one pass, making them faster and suitable for real-time applications. YOLO, for example, frames object detection as a regression problem, allowing it to process images quickly while maintaining accuracy.

[2] Source: https://www.augmentedstartups.com/blog/how-to-implement-object-detection-using-deep-learning-a-step-by-step-guide

- **Transformer-Based Detectors**: Recently, transformer-based models, such as DETR (Detection Transformer), have emerged, simplifying the object detection pipeline by eliminating complex hand-crafted processes like non-maximum suppression.

Metrics for Evaluating Object Detection

Performance in object detection is measured using several key metrics:

- **Precision and Recall**: Precision measures how many of the detected objects are true positives, while recall measures the ability of the model to detect all relevant objects.
- **mAP (mean Average Precision)**: The mean of the Average Precision across all classes, evaluated at different Intersection over Union (IoU) thresholds. IoU measures the overlap between predicted and ground truth bounding boxes.

Challenges in Object Detection

- **Variability in Object Appearance**: Objects can vary significantly in appearance due to size, shape, orientation, or occlusion.
- **Environmental Conditions**: Lighting, shadows, and background clutter can affect the model's ability to detect objects accurately.
- **Overlapping Objects**: Detecting objects that are partially or fully overlapped with others remains a challenging task in dense environments.

Object detection is a prerequisite for 6D pose estimation, as identifying and localizing objects in an image is the first step in determining their pose. Without accurate object detection, the subsequent stages of pose estimation would be compromised. This section underscores the importance of reliable object detection methods, such as YOLO, which will be integrated into the proposed pipeline to improve the system's accuracy.

2.3 6D Pose Estimation

Six-dimensional (6D) pose estimation determines an object's position and orientation in a three-dimensional space relative to a reference coordinate system. This involves calculating the object's six degrees of freedom (DoF): three for translation (movement along the x, y, and z axes) and three for rotation (roll, pitch, and yaw around those axes). Figure 3 provides a visual representation of 6D pose estimation, which is essential in tasks requiring robots to interact with objects, such as picking, placing, or assembling parts.

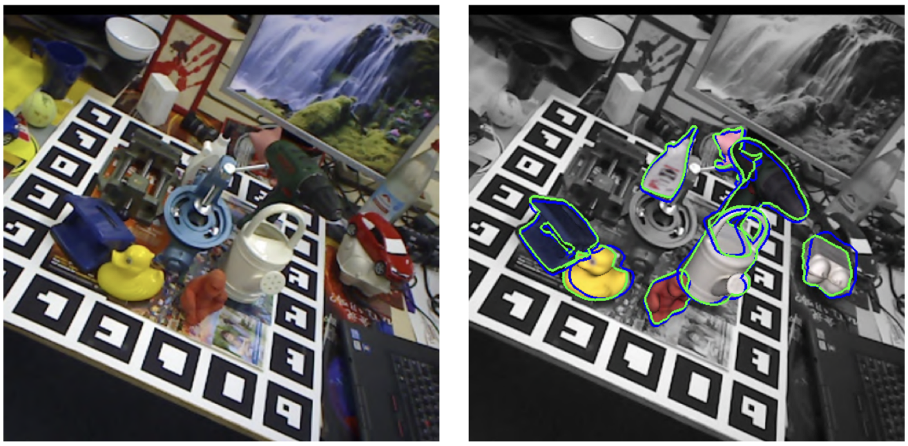


Figure 3: 6D Pose estimation

Source: [3]

Core Components of 6D Pose Estimation

- **Translation**: The object's position in 3D space, defined by the coordinates (x, y, z) relative to a reference point, such as a camera or the scene's origin.
- **Rotation**: Describes the object's orientation, often represented by Euler angles (roll, pitch, and yaw), rotation matrices, or quaternions, each offering different advantages for computation and interpretation.

Methods for 6D Pose Estimation

Several techniques have been developed to tackle 6D pose estimation, each with its advantages and limitations:

- **Template Matching**: This method compares current object views with pre-defined templates. It can be computationally efficient but struggles with occlusions and variations in viewpoint.
- **Feature-Based Methods**: These approaches detect and match key points from the object to a known model using feature descriptors like SIFT or SURF. Feature-based methods are sensitive to noise and object occlusion.
- **Deep Learning**: Convolutional neural networks (CNNs) have advanced 6D pose estimation by directly predicting pose from image data. These networks can be trained on large datasets to handle complex environments and diverse object shapes.
- **Data Fusion**: By combining RGB data with depth information (e.g., from LIDAR or stereo cameras), data fusion techniques improve robustness in challenging environments where relying solely on visual data may not suffice.

[3] Source: `https://arxiv.org/pdf/2102.12145`

Metrics for Evaluating 6D Pose Estimation

To evaluate the performance of 6D pose estimation methods, the following metrics are commonly used:

- **Translation Error**: The Euclidean distance between the estimated and true position of the object.
- **Rotation Error**: The angular difference between the predicted and actual orientation of the object.
- **ADD (Average Distance of Model Points)**: Measures average distance between corresponding points on the object's 3D model when it is transformed by both the estimated pose and the ground truth pose. A lower ADD value indicates a more accurate pose estimation.
- **ADD-S (Symmetric Objects)**: This variant is used for symmetric objects, where multiple pose configurations can be equally valid due to symmetry. For symmetric objects, instead of finding the closest points between corresponding locations, the ADD-S computes the smallest distance between each point in the transformed model and any of the points on the ground truth transformed model. This allows the metric to account for the ambiguity of symmetric shapes and avoids penalizing pose predictions that are correct but differ due to symmetry.

Challenges in 6D Pose Estimation

While 6D pose estimation has seen significant advancements, challenges remain:

- **Occlusion and Clutter**: In real-world scenes, objects may be partially obscured or surrounded by other objects, complicating the estimation process.
- **Lighting and Environmental Variations**: Changes in lighting and environmental conditions can affect the appearance of objects, making pose estimation more difficult.
- **Real-Time Constraints**: Many applications, such as robotic manipulation or augmented reality, require real-time pose estimation, which demands highly efficient and optimized algorithms.

6D pose estimation is critical for robotic manipulation tasks that require understanding the position and orientation of objects in three-dimensional space. The accuracy of pose estimation largely determines the performance of robotic systems in functions such as picking and placing. This section provides the foundation for the thesis, which seeks to improve pose estimation by integrating super-resolution techniques.

2.4 Image Super-Resolution

Image super-resolution (SR) enhances an image's resolution by increasing its detail and clarity. As depicted in Figure 4, super-resolution techniques can significantly improve the quality of low-resolution images, making them more usable in critical applications such as medical imaging, satellite imagery, and security surveillance. Super-resolution provides a powerful solution to improving image fidelity where high-quality visuals are required.

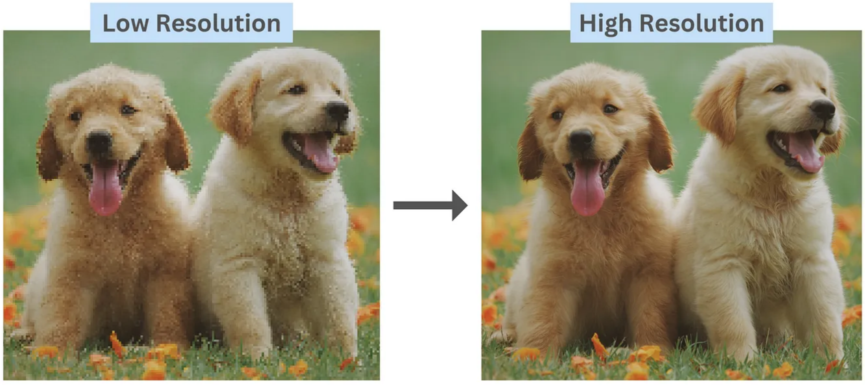


Figure 4: Image Super-Resolution

Source: [4]

Core Concepts of Super-Resolution

- **Resolution Enhancement**: This involves increasing the number of pixels in an image to improve its quality.
- **Upscaling vs Super-Resolution**: Traditional upscaling methods, like bilinear or bicubic interpolation, enlarge images but can result in blurring. Super-resolution, by contrast, uses sophisticated models to hallucinate high-frequency details, leading to sharper images.
- **Single vs. Multi-Image Super-Resolution**: Single-image super-resolution works on one image, while multi-image super-resolution combines multiple images to produce a high-resolution result.

Super-Resolution Methods

There are various approaches to improving image resolution:

- **Classical Methods**: Although computationally simple, techniques like bilinear or bicubic interpolation are often limited in preserving details.

[4] Source: `https://www.v7labs.com/blog/image-super-resolution-guide`

- **Learning-Based Methods**: Deep learning methods, particularly Convolutional Neural Networks (CNNs), have revolutionized super-resolution by learning complex mappings between low-resolution and high-resolution images. Models like SRCNN, VDSR, and ESRGAN have set new benchmarks for SR quality.

Metrics for Evaluating Super-Resolution

The effectiveness of super-resolution methods is evaluated using several metrics:

- **PSNR (Peak Signal-to-Noise Ratio)**: A widely used metric for quantifying the quality of a reconstructed image by measuring the ratio of signal to noise.
- **SSIM (Structural Similarity Index)**: Evaluates the perceived quality of an image by comparing structural information such as luminance, contrast, and texture with the ground truth.

Challenges in Super-Resolution

- **Computational Complexity**: High-resolution super-resolution, especially with deep learning models, is computationally intensive.
- **Artifacts**: Some super-resolution methods may introduce unwanted artefacts, such as blurring or halo effects.
- **Generalization**: Super-resolution models may struggle to generalize well to new datasets or real-world images if trained on narrow datasets.

The motivation for including image super-resolution lies in its ability to enhance the quality of visual data used in pose estimation. Poor image resolution can lead to degraded performance in object detection and pose estimation tasks. By applying super-resolution techniques, the goal is to improve the detail and clarity of input images, thereby boosting the accuracy of 6D pose estimation. This section provides the technical background for the DRLN network, which will be used to address the resolution challenges in the industrial dataset introduced later in the thesis.

2.5 ROS 2

Robotic Operating System 2 (ROS 2) is an open-source, flexible framework that has become an essential tool in developing modern robotic systems. Building on the success of its predecessor, ROS, ROS 2 offers enhanced features, improved scalability, and better support for real-time systems, making it a preferred choice for complex robotic applications. ROS 2 facilitates the development and integration of software components, enabling researchers and engineers to create robust robotic pipelines that can handle various tasks, from perception to action execution.

In robotic pipelines, ROS 2 provides the necessary tools to connect various computational nodes that perform specific tasks, allowing for seamless data flow between these nodes. This modular approach is highly beneficial in robotics, where different subsystems—such as perception, planning, and control—must work together harmoniously. The

ROS 2 ecosystem supports communication between nodes using topics, services, and actions, making it possible to design and implement intricate pipelines with a high degree of customization and control.

Advantages of Using ROS 2

ROS 2 provides several advantages for this robotic pipeline:

- **Modularity:** Each system component (e.g., object detection, pose estimation, robotic manipulation) is encapsulated in its node, allowing easy updates and maintenance.
- **Real-Time Performance:** ROS 2 is optimized for real-time applications, which is crucial for the success of the object detection and pose estimation pipeline.
- **Scalability:** ROS 2's design allows the system to scale quickly, meaning new functionalities (e.g., adding depth image processing) can be integrated with minimal disruption to the existing pipeline.

The ROS 2-based pipeline enables flexible, efficient integration of various computer vision and robotic manipulation tasks, allowing the system to operate in real-time environments, such as industrial automation scenarios.

Chapter 3

Methodology

3.1 Overview

This chapter presents the approach employed to address the challenges outlined in the previous chapters, explicitly focusing on enhancing 6D pose estimation by integrating super-resolution techniques. The process flow for the proposed solution is divided into several key stages: object detection, super-resolution, 6D pose estimation, and robotic picking. The overall workflow is depicted in Figure 5, illustrating how the system components interact.

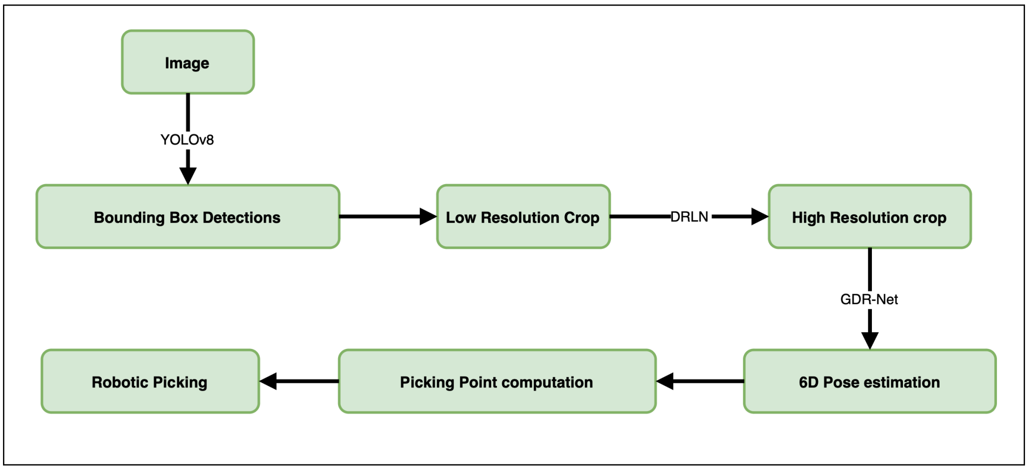


Figure 5: Pipeline Overview

3.1.1 Workflow Description

The workflow begins with object detection, followed by image super-resolution to enhance resolution and detail in the detected regions. These enhanced images are then passed to a 6D pose estimation network to determine the object's orientation and position in 3D space. Finally, the estimated pose information is used for robotic picking, guiding the robotic arm to manipulate the object accurately.

Key Stages of the Pipeline:

1. **Object Detection:** YOLOv8 detects and localizes objects in the scene, providing bounding boxes for each object.

2. **Super-Resolution:** Cropped objects produce low-resolution images that are enhanced using a super-resolution network (DRLN) to improve their quality.

3. **6D Pose Estimation:** The high-resolution images are fed into GDR-Net to estimate each object's 6D pose (position and orientation).

4. **Robotic Picking:** Using the 6D pose information, the robotic arm determines optimal picking points for accurate manipulation.

This structured workflow enhances image quality before pose estimation, improving accuracy in cluttered and complex environments.

3.2 Dataset

In the context of 6D pose estimation, the quality and diversity of datasets play a crucial role in developing and validating robust models. The performance of these models is highly dependent on the variability and realism of the training data. Conventional datasets, though useful, often lack sufficient diversity in terms of object poses, lighting conditions, and backgrounds, limiting the models' ability to generalize effectively to real-world scenarios.

To overcome these limitations, this thesis develops a highly realistic synthetic dataset tailored explicitly for 6D pose estimation for robotic applications. This dataset is a simplified version of SyGrid (Synthetically Generated realistic industrial dataset) and it is meticulously designed to support research in robotic manipulation, adhering to several essential criteria:

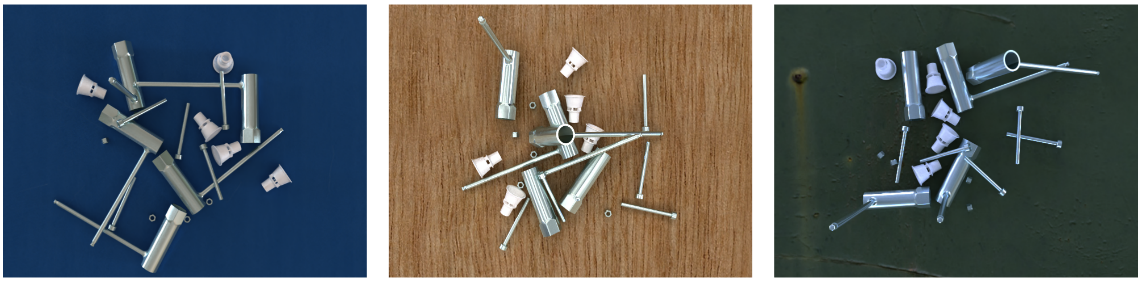
- High Realism: In applied robotics, where annotating real-world data is often costly, synthetic data generation presents a viable alternative. However, overcoming the domain shift between training and test data remains a significant challenge, as highlighted by [8]. Therefore, ensuring high realism in the generated scenes is critical in this work.

- Occlusion and Object Interaction: Objects in the scenes are not merely placed on a plane; they can interact, including overlapping. Such interactions are simulated in our environment, leading to 'Highly Cluttered' scenes.

- Suitability for Grasping: The selected objects are designed with shapes and sizes that make them suitable for manipulation by a wide range of robotic end effectors.

- Material Property Variability: Objects in the dataset exhibit various material properties, including reflective, metallic, transparent, and opaque surfaces. Unlike other datasets, our scenes contain a mix of these material types.

- Multi-object, Multi-instance Scenes: The scenes are typically populated with a variety of objects, averaging about 20 instances in total, with approximately five instances per object. In this context, an 'object' refers to the class or CAD model, while an 'instance' refers to the specific occurrence of that object within an image. Due to the nature of the physics simulation, some instances may occasionally end up outside the scene boundaries.



Blue plastic plane texture Wood plane texture Green metallic plane texture

Figure 9: Samples from the dataset with different backgrounds and lights

As shown in Figure 9, the dataset incorporates various textures, including plastic, wood, and metallic materials, along with diverse lighting conditions, enhancing the realism of the training data for 6D pose estimation.

Property	**Value**
Number of objects	4
Number of frames	2,000
Number of object instances	$\sim 40,000$
Image resolution	640x480 pixels

Table 3.1: Dataset Statistics

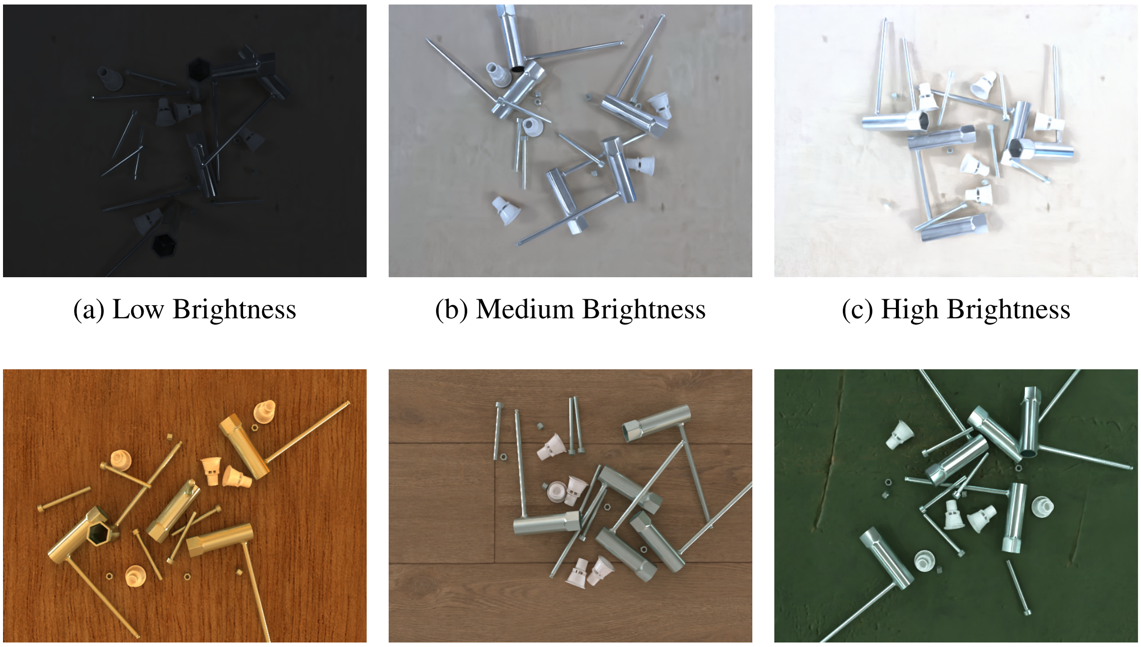
Table 3.1 provides an overview of the dataset, which consists of 4 distinct objects, 2,000 frames, and approximately 40,000 object instances, offering a substantial and diverse set of images for robust 6D pose estimation model training.

3.2.1 Dataset Versions

The dataset was developed in two distinct versions, each with its own set of variations:

- **Version 1:** This version was used for preliminary testing and includes a more straightforward setup with a single lighting environment, always the same plane texture, and lower rendering quality due to a less detailed rendering process. The lighting conditions in this version can be seen in Figures 10a, 10b, and 10c, which depict low, medium and high brightness levels, respectively. This limited variation in lighting and texture in Version 1 provided a controlled environment for initial testing.

- **Version 2:** The second version introduces significant improvements, including seven distinct lighting environments, seven different plane textures, and improved object rendering to simulate real-world industrial conditions better. Examples of these lighting conditions are shown in Figures 10d, 10e, and 10f, which depict warm light on a wood plane, natural light on a wood plane, and cold light on a metallic plane, respectively. This version aims to test models under more complex and varied conditions, making it more suitable for real-world applications.



(a) Low Brightness

(b) Medium Brightness

(c) High Brightness

(d) Warm Light, wood plane

(e) Natural Light, wood plane

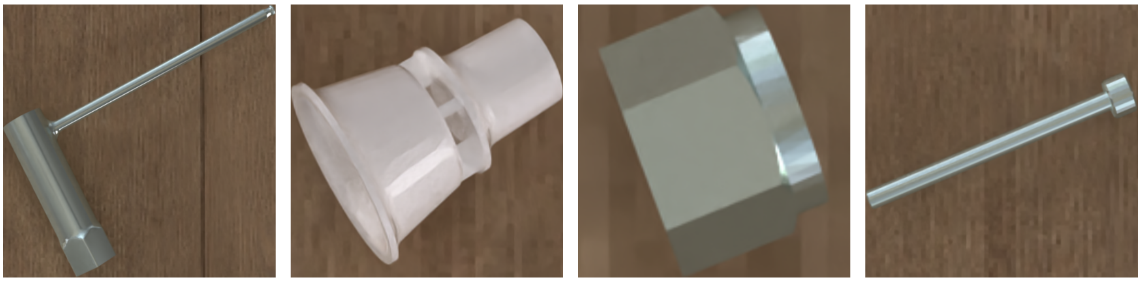
(f) Cold Light, metallic plane

Figure 10: Visual comparison between samples of the two dataset versions.

These two versions allowed the models to be tested progressively in more complex environments, leading to a better understanding of their robustness and limitations.

3.2.2 Set of Objects

The objects selected for this dataset are representative of those commonly found in industrial settings, chosen for their distinct features in terms of material composition, shape, and size. These characteristics present a challenging testbed for computer vision tasks. The dataset includes four distinct objects (shown in Figure 11), each described by an image, its name, and its material properties. Multiple instances of the same object can appear in a single image.



(a) Reflective metallic spark plug key (b) White plastic nozzle (c) Reflective metallic nut (d) Reflective metallic screw

Figure 11: Objects

3.2.3 Data Generation and Rendering

This dataset's RGB images and corresponding data were generated using a standalone physically-based rendering (PBR) application. This renderer accurately simulates global illumination, enabling realistic modelling of light interactions with materials with diffusive, reflective, refractive, and emissive properties. Light is modelled as rays emitted from the camera, with each ray passing through a pixel to interact with the scene's materials. When a ray strikes a material, its albedo is recorded, and a new ray is generated based on the material's properties. This process continues until the ray encounters an emissive material or reaches a pre-defined limit of bounces, with the accumulated colour representing a radiance sample. Since the rendering equation for physically-based rendering (PBR) is expressed as an integral that cannot be solved analytically [2], we approximate the solution using a Monte Carlo estimator.

In our implementation, a sample is accumulated at every frame for each pixel, and each final RGB image is generated by combining a predetermined number of frames. Due to the stochastic nature of Monte Carlo methods, the resulting images may exhibit noise, indicating that some pixels have not fully converged to the desired radiance. We applied an AI-based denoising pass to the final RGB images to reduce this noise.

Additionally, the renderer supports auxiliary buffers, which store arbitrary data for each pixel. These buffers contain a tuple with the object ID, instance ID, and depth value—essential for calculating visibility percentages and establishing labelling thresholds for computer vision tasks.

To generate realistic object positioning and clutter, the renderer is integrated with a custom rigid body simulation application [1], which ensures that objects are placed naturally by allowing them to fall and interact with one another, creating cluttered and occluded scenes. The renderer also generates masks for each object, allowing for visibility analysis and better training of 6D pose estimation models.

3.2.4 Dynamic Parameters of the Renderer

To ensure the dataset includes a wide range of realistic scenarios, several parameters were dynamically varied during the rendering process:

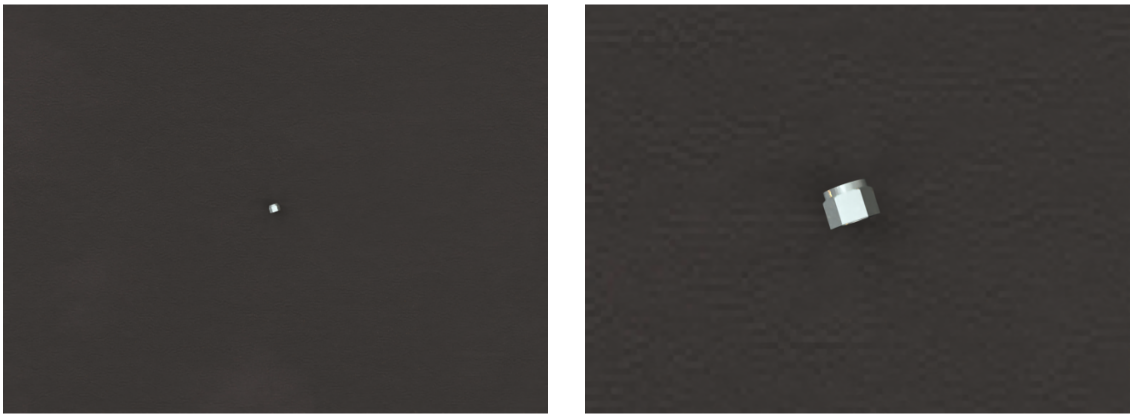
[1] https://pybullet.org/

- **Background and Lighting:** The environment is represented by HDR maps [2] that influence both background color and lighting. These maps were randomly selected from a set of 7 equirectangular HDR textures.
- **Camera Position:** The camera was positioned at a fixed distance of 450 mm from the plane to simulate the scenario where the robotic arm system uses vision data from this point to obtain information.
- **Object Starting Position:** Objects were randomly positioned above a static plane, allowing them to fall naturally and interact. The 6D poses were stored for use in the renderer.
- **Plane Texture:** The textures used for the plane on which objects were placed were dynamically varied, with random rotations applied to the texture to simulate different industrial scenarios.

3.2.5 Additional Dataset for Super-Resolution

Alongside the primary dataset used for 6D pose estimation, a supplementary dataset was explicitly created to train super-resolution networks. This additional dataset comprises image pairs, each capturing the same scene from varying camera distances. Each pair includes one high-detail image and one low-detail image generated by positioning the camera closer to or further from the object. This setup produces representations of the objects with differing levels of detail.

The dataset enables direct comparison between low- and high-resolution images, supporting the development of super-resolution models to improve the quality of lower-resolution images used in 6D pose estimation.



(a) Low Resolution

(b) High Resolution

Figure 12: Sample from the Super-Resolution dataset

The dynamic parameters of the renderer, such as lighting and object placement, remain consistent across both images in each pair, ensuring that the only significant variable is the camera distance. This consistency is crucial for isolating the effects of resolution on

[2] https://polyhaven.com/hdris

6D pose estimation and super-resolution tasks. By providing these paired images, the dataset supports a deeper exploration of how resolution impacts pose estimation accuracy and offers a robust foundation for training super-resolution models that can be applied in real-world robotic systems.

3.3 YOLOv8: Object Detection

YOLOv8 was chosen for object detection due to its efficiency in real-time applications and ability to handle object detection in cluttered and occluded environments. YOLOv8 builds on the success of previous YOLO versions by improving feature extraction and incorporating new architectural elements like CSP-PAN (Cross-Stage Partial Network with Path Aggregation), which enhances its performance.

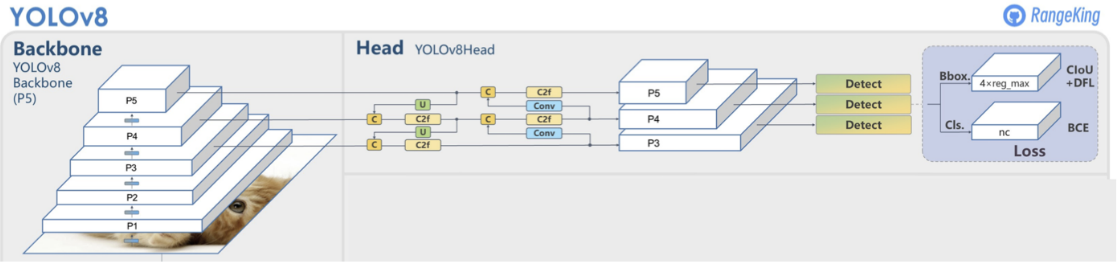


Figure 13: YOLOv8 Architecture

Source: [3]

3.3.1 YOLOv8 Architecture

The architecture of YOLOv8 (shown in Figure 13) is divided into three main components:

- **Backbone:** YOLOv8 utilizes a backbone derived from Darknet53, emphasizing feature extraction through smaller filter windows combined with residual connections for efficient feature extraction. This design facilitates cross-stage partial connections.
- **Neck:** It uses Spatial Pyramid Pooling (SPP) and Cross-Stage Partial Network (CSP-PAN) to refine the features extracted by the backbone and integrate them across different scales.
- **Head:** YOLOv8's prediction head comprises three specialized branches, each designed to predict object size and classification scores. It uses anchor boxes, which are predefined to correspond to anticipated object sizes, enhancing object detection accuracy.

[3] Source: https://blog.roboflow.com/whats-new-in-yolov8/

While retaining the foundational principles of YOLOv5, YOLOv8 surpasses it in several areas, including model architecture and feature integration. It offers fewer outliers in detection, demonstrating more reliable performance across diverse datasets.

For this thesis, the YOLOv8 model has been deployed in its large version to enhance capability and performance. It was trained using the previously extensively annotated dataset, addressing object detection and instance segmentation tasks. This dual application is essential as future developments in the project will require segmentation masks to further refine and enhance the model's utility in real-world scenarios.

3.4 GDR-Net: 6D Pose Estimation

The predominant strategy in monocular 6D pose estimation has been indirect methods, where the process typically involves two stages: first, establishing 2D-3D correspondences between image coordinates and the object's coordinate system, and then solving for the 6D pose using algorithms such as PnP (Perspective-n-Point) coupled with RANSAC for outlier rejection [9]. These methods have successfully produced accurate pose estimates; however, they are inherently not end-to-end trainable, posing a limitation for applications requiring differentiable poses [9]. Notable examples include works that leverage fixed control points or dense correspondences for robust estimation [9].

To address the limitations of indirect methods, direct regression approaches have been proposed, where the network directly predicts the 6D pose from image features. These methods benefit from being fully differentiable, making them suitable for tasks requiring seamless integration with other learning processes, such as self-supervised learning. However, direct methods have historically underperformed compared to their indirect counterparts, mainly due to challenges in accurately regressing the complex rotation and translation parameters [9].

3.4.1 GDR-Net Architecture

GDR-Net is an innovative architecture that integrates the strengths of both direct regression and correspondence-based methods for 6D object pose estimation. The primary goal of GDR-Net is to accurately predict objects' position and orientation (6D pose) in a scene using RGB images.

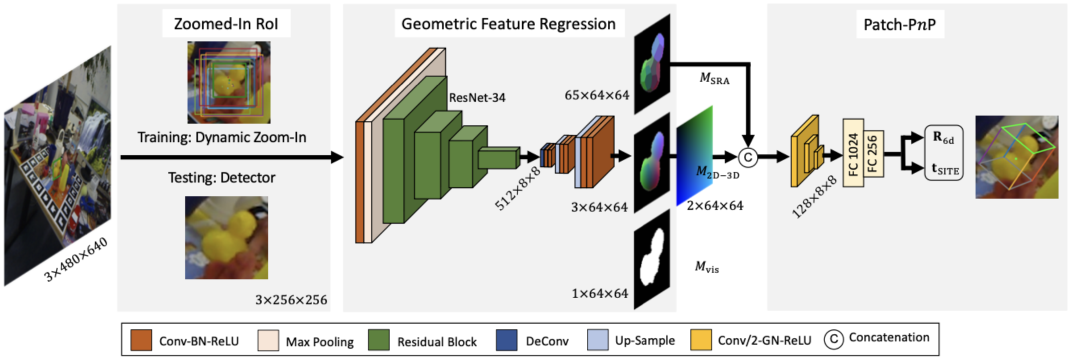


Figure 14: GDR-Net Architecture

Source: [4]

GDR-Net enhances pose estimation accuracy by leveraging intermediate geometric features—such as dense 2D-3D correspondences and surface region attention maps—while ensuring a fully differentiable framework. This architecture (shown in Figure 14) not only provides high-precision predictions but also allows for efficient end-to-end training, making it well-suited for real-time applications and tasks that require robust generalization across diverse object types and scenarios.

3.4.2 Object Detection and Input Processing

The GDR-Net pipeline begins with an external object detector that processes the input RGB image. This external detector is typically an object detection model, which identifies the localization of objects of interest within the image. The detector provides bounding boxes around the detected objects, acting as an initial indication of the object's location in the scene. These bounding boxes are then used to crop the relevant image regions, isolating the objects for further analysis.

In addition to detecting the object's location, the object detector also classifies the detected object, determining what specific object is inside each crop. This classification step helps the system identify the object type, enabling more targeted and relevant feature extraction for subsequent processing.

To ensure consistent input dimensions for the neural network, the cropped region containing the object is upsampled to a fixed resolution of **256x256 pixels**. This resizing prepares the object crops for processing by the pre-trained ResNet backbone, which extracts robust image features containing appearance and geometric information about the object.

[4] Source: https://arxiv.org/pdf/2102.12145

3.4.3 Feature Extraction Using ResNet

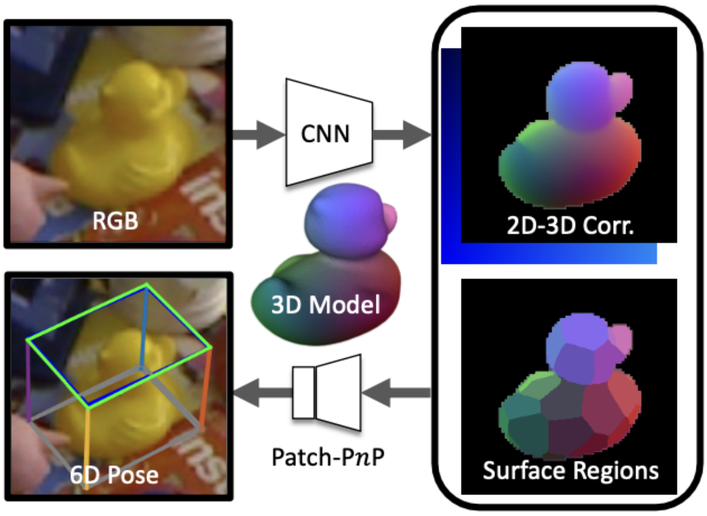


Figure 15: Process Overview

Source: [5]

Once the cropped image of the object is resized, it is passed through the ResNet-based feature extractor. ResNet, known for its ability to learn deep feature representations, plays a crucial role in GDR-Net by generating discriminative feature maps. These maps capture high-level semantic features and spatial details important for accurate pose estimation.

The features extracted by ResNet are further processed by intermediate geometric modules focusing on geometric reasoning. The most important of these geometric features is the **Dense 2D-3D Correspondence Map (M2D-3D)**, which links 2D image coordinates to the 3D model of the object.

3.4.4 Surface Region Attention Maps (MSRA)

In addition to dense 2D-3D correspondences, GDR-Net employs **Surface Region Attention Maps (MSRA)**. These attention maps focus the network's attention on specific surface regions of the object, guiding the prediction process toward the object's most relevant and informative parts. MSRA helps the network account for object symmetries and pose ambiguities by assigning importance to key surface areas.

3.4.5 Patch-PnP and Pose Regression

The core of GDR-Net is its ability to unify **direct regression** and **correspondence-based methods** for pose estimation. After processing the intermediate geometric features, the **Patch-PnP module** directly regresses the 6D pose of the object from the 2D-3D correspondences and surface region attention maps. This module consists of several convolu-

[5] Source: `https://arxiv.org/pdf/2102.12145`

tional and fully connected layers, which process the geometric features and predict both the 3D rotation and translation of the object.

3.4.6 End-to-End Differentiability and Training

A key advantage of GDR-Net is its **fully differentiable** architecture, which allows for seamless end-to-end training. The network is trained using a combination of pose estimation losses that supervise both the regression of the 6D pose and the prediction of intermediate geometric features. This end-to-end training ensures that the network learns visual and geometric information in a unified manner.

A **dynamic zoom-in (DZI)** strategy is employed during training. This technique randomly shifts and scales the bounding boxes to improve the model's robustness. This dynamic resizing ensures that the network generalizes well to varying object scales and positions in the image.

3.4.7 Real-Time Performance and Generalization

Given its streamlined architecture and reliance on direct regression for 6D pose estimation, GDR-Net is optimized for **real-time performance**. GDR-Net can generalise effectively across different scenes by leveraging geometric features like dense correspondences and surface region attention maps while maintaining high computational efficiency. This makes GDR-Net ideal for applications that demand both precision and responsiveness, such as robotics and industrial automation.

The effectiveness of GDR-Net is further underscored as it underpins the recent winner of the BOP (Benchmark for 6D Object Pose Estimation) Challenge [6]. The BOP Challenge is a rigorous benchmark designed to evaluate the accuracy and robustness of 6D object pose estimation methods across various datasets and scenarios. The GDR-Net-based approach that won the challenge outperformed other leading methods, demonstrating its superior ability to handle complex scenes, including those with occlusions, clutter, and varying lighting conditions.

3.5 DRLN: Super-Resolution Network

The advent of CNNs marked a significant shift in SISR (Single image super-resolution) techniques. One of the pioneering works in this domain was the Super-Resolution Convolutional Neural Network (SRCNN) proposed by [1], which introduced a three-layer CNN for image super-resolution. However, SRCNN laid the groundwork, and subsequent methods sought to improve performance by increasing network depth and introducing advanced architectures.

For instance, the VDSR (Very Deep Super-Resolution) [3] network incorporated residual learning to address the vanishing gradient problem, which is prevalent in deep networks. This was followed by the Enhanced Deep Super-Resolution (EDSR) network [4], which utilized a simpler but deeper architecture, significantly improving the state of the

[6] Source: `https://bop.felk.cvut.cz/leaderboards/`

art in SISR. These models, however, often required substantial computational resources, leading to a growing interest in developing more efficient architectures without sacrificing performance.

While deep networks like RCAN (Residual Channel Attention Network) [11] and EDSR have achieved state-of-the-art results, drawbacks include large model sizes and extensive training times. Additionally, most existing methods treat features at different scales equally, limiting their ability to adaptively capture and utilize information across various frequency bands.

3.5.1 DRLN Architecture

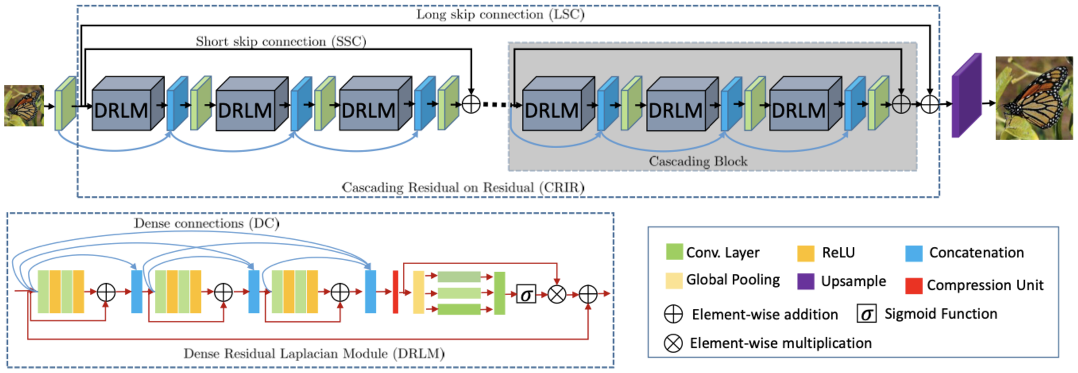


Figure 16: DRLN Architecture

Source: [7]

The **Densely Residual Laplacian Network (DRLN)** was developed to tackle the limitations found in traditional deep residual networks, particularly in the task of single-image super-resolution (SISR). As highlighted in Figure 16, DRLN utilizes a series of cascading residual blocks that are densely connected, ensuring efficient feature reuse across the network. This design enhances the model's ability to process low-frequency information while focusing on extracting mid- and high-frequency details—essential for recovering finer textures in low-resolution images.

DRLN adopts a residual-on-residual structure, employing short and long skip connections. This cascading structure promotes efficient training, prevents vanishing gradients, and helps the network learn richer representations across multiple levels of the image. Specifically, the dense connection between residual blocks allows previously extracted features to contribute to the learning process, increasing both accuracy and efficiency.

In terms of performance, the network balances depth and width by utilizing **Dense Residual Laplacian Modules (DRLMs)**. Each DRLM consists of densely connected residual units that increase the network's representative capacity while keeping the number of parameters in check. This structure, combined with a compression unit, mitigates computational overhead while ensuring that the network maintains a high level of detail reconstruction.

[7] Source: `https://arxiv.org/pdf/1906.12021`

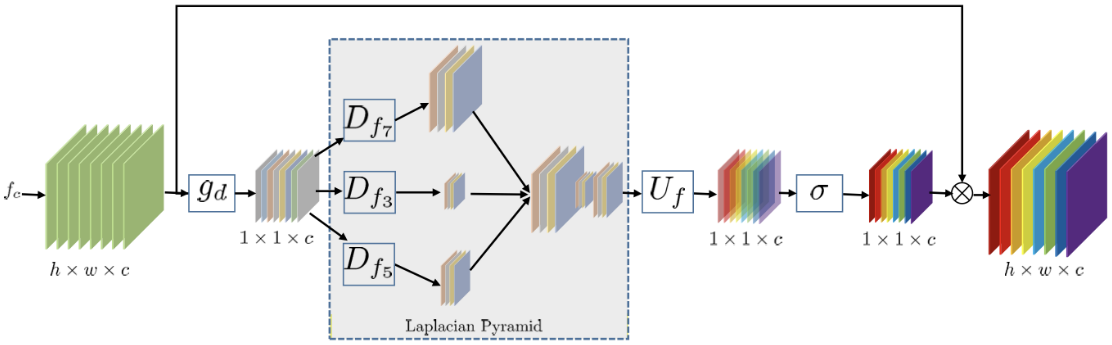


Figure 17: Laplacian Attention

Source: [8]

Moreover, as illustrated in Figure 17, the **Laplacian Attention Mechanism** plays a crucial role in DRLN. This mechanism enables the network to selectively emphasize important features at different frequency bands by leveraging multi-scale attention. It operates as a pyramid of sub-band attention layers that adaptively weigh features based on their frequency content. This helps handle complex textures and minute details within the image, making DRLN particularly effective at addressing the challenges posed by real-world super-resolution tasks.

Laplacian attention introduces a non-linear feature selection process that captures critical patterns while reducing less important information. This, in turn, enables the DRLN to preserve structural integrity and deliver sharper results compared to other super-resolution methods.

3.5.2 DRLN Performance in Super-Resolution

One of the main advantages of DRLN is its superior performance in **image super-resolution**, especially when dealing with high-frequency details such as textures and fine edges. The network achieves state-of-the-art results across multiple datasets and degradation types, including bicubic downsampling and blur-downscale images. As noted in various experiments conducted by the authors, DRLN outperforms other networks such as RCAN and EDSR in terms of both **Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index (SSIM)**, particularly in challenging datasets like **URBAN100** and **MANGA109**.

The DRLN architecture also shows significant robustness in **noisy environments**. By employing the multi-scale Laplacian attention mechanism, the network is capable of suppressing noise while recovering fine details, making it particularly effective for applications where real-world artefacts compromise image quality.

3.5.3 Motivation for Integrating DRLN in the Pipeline

Enhancing Input Quality

The main motivation for incorporating DRLN into the pipeline is to improve the quality of low-resolution input images before GDR-Net processes them for pose estimation. As

[8] Source: `https://arxiv.org/pdf/1906.12021`

outlined in 3.4.2, the object crop needs to be upsampled to a resolution of 256x256. However, traditional upsampling methods, such as bilinear interpolation, result in significant loss of detail, particularly for smaller objects. DRLN addresses this issue by enhancing the clarity and detail of the object crops, thereby increasing the accuracy of pose estimation, especially in scenarios where the input resolution is too low for GDR-Net to capture object features reliably.

Real-World Challenges Addressed

In real-world applications, the quality of image data can vary significantly due to factors such as varying lighting conditions, sensor noise, and the inherent limitations of imaging hardware. These variations can result in image crops that lack the necessary detail for accurate pose estimation. By incorporating DRLN into the pipeline, we can mitigate these issues, ensuring that the image crops retain the detail and quality needed for reliable object pose estimation by GDR-Net.

3.6 ROS2: Robotic pipeline

ROS 2 is used to implement a pipeline that integrates object detection, super-resolution, 6D pose estimation, and robotic manipulation into a single workflow. This modular approach allows efficient data exchange between nodes responsible for specific tasks. The following sections describe the individual components of the pipeline.

3.6.1 Pipeline Overview

The pipeline consists of multiple nodes, each designed to handle a specific function in the system. These nodes communicate with each other through ROS 2 topics, services, and actions, facilitating the flow of data from one stage to the next. Below are the key nodes and their roles in the pipeline:

Camera Node

The camera node is the entry point of the pipeline. It continuously captures RGB images of the scene and publishes them to a specific ROS 2 topic. This node provides the visual data needed for object detection and subsequent processing steps. In future enhancements, depth images could also be captured to refine pose estimation further, but they are unused in this current pipeline setup.

YOLOv8 Node

The YOLOv8 node subscribes to the camera node's output and performs object detection on the RGB images. The node outputs bounding boxes and segmentation masks, which are critical for isolating objects in cluttered environments. These outputs are published to a topic that the DRLN and GDR-Net nodes subscribe to, ensuring the seamless flow of data from detection to super-resolution and pose estimation.

DRLN Node

The DRLN node enhances the quality of the cropped images produced by the YOLOv8 node. By applying super-resolution techniques, this node ensures that the object crops have sufficient detail for accurate 6D pose estimation. The super-resolved images are then published to the GDR-Net node for pose estimation.

GDR-Net Node

The GDR-Net node is responsible for estimating the 6D pose of the objects based on the high-resolution images provided by the DRLN node. The pose is computed regarding position and orientation in three-dimensional space, making it suitable for robotic manipulation tasks. The estimated 6D poses are published to a topic that the Pose-to-Pick node subscribes to.

Pose-to-Pick Node

The final node in the pipeline is the Pose-to-Pick node, which takes the 6D pose data from GDR-Net and determines the optimal picking points for the robotic arm. This node considers the object's orientation, the robot's kinematic constraints, and the surrounding environment. Once the picking points are calculated, the node sends the appropriate commands to the robot for executing the pick-and-place task.

Chapter 4

Experimental Setup and Results

4.1 Introduction

This chapter explains the experimental design, implementation, and results of integrating super-resolution (SR) techniques with 6D pose estimation. The experiments are designed to evaluate the performance of the proposed pipeline across varying conditions, emphasizing improvements in pose estimation accuracy resulting from super-resolution models.

The experimental pipeline begins with object detection using YOLOv8, followed by super-resolution using the Densely Residual Laplacian Network (DRLN), and concludes with 6D pose estimation using GDR-Net. The following sections outline the dataset, model configurations, evaluation metrics used throughout the experiments, and a thorough analysis of the results.

4.2 Experimental Design

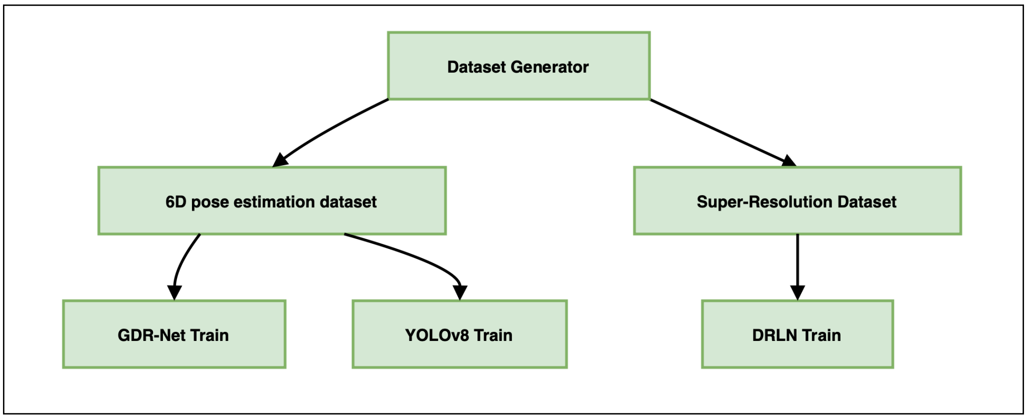


Figure 18: Training Overview

The pipeline combines several stages:

1. **Object Detection:** The YOLOv8 model is employed to detect objects within the scene. It identifies and localizes objects by generating bounding boxes for each object, which are utilized in subsequent processes.

2. **Super-Resolution:** The detected objects are extracted from the scene and passed through the DRLN model to enhance the resolution of the images. This resolution improvement is vital for refining image details, especially for objects with small bounding boxes.

3. **6D Pose Estimation:** The enhanced object crops are then passed into GDR-Net, which estimates the 6D pose of the objects, predicting both their translation and rotation within the 3D space.

We hypothesize that improving the resolution of the object crops will significantly enhance pose estimation performance. This is particularly important for small objects, where low-resolution bounding boxes can obscure critical details, potentially affecting the accuracy of the 6D pose estimation.

4.3 Datasets Overview

The experiments in this thesis utilize two datasets: a 6D pose estimation dataset and a super-resolution dataset. The specifics of these datasets, including object types, variations in lighting conditions, and the rendering process, were detailed in section 3.2.

4.3.1 6D Pose Estimation Dataset

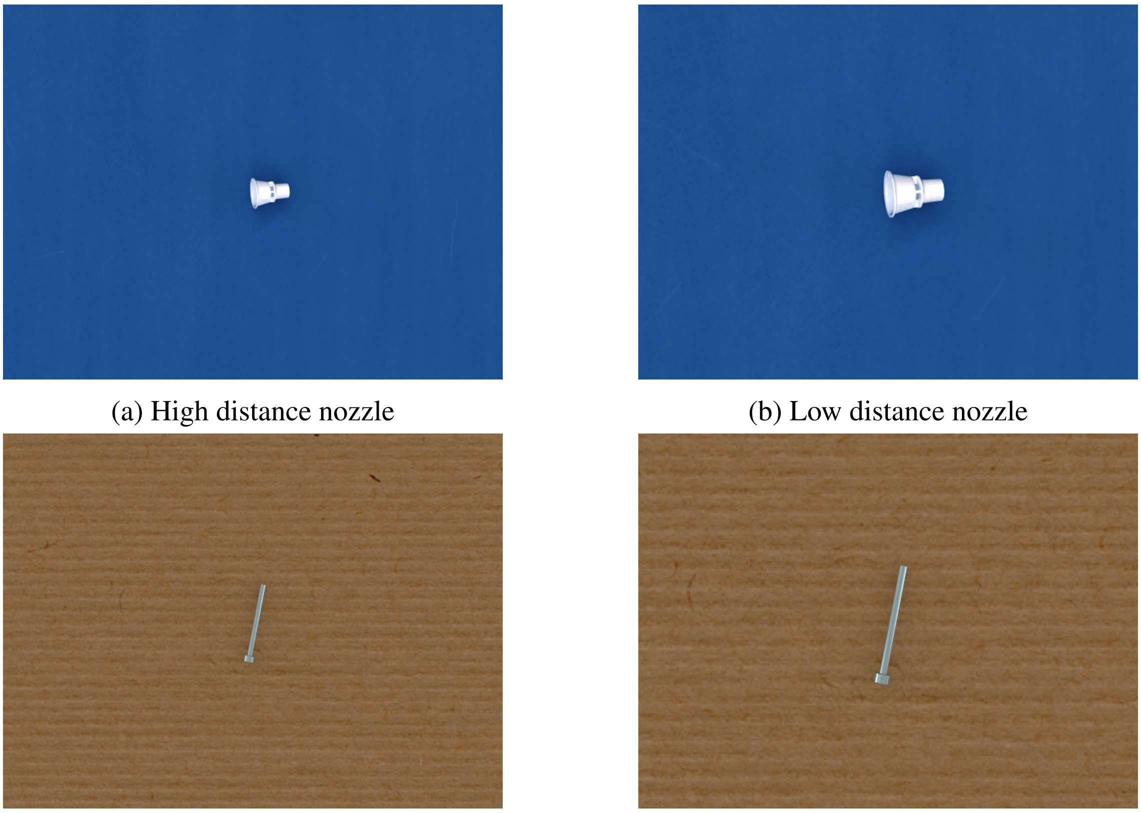
This dataset, tailored for 6D pose estimation in industrial settings, consists of synthetic images featuring multiple objects in cluttered scenes with diverse material properties. It includes various lighting environments and a mixture of occlusion levels to simulate real-world industrial conditions.

In the context of these experiments, this dataset was employed to train and evaluate both the YOLOv8 and GDR-Net models for 6D pose estimation. Since the dataset exists in two versions, it is important to clarify that Sections 4.6.2 and 4.6.3 utilize the first version, while the subsequent sections make use of the updated version.

4.3.2 Super-Resolution Dataset

The super-resolution dataset, described in Section 3.2.5, consists of paired low and high-resolution images generated from varying camera distances, as shown in Figure 19. This dataset was used to train the DRLN model to enhance the quality of low-resolution image crops before 6D pose estimation. The super-resolution step aimed to improve the pose estimation accuracy by enhancing visual detail, especially for small or occluded objects.

By applying these two datasets, the pipeline was tested for object detection, super-resolution, and pose estimation performance, particularly focusing on the impact of super-resolution in improving pose accuracy.



(a) High distance nozzle

(b) Low distance nozzle

(c) High distance screw

(d) Low distance screw

Figure 19: Visual hint on pairs of images for super-resolution training

4.4 Evaluation Metrics

The performance of the integrated pipeline was evaluated using different metrics for object detection, pose estimation and super-resolution.

- **Object Detection Metrics**:
  - **Precision**: Precision measures the proportion of true positives (TP) out of all predicted positives ($TP + FP$).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

  - **Recall**: Recall measures the proportion of true positives detected out of all actual positives ($TP + FN$).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

  - **Mean Average Precision (mAP)**: mAP is the mean of the average precision (AP) across all classes. AP is calculated by taking the area under the precision-recall curve for each class at different Intersection over Union (IoU) thresholds.

$$\text{AP} = \int_0^1 \text{Precision}(r)\, dr \tag{4.3}$$

where r is the recall. The mAP is the mean of the AP values over all classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{4.4}$$

where N is the total number of classes.

- **Pose Estimation Metrics**:
  - **Translation Error**: Measures the Euclidean distance between the predicted (T_p) and ground truth (T_{gt}) object positions in 3D space.

    $$\text{Translation Error} = \|T_p - T_{gt}\| \tag{4.5}$$

  - **Rotation Error**: The rotation error measures the angular difference between the predicted rotation matrix R_p and the ground truth rotation matrix R_{gt}. For rotation matrices, the error is computed as:

    $$\text{Rotation Error} = \arccos\left(\frac{\text{Tr}(R_p R_{gt}^{-1}) - 1}{2}\right) \tag{4.6}$$

    where Tr denotes the trace of the matrix.
  - **ADD (Average Distance of Model Points)**: ADD measures the average distance between corresponding points on the object model transformed by the predicted pose and the ground truth pose. This metric is used for non-symmetric objects.

    $$\text{ADD} = \frac{1}{|M|}\sum_{x \in M} \|(R_p x + T_p) - (R_{gt} x + T_{gt})\| \tag{4.7}$$

    where M is the set of object model points, R_p, T_p are the predicted rotation and translation, and R_{gt}, T_{gt} are the ground truth rotation and translation.
  - **ADD-S (Average Distance for Symmetric Objects)**: For symmetric objects, the ADD-S metric measures the average distance between model points transformed by the estimated pose and the closest corresponding points in the ground truth pose. This avoids issues caused by ambiguities in point correspondences for symmetric objects.

    $$\text{ADD-S} = \frac{1}{|M|}\sum_{x \in M} \min_{y \in M} \|(R_p x + T_p) - (R_{gt} y + T_{gt})\| \tag{4.8}$$

  - **Recall for ADD/ADD-S**: For the evaluation tables, the recall will be reported for the ADD or ADD-S metrics where the error is less than 10% of the object's diameter. This means the proportion of predictions with ADD or ADD-S errors below $0.1 \times d$, where d is the object's diameter, will be used to evaluate pose estimation performance.

    $$\text{Recall} = \frac{\text{Number of predictions with error } < 0.1 \times d}{\text{Total number of predictions}} \tag{4.9}$$

- **Super-Resolution Metrics**:
  - **PSNR (Peak Signal-to-Noise Ratio)**: PSNR quantifies the ratio between the maximum possible value of the signal and the power of the noise that affects the fidelity of its representation. Given two images I_{gt} (ground truth) and I_p (predicted), PSNR is defined as:

    $$\text{PSNR} = 10\log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{4.10}$$

    where MAX is the maximum possible pixel value and MSE is the mean squared error:

    $$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_{gt}(i,j) - I_p(i,j))^2 \tag{4.11}$$

  - **SSIM (Structural Similarity Index)**: SSIM measures the structural similarity between the ground truth image I_{gt} and the predicted image I_p in terms of luminance, contrast, and structure. It is defined as:

    $$\text{SSIM}(I_{gt}, I_p) = \frac{(2\mu_{gt}\mu_p + C_1)(2\sigma_{gt,p} + C_2)}{(\mu_{gt}^2 + \mu_p^2 + C_1)(\sigma_{gt}^2 + \sigma_p^2 + C_2)} \tag{4.12}$$

    where μ_{gt} and μ_p are the means of I_{gt} and I_p, σ_{gt} and σ_p are the variances, $\sigma_{gt,p}$ is the covariance, and C_1 and C_2 are small constants to stabilize the division.

4.5 Model Training and Optimization

The models were trained independently using two distinct datasets: the super-resolution dataset for training the DRLN model and the 6D pose estimation dataset for training YOLOv8 and GDR-Net.

4.5.1 YOLOv8 Object Detection Training

The YOLOv8 model was trained on a 6D pose estimation dataset to detect and localize these objects by predicting bounding boxes and segmentation masks. YOLOv8 underwent training over 100 epochs with the following configuration:

- **Optimizer**: Set to *auto*, allowing the model to select the most appropriate optimizer based on the configuration automatically.
- **Initial Learning Rate**: 0.01
- **Batch Size**: 16
- **Loss Functions** [10]:
  - **CIoU Loss** (Complete Intersection over Union) for bounding box regression, improving object localisation accuracy.

  - **DFL Loss** (Distribution Focal Loss) to refine object category predictions' refinement.
  - **VFL Loss** (Varifocal Loss) to handle class imbalance and uncertainties in the object classification task.

The results from training, including bounding box predictions and segmentation masks, can be seen in Figure 20, showcasing an example of YOLOv8's performance during training.

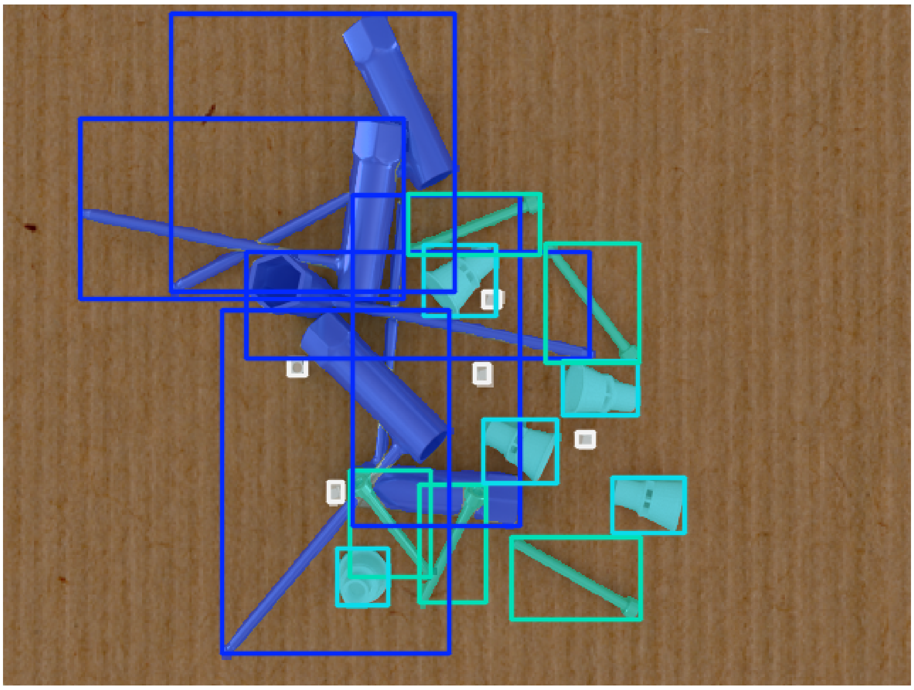


Figure 20: Examples of YOLOv8's predictions during training

4.5.2 DRLN Super-Resolution Training

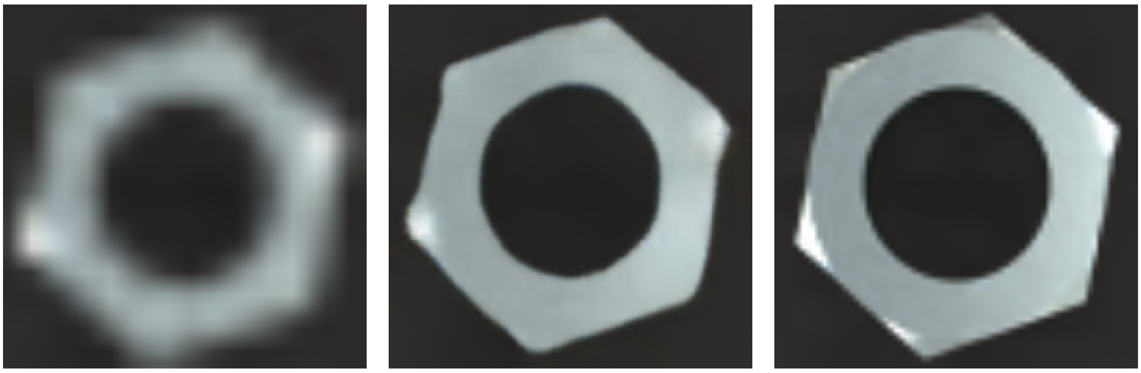
The DRLN was trained on the super-resolution dataset of paired low-resolution and high-resolution images. The training aimed to enhance low-resolution images by restoring high-frequency details and improving overall image quality.

The training process was conducted over 60 epochs with the following configuration:

- **Optimizer**: AdamW optimizer, chosen for its adaptive learning rate mechanism and decoupled weight decay, with an initial learning rate of 0.001.
- **Batch Size**: 8
- **Loss Functions**:
  - **L1 Loss**: A pixel-wise loss function that calculates the mean absolute error between the predicted super-resolved image and the ground truth high-resolution image. L1 loss is less sensitive to outliers than L2 loss, making it effective for maintaining pixel-level accuracy while preserving the overall structure of the image.

  - **Edge-aware Loss**: This loss function uses the Sobel operator to compute edge information, ensuring that the model focuses on enhancing the details along edges, where high-frequency details are typically lost. The network can produce sharper, more defined edges by incorporating edge-aware loss, leading to more visually appealing super-resolved images.

Figure 21 visually compares the rescaled input, the super-resolved image produced by DRLN, and the ground truth high-resolution image. The use of edge-aware loss, in particular, helps to refine edge details, leading to a significant improvement in perceptual quality.



(a) Rescaled input crop (b) Super-resolved crop (c) Ground truth crop

Figure 21: Visual comparison between traditional rescaling and Super-Resolution

4.5.3 GDR-Net Pose Estimation Training

GDR-Net was trained using a 6D pose estimation dataset, where the model was tasked with predicting the 6D pose of objects from images. The standard approach involves cropping the object using a ground-truth bounding box and then warping the crop to 256x256 pixels before predicting the pose. In a super-resolution-enhanced version, the crop is first upscaled using the DRLN model before being passed to GDR-Net for pose estimation.

The training was conducted over 160 epochs with the following configuration:

- **Optimizer**: Ranger optimizer combines RAdam with Lookahead and Gradient Centralization techniques, offering improved convergence and training stability [9].
- **Loss Functions**:
  - **Disentangled 6D Pose Loss** (L_{Pose}): This loss function supervises the model's prediction of three critical aspects of the object pose:
    - **Rotation Loss** (L_R): Ensures accurate orientation of the object.
    - **Center Loss** ($L_{\textbf{center}}$): Supervises the prediction of the scale-invariant 2D center of the object.
    - **Distance Loss** (L_z): Ensures accurate depth prediction by penalizing the difference between predicted and ground-truth object distances from the camera.

The overall 6D pose loss is given as:

$$L_{\text{Pose}} = L_R + L_{\text{center}} + L_z$$

- **Rotation Loss** (L_R): This term measures the error between the predicted rotation matrix $\hat{R}$ and the ground truth rotation matrix $\bar{R}$, where $x \in M$ represents the object's 3D points. The formula ensures that the rotation aligns the object correctly:

$$L_R = \frac{1}{|M|} \sum_{x \in M} \|\hat{R}x - \bar{R}x\|_1$$

- **Center Loss** (L_{center}): Supervises the 2D center coordinates of the object in the image plane. The loss penalizes deviations of the predicted center ($\hat{\delta}_x, \hat{\delta}_y$) from the ground-truth center ($\bar{\delta}_x, \bar{\delta}_y$) as follows:

$$L_{\text{center}} = \|\hat{\delta}_x - \bar{\delta}_x\|_1 + \|\hat{\delta}_y - \bar{\delta}_y\|_1$$

- **Distance Loss** (L_z): Supervises the object's depth (distance from the camera) by penalizing the difference between predicted depth $\hat{\delta}_z$ and ground truth depth $\bar{\delta}_z$:

$$L_z = \|\hat{\delta}_z - \bar{\delta}_z\|_1$$

For symmetric objects (where multiple orientations look the same), the rotation loss becomes **symmetry-aware**. The network compares the predicted rotation $\hat{R}$ with the closest symmetrical rotation in the ground-truth set $\bar{R}_{\text{sym}}$, ensuring the minimal loss:

$$L_{R,\text{sym}} = \min_{\bar{R} \in \bar{R}_{\text{sym}}} L_R(\hat{R}, \bar{R})$$

– **Geometric Loss** (L_{Geom}): This loss helps the model learn geometric features such as object visibility, depth, and surface region attention. The geometric loss is the sum of three terms:

$$L_{\text{Geom}} = L_{\text{depth}} + L_{\text{vis}} + L_{\text{SRA}}$$

Where:

- **Depth Map Supervision** (L_{depth}): Supervises the prediction of the 3D points on the visible surface of the object, penalizing the difference between predicted $\hat{M}_{XYZ}$ and ground-truth 3D coordinates $\bar{M}_{XYZ}$, weighted by a visibility mask $\bar{M}_{\text{vis}}$:

$$L_{\text{depth}} = \|\bar{M}_{\text{vis}} \cdot (\hat{M}_{XYZ} - \bar{M}_{XYZ})\|_1$$

- **Visibility Loss** (L_{vis}): Supervises the prediction of whether a part of the object is visible or not, comparing the predicted visibility map $\hat{M}_{\text{vis}}$ with the ground-truth visibility map $\bar{M}_{\text{vis}}$:

$$L_{\text{vis}} = \|\hat{M}_{\text{vis}} - \bar{M}_{\text{vis}}\|_1$$

- **Surface Region Attention Loss** (L_{SRA}): Supervises the attention map over surface regions, where $\hat{M}_{\text{SRA}}$ and $\bar{M}_{\text{SRA}}$ represent the predicted and ground-truth surface region attention maps, respectively. The cross-entropy loss is used to penalize incorrect predictions:

$$L_{\text{SRA}} = \text{CE}(\bar{M}_{\text{vis}} \cdot \hat{M}_{\text{SRA}}, \bar{M}_{\text{SRA}})$$

CE denotes the cross-entropy loss that encourages the model to focus on surface regions important for pose estimation.

The total loss for training GDR-Net is the sum of the disentangled 6D pose loss and the geometric loss:

$$L_{\text{GDR-Net}} = L_{\text{Pose}} + L_{\text{Geom}}$$

This ensures that the model learns both accurate 6D pose estimation and detailed geometric features during training.

4.6 Sequence of Experiments

This section outlines the experiments conducted to evaluate the models' performance and analyze the effect of incorporating super-resolution into the workflow. Below is a concise summary of the experiments:

1. Evaluation of GDR-Net on standard literature datasets
2. Evaluation of GDR-Net on the 6D pose dataset (version 1)
3. Evaluation of GDR-Net on the 6D pose dataset (version 1) with additional modifications
4. Evaluation of YOLOv8 on the 6D pose dataset (version 2)
5. Testing DRLN and other super-resolution networks on a super-resolution dataset
6. Evaluation of GDR-Net on the 6D pose dataset (version 2)
7. Evaluation of GDR-Net in combination with DRLN (frozen) on the 6D pose dataset (version 2)
8. Real-world testing of the models

4.6.1 Testing GDR-Net on Literature Datasets

The first step involved testing GDR-Net alone using a dataset from the literature to validate its performance and ensure that it behaves as expected under controlled conditions. The model was evaluated on LineMod (LM) and LineMod Occluded (LMO). These tests were conducted across all objects. The results closely matched those reported in the original paper, confirming the reliability of the architecture.

Table 4.1 presents the model's results when retrained with different batch sizes and random seeds on the LineMod dataset. We observe that the ADD(S) (computed as shown by 4.9) metric remains consistently high, close to the 93.69% reported in the paper, regardless of the batch size or seed used, indicating that the model's pose estimation performance is stable. The rotation error (RE) and translation error (TE) also align with expectations, with negligible differences across the variations. Notably, the model achieved slightly improved results with a batch size of 24 and seed 0, showing a higher ADD(S) of 93.92% and a lower RE of 1.96°.

Similarly, Table 4.2 displays the results for the LineMod Occluded dataset. While the performance is naturally lower due to occlusions, the ADD(S) values remain robust and close to the reported 56.14%. The RE and TE metrics show no significant deviations, confirming the model's ability to handle partial occlusions effectively.

	Paper	Bs 24, seed 42	Bs 48, seed rnd	Bs 24, seed rnd	Bs 24, seed 0
ADD(S) ↑	93.69	93.64	93.23	93.77	93.92
RE ↓ (°)	1.97	1.99	2.09	1.96	2.00
TE ↓ (m)	0.01	0.01	0.01	0.01	0.01

Table 4.1: Results on Linemod

	Paper	Bs 24, seed rnd	Bs 48, seed 42
ADD(S) ↑	56.14	55.47	56.09
RE ↓ (°)	12.41	12.51	12.46
TE ↓ (m)	0.05	0.05	0.05

Table 4.2: Results on Linemod Occluded

Figure 22 visually compares the recall for rotation and translation errors, focusing on errors below 2% of the object's diameter. Specifically, the figure highlights four key metrics:

- **ad_10**: Represents the ADD(S) metric, as seen in the table above. This measures the accuracy of pose estimation, and errors below this threshold are considered successful predictions.

- **re_2**: Captures the rotation error (RE) and reflects the percentage of poses for which the rotation error is below 2°. This metric assesses how well the model predicts the object's orientation.

- **te_2**: Evaluates the translation error (TE), reporting the percentage of poses with translation errors below 2% of the object's diameter, offering a measure of positional accuracy.

- **rete_2**: A combined metric that reports the percentage of poses where the rotation and translation errors (RE and TE) are below the aforementioned thresholds. This metric offers insight into the overall quality of pose estimation.

These metrics provide a more stringent evaluation of the model's precision, focusing on scenarios where rotation and translation must be extremely accurate. The consistent performance of GDR-Net across these stringent conditions further validates its robustness and reliability across different configurations.

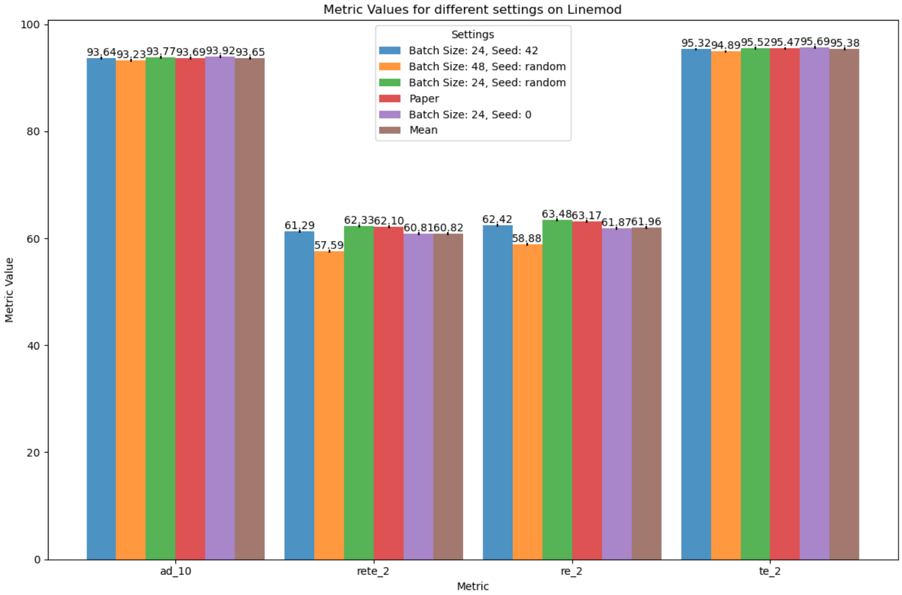


Figure 22: Linemod results

4.6.2 Testing GDR-Net on the First 6D Pose Estimation Dataset

After validating GDR-Net on Linemod and Linemod Occluded, we tested it on the 6D pose estimation dataset version 1 to observe its performance in a more challenging scenario. As shown in Table 4.3, GDR-Net achieved strong results for objects such as the nozzle and screw when provided with ground-truth bounding boxes.

	Spark Plug Key	Nut	Nozzle	Screw	Average
ADD(S) ↑	75	13.89	94.87	92.5	69.07
RE ↓ (°)	18.31	23.67	2.55	3.03	11.89
TE ↓(m)	0.01	0.02	0.00	0.00	0.01

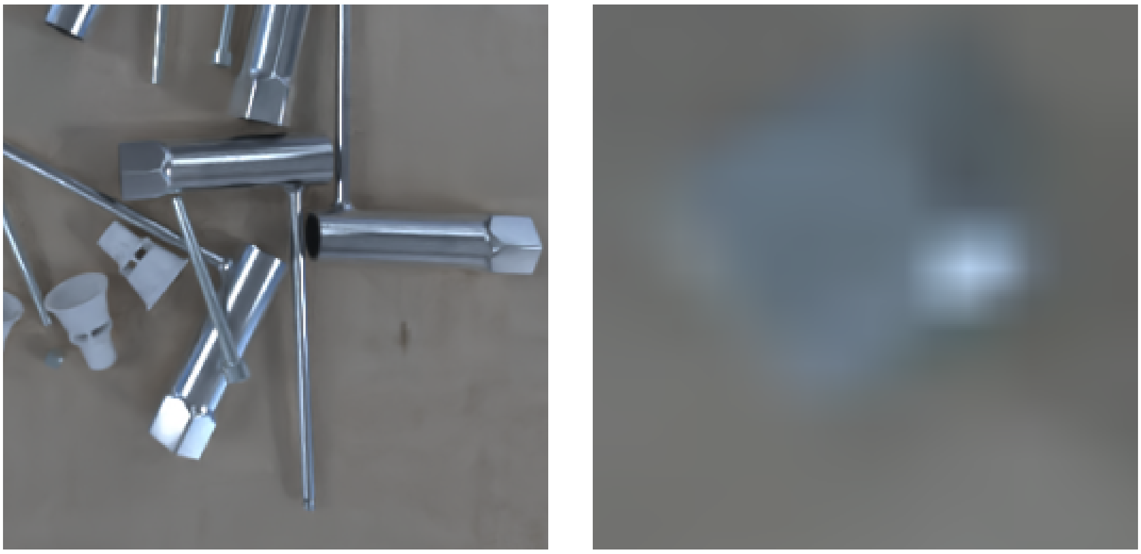
Table 4.3: Results on the first 6D dataset

The model's nut and spark plug key performance exhibited a significant decline, leading to a more in-depth analysis. Through this investigation, we identified two key factors contributing to these suboptimal results:

- The spark plug key is relatively large, and in cluttered environments, its cropped image often contains parts of other objects or multiple instances of the same object in different orientations, likely confusing the model.

- For the nut, the simulated camera in the dataset is positioned 45 cm away, causing the nut to appear too small in the cropped image. When upsampled, the image becomes challenging for the model to accurately distinguish due to the loss of detail.

Figure 23 illustrates examples of these challenges, where regions of interest (ROIs) were augmented with scaling and shifting to simulate potential object detection errors during testing. For further details on the ROI augmentation process, please refer to Section 3.4.6



(a) Augmented ROI for spark plug key

(b) Augmented ROI for nut

Figure 23: Rescaled Region of Interest used for training GDR-Net

4.6.3 Evaluating GDR-Net on Different Dataset Variations

To validate our hypotheses and improve model performance, we conducted several experiments using variations of the dataset:

- A dataset containing only one instance of each object per image to tackle the spark plug key challenge.
- A dataset focused exclusively on nuts, captured with a closer camera to produce more detailed ROIs.
- Tests on the original dataset with different approaches to ROI generation.

Single Instance per Image Dataset

In this experiment, we ensured that each image contained only one object instance (see Figure 24). This setup eliminated the confusion caused by multiple object poses within the same ROI. The results, summarized in Table 4.6, show a significant performance improvement for the spark plug key.

	Spark Plug Key	Nut	Nozzle	Screw	Average
ADD(S) ↑	95	25	96.88	94.2	77.77
RE ↓ (°)	1.83	7.18	0.99	2.05	3.01
TE ↓ (m)	0.00	0.01	0.00	0.00	0.00

Table 4.4: Results on single-instance dataset

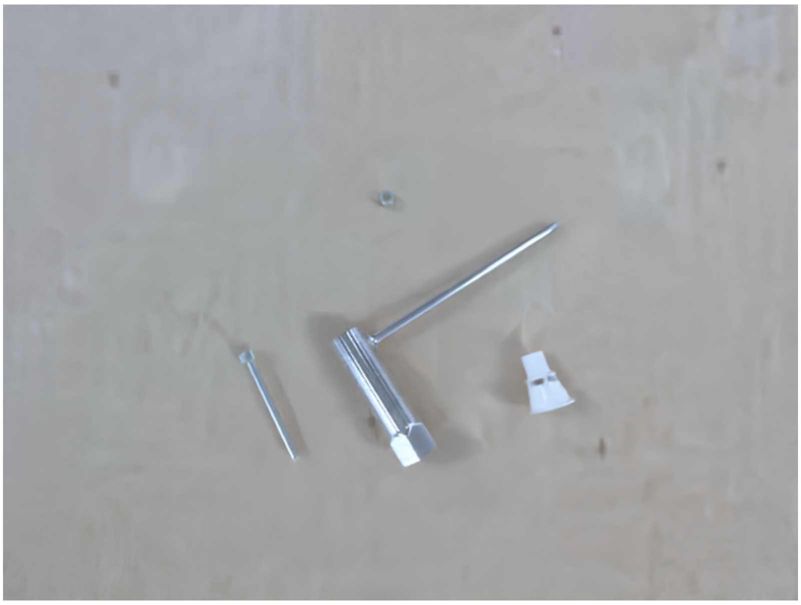


Figure 24: Sample from the single instance dataset

Nut-Focused Dataset

Next, we created a dataset with images exclusively featuring nuts (see Figure 25), with the camera positioned closer to improve the level of detail in the ROIs. The results in Table 4.5 reveal a significant performance boost, suggesting that the previous subpar results were due to the greater camera distance.

	Nut
ADD(S) ↑	97.95
RE ↓ (°)	2.07
TE ↓ (m)	0.00

Table 4.5: Results for the nuts dataset

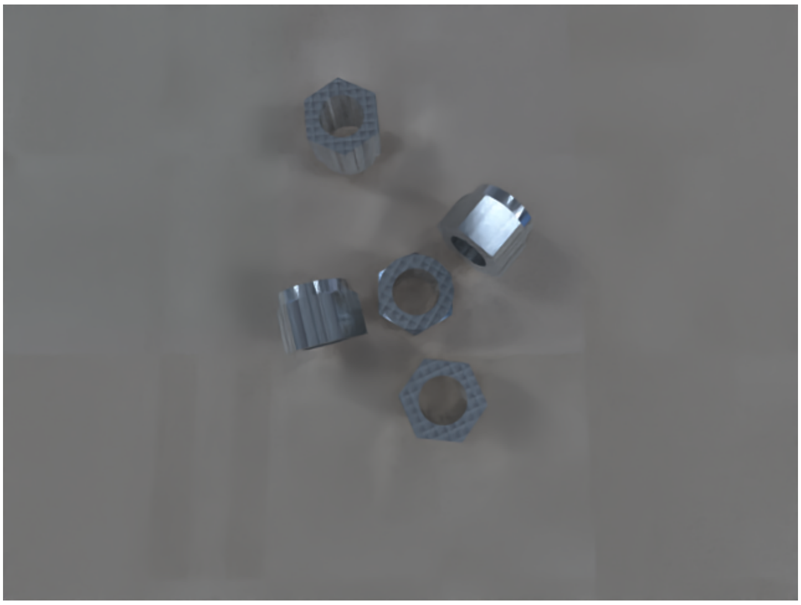


Figure 25: Sample from the nuts dataset

Based on these findings, we explored the potential of integrating super-resolution techniques to enhance the quality of ROIs, which could further improve performance.

ROI Variations

Finally, we experimented with different ROI modification techniques, including:

- Removing ROI augmentation and directly using rescaled bounding boxes.
- Padding the ROIs with black or random backgrounds.
- Segmenting the object so that only the object appears within the ROI.

The most significant improvements were observed when using object segmentation, particularly for the spark plug key, as it reduced interference from other objects in the ROI. However, segmentation led to worse performance for the nuts, as the small number of visible pixels after rescaling made it difficult to generate a recognizable object.

	Spark Plug Key	Nut	Nozzle	Screw	Average
ADD(S) ↑	98.15	12	98	96	76.03
RE ↓ (°)	4.62	20.08	2.63	3.28	7.68
TE ↓ (m)	0.00	0.01	0.00	0.00	0.01

Table 4.6: Results on the 6D dataset with segmentation

4.6.4 Testing the YOLOv8 on the New 6D Pose estimation dataset

The YOLOv8 model achieved an overall mean Average Precision (mAP) of 90.4% across all objects in the 6D pose estimation dataset. Table 4.7 summarizes individual objects' precision and recall rates.

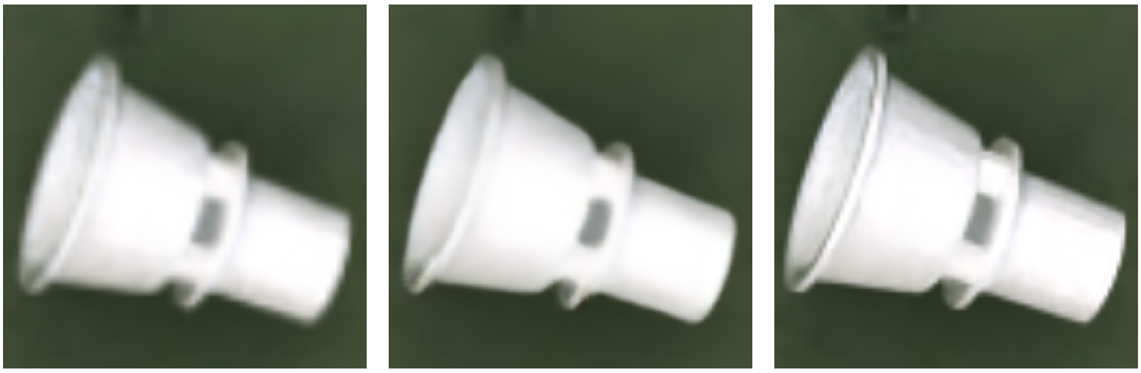
Object Class	Precision ↑	Recall ↑	mAP50 ↑	mAP50-95 ↑
Spark Plug Key	0.99	0.977	0.994	0.977
Nozzle	0.998	0.999	0.995	0.965
Nut	0.993	0.963	0.989	0.733
Screw	0.992	0.985	0.994	0.941

Table 4.7: Object detection results on the 6D pose estimation dataset.

The model performs well overall, though the nut proves more challenging due to its small size, making it difficult to predict its bounding box with high precision.

4.6.5 Testing Different Super-Resolution Networks

To verify DRLN as the optimal choice for super-resolution, we tested various networks (SRCNN [1], EDSR [4], VDSR [3]) on the new dataset. DRLN proved to be the most suitable lightweight model, particularly for small objects like the nut, whose crop size is as small as 32x32 pixels.



(a) Rescaled input crop

(b) Super-resolved crop

(c) Ground truth crop

Figure 26: Visual comparison of super-resolution effects on object crops.

We trained individual DRLN networks for each object to optimize performance based on object size. The final upscaled image size was 256x256 pixels. The following configurations were used for different objects:

- 8× for the nut (input rescaled to 32x32, output upscaled to 256x256)
- 4× for the nozzle (input rescaled to 64x64)
- 2× for the screw and spark plug key (input rescaled to 128x128)

Integrating DRLN improved the image quality of low-resolution object crops in the super-resolution dataset. Table 4.8 presents the PSNR and SSIM values before and after applying DRLN, highlighting the improvement in image quality. It is worth noting that for objects with bounding boxes larger than 128x128, like the spark plug key, the improvement from super-resolution was less significant due to the loss of information during compression and expansion.

Object Class	Without DRLN		With DRLN	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Spark Plug Key	26.82	0.813	26.50	0.807
Nozzle	25.63	0.870	26.29	0.892
Nut	22.16	0.829	25.40	0.892
Screw	29.87	0.880	31.27	0.916

Table 4.8: Super-resolution results on the 6D pose estimation dataset using DRLN.

4.6.6 Testing GDR-Net on the New 6D Pose Estimation Dataset

GDR-Net was retrained after adjusting the dataset with varying lighting conditions and backgrounds and improving rendering. Given the increased complexity, we expected results comparable to or slightly worse than previous datasets.

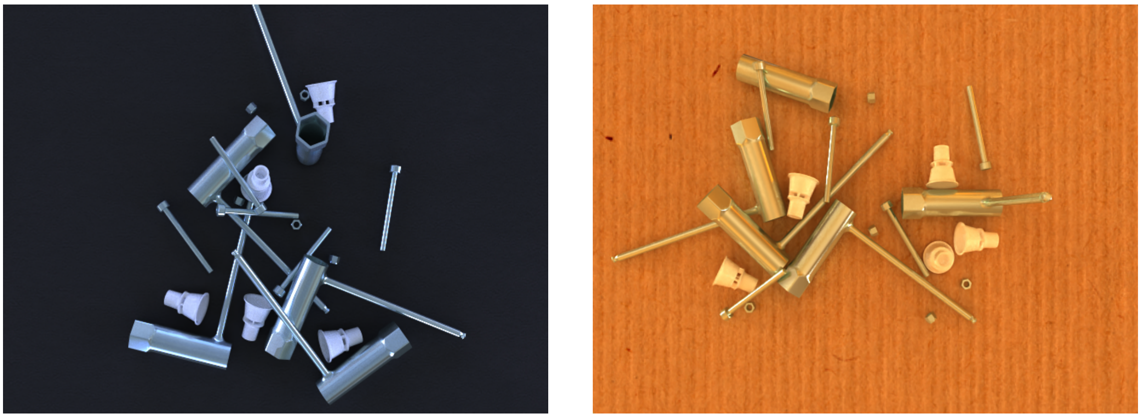


Figure 27: Samples from the new 6D pose estimation dataset.

Table 4.9 shows that GDR-Net handled the more complex dataset reasonably well, producing results comparable to earlier tests. However, some minor degradation in performance was observed due to the increased complexity of the scenes.

4.6.7 Integrating DRLN with GDR-Net for Enhanced Pose Estimation

We integrated DRLN into the GDR-Net pipeline to assess whether super-resolution could improve 6D pose estimation. Encouragingly, as shown by Table 4.9, the results showed improved translation and rotation errors and an increase in the ADD(S) metric. The improvements indicate that enhancing image quality through super-resolution leads to better feature extraction and more accurate pose estimation.

However, the performance for specific objects, such as the nut, remained suboptimal. This is likely due to artefacts introduced by super-resolution, which may confuse the model. More advanced super-resolution techniques could potentially address these issues and further improve accuracy.

Object Class	**Without DRLN**			**With DRLN**		
	ADD(S) ↑	**RE** ↓ (°)	**TE** ↓ (m)	**ADD(S)** ↑	**RE** ↓ (°)	**TE** ↓ (m)
Spark Plug Key	95.69	7.72	0.00	94.80	8.03	0.00
Nozzle	94.60	2.31	0.00	97.40	1.82	0.00
Nut	17.32	3.96	0.01	23.63	3.73	0.01
Screw	90.88	2.96	0.01	97.75	3.22	0.00

Table 4.9: Comparison of 6D pose estimation results with and without DRLN.

The results highlight a clear performance improvement when DRLN is used, particularly in the ADD(S) scores for objects like the nut (6-point improvement) and the screw (7-point improvement). Although the nut's performance remains low, the addition of more sophisticated super-resolution models could lead to further enhancements in future experiments.

4.6.8 Ablation Study

The table below summarizes the performance of the pipeline with and without super-resolution integration. The integration of DRLN reduced translation and rotation errors, with an overall improvement in the ADD(S) score.

Model Configuration	**TE** ↓ (m)	**RE** ↓ (°)	**ADD(S)** ↑
YOLOv8 + GDR-Net	0.005	4.24°	74.62
YOLOv8 + DRLN + GDR-Net	0.0025	4.20°	78.40

Table 4.10: Average results comparing pipeline components.

4.7 Real-World Testing

The final model was evaluated in a real-world robotic picking task. In this test, various objects were placed randomly in cluttered environments to simulate a realistic industrial scenario. The objects consisted of tools and components of different shapes and sizes, including spark plug keys, screws, nozzles, and nuts. These were positioned with overlapping and occlusions to assess the system's robustness.

The robotic system utilized a **UR5e collaborative robot (cobot)** arm, which was controlled through multiple ROS2 (Robot Operating System) nodes. Figure 28 illustrates different bounding box annotations for detected objects, including their classification and confidence scores, facilitating the grasp planning for the robotic arm. Figure 28 is divided into three sub-images:

- The **first sub-image** shows the output from the YOLOv8 object detection algorithm. Each object is enclosed in a bounding box, with different colours representing distinct object classes, such as screws, nozzles, and spark plug keys. The confidence scores for each prediction are also displayed.

- The **second sub-image** highlights the region of interest (ROI) chosen for further processing by DRLN, which refines the quality of the ROIs.

- The **third sub-image** illustrates the 6D pose estimation, where 3D bounding boxes represent the orientation and position of objects in space. Upon closer inspection, it is evident that most of the boxes are accurately positioned, while a few appear slightly misaligned. This misalignment is likely due to the transition from synthetic to real images.

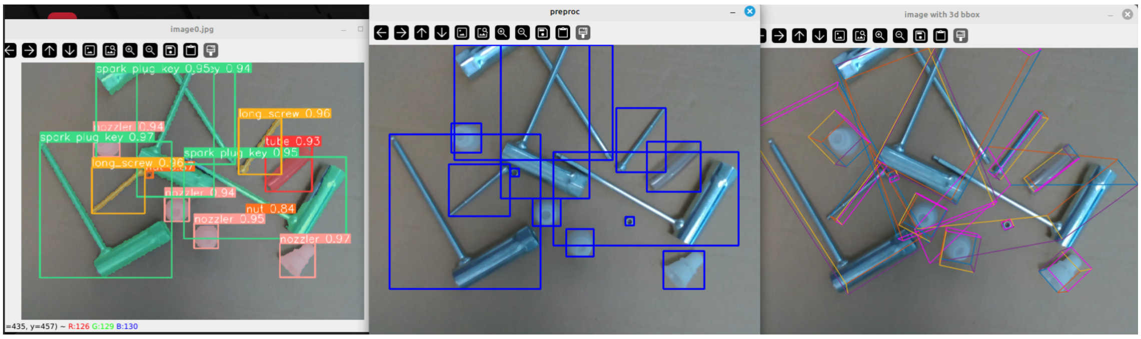


Figure 28: Example of ROS pipeline during real-world.

Chapter 5

Conclusions

This thesis investigated the integration of super-resolution techniques within a 6D pose estimation pipeline, leveraging state-of-the-art neural networks such as YOLOv8 and GDR-Net, with DRLN employed for super-resolution enhancement. The research demonstrates significant improvements in pose estimation accuracy, particularly for small to medium-sized objects and in complex, cluttered environments where high-resolution detail is essential for precise object recognition and pose determination.

5.1 Summary of Key Findings

- The use of DRLN super-resolution markedly improved the clarity and detail of image crops used for pose estimation, resulting in more accurate pose predictions.
- Incorporating super-resolution into the pose estimation pipeline led to measurable accuracy gains, particularly for objects that previously suffered from low resolution due to their size or distance from the camera.
- Real-world testing validated the model's robustness across a variety of scenarios, confirming the efficacy of the experimental results achieved in controlled environments.

5.2 Implications of the Work

The results of this research highlight the potential of combining super-resolution and pose estimation technologies to enhance robotic vision systems, especially in industrial settings where precision and reliability are critical. This integration could lead to more advanced robotic systems capable of performing detailed and nuanced object interactions in real-time.

5.3 Limitations and Challenges

- The performance improvements, while notable, varied between different object types, suggesting the need for object-specific tuning of the super-resolution models or limiting its application to manually specified objects.

- In real-world applications, further optimization may be necessary to balance processing speed and accuracy, particularly when deploying the system on hardware with limited computational resources.

5.4 Recommendations for Future Research

Future research could explore the following avenues:

- Further refinement of the super-resolution models to ensure consistent performance improvements across various object types.
- Integration of depth information to enhance the accuracy of 6D pose estimation.
- Development of optimized models for deployment on edge devices, focusing on improving computational efficiency.

5.5 Concluding Remarks

This research marks a significant advancement in robotic vision systems by applying AI-driven techniques, setting the stage for future breakthroughs. The integration of super-resolution with pose estimation improves accuracy and expands the scope of potential robotic applications in complex and dynamic environments.

Bibliography

[1] Chao Dong et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[2] James T Kajiya. “The rendering equation”. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, pp. 143–150.

[3] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Accurate image super-resolution using very deep convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1646–1654.

[4] Bee Lim et al. “Enhanced deep residual networks for single image super-resolution”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[5] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37.

[6] J Redmon. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[7] *Robot Types*. URL: `https://toolbox.igus.com/7141/what-are-pick-and-place-robots`.

[8] Stefan Thalhammer et al. *Challenges for Monocular 6D Object Pose Estimation in Robotics*. 2023. arXiv: `2307.12172 [cs.RO]`.

[9] Gu Wang et al. “Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16611–16621.

[10] *YOLOv8 Loss*. URL: `https://github.com/ultralytics/ultralytics/issues/10465`.

[11] Yulun Zhang et al. “Image super-resolution using very deep residual channel attention networks”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 286–301.

Acknowledgements

I would like to thank my advisor, Professor Samuele Salti, for their guidance and support throughout this process. My gratitude also goes to the Hipert Lab team for their collaboration and inspiration. Lastly, to my family and friends, thank you for your constant love and encouragement.