ElectricEye: Metallic Object Pose Estimation

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)

unter der Leitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Markus Vincze
Tessa Pulli Dott.mag.
Peter Hönig MSc.

eingereicht an der
Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungs- und Regelungstechnik

von
Lukas Leimeister B.Eng.
[redacted]

Wien, im Januar 2025

Technische Universität Wien
Karlsplatz 13, 1040 Wien, Österreich

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, _____ _____

Datum Unterschrift

Danksagung

An dieser Stelle möchte ich allen Menschen danken, die mich auf dem Weg zu dieser Masterarbeit unterstützt haben. Mein Dank gilt all den Wegbegleitern, die mich in den vergangenen Jahren sowohl akademisch als auch persönlich unterstützt haben. Ohne ihre Hilfe, ihr Verständnis und ihren Zuspruch wäre dieses Studium in dieser Form nicht möglich gewesen.

Mein besonderer Dank gilt meinem Professor Dipl.-Ing. Dr.techn. Markus Vincze sowie dem Institut für Automatisierungs- und Regelungstechnik der TU Wien, die mir diese Arbeit ermöglicht haben. Ein großes Dankeschön möchte ich auch an meine beiden Betreuenden, Dott.mag. Tessa Pulli und MSc. Peter Hönig, richten. Ihre fachliche Expertise, ihre Geduld und ihre konstruktiven Ratschläge waren für den Erfolg dieser Arbeit von unschätzbarem Wert. Sie haben mir stets mit Rat und Tat zur Seite gestanden und mir geholfen, die richtigen wissenschaftlichen Wege einzuschlagen.

Ein herzliches Dankeschön gilt meiner Familie, die mir stets den Rücken gestärkt hat. Ihre Unterstützung, Geduld und das Vertrauen haben mich in jeder Phase meines Studiums motiviert.

Mein tiefster Dank gilt meiner Partnerin Lisa. Sie war während der gesamten Zeit eine unermüdliche Stütze, hat mich in schwierigen Momenten ermutigt und mir stets die nötige Kraft gegeben, weiterzumachen. Ohne ihre Liebe, Geduld und ihr Verständnis wäre dieser Weg um vieles schwieriger gewesen. Ebenso danke ich Louie und Remy, die mich mit Rat und Tat begleitet haben und stets ein offenes Ohr für mich hatten.

Lukas Leimeister, B.Eng.

Wien, Januar 2025

Abstract

Precisely estimating an object's pose represents a fundamental component in many applications utilizing computer vision, including those within industrial robotics: the textureless surface and high reflectivity of metallic objects present pose estimation challenges. The objective of this master's thesis is to develop a method that enables the robust and accurate estimation of 6DoF poses for metallic and reflective objects in an industrial context.

This thesis builds on the most recent findings in this domain and employs a methodology incorporating contour-based object representation. The method comprises three principal components: a network for object detection and segmentation, a diffusion model for edge detection, and a newly developed network for estimating object poses from edge images. Furthermore, this research entails the creation of datasets that facilitate the training of the networks mentioned above. In this context, a novel rendering pipeline will be developed within the framework of this study, aimed at generating photorealistic training images alongside corresponding ground-truth edge images. The functionality of this pipeline is based on the rendering of realistic textures and illumination conditions, which allows the training data to be adapted to reflect the actual challenges.

The proposed method, called Edge2Pose, involves the detection of the target object by utilizing a YOLOv8 segmentation model. Subsequently, the DiffusionEdge network is employed to detect edges extracted from the scene by the specified region of interest. The edge images are transmitted to the network for pose estimation, which predicts the 3D coordinates based on the edges depicted in the images. This process is analogous to CDPN, Pix2Pose, and DPOD methods. Initially, the 3D coordinates of the model are transformed into RGB values and subsequently predicted by the network. The ultimate pose estimation is achieved by establishing 2D-3D correspondences, which are then processed using the PnP/RANSAC algorithm.

The results of the experiments conducted with diverse data sets (RT-Less, T-Less, and MP-6D) illustrate that the employed methodology is a practical approach for estimating the poses of metallic and reflective objects. Furthermore, this methodology provides considerable advantages in scenarios where the camera consistently focuses on the scene, such as pick-and-place operations.

Kurzfassung

Die präzise Schätzung der Pose von Objekten stellt einen grundlegenden Bestandteil in einer Vielzahl von Anwendungen der Computer Vision dar, wobei auch die Industrierobotik zu nennen ist. Aufgrund ihrer texturlosen Oberfläche sowie ihres hohen Reflexionsvermögens stellen diese Objekte eine besondere Herausforderung im Bereich der Posenschätzung dar. Das Ziel dieser Masterarbeit besteht in der Entwicklung einer Methode, welche eine robuste und präzise Schätzung von 6DoF-Posen für metallische, reflektierende Objekte in einem industriellen Kontext ermöglicht.

Die vorliegende Arbeit basiert auf den jüngsten Forschungsergebnissen in diesem Bereich und verwendet einen Ansatz, der die Objektrepräsentation durch Konturen umfasst. Die Methode besteht dabei aus drei Hauptkomponenten: ein Netzwerk zur Objekterkennung und -segmentierung, ein Diffusionsmodell zur Erkennung von Objektkanten sowie ein neu entwickeltes Netzwerk zur Schätzung der Objektposen aus Kantenbildern. Ein weiterer Bestandteil dieser Forschung ist die Generierung von Datensätzen, welche das Training der verwendeten Netzwerke ermöglicht. Um dieses Ziel zu erreichen, wird eine neue Rendering-Pipeline implementiert, die fotorealistische Trainingsbilder in Kombination mit Ground-Truth-Kantenbildern erzeugt. Die Funktionsweise dieser Pipeline basiert auf der Simulation realistischer Texturen und Beleuchtungsbedingungen, wodurch eine Anpassung der Trainingsdaten an die tatsächlichen Herausforderungen möglich ist.

Die implementierte Methode der Posenschätzung, genannt Edge2Pose, umfasst die Erkennung des Zielobjekts durch die Verwendung eines YOLOv8-Segmentierungsmodells. Die Erkennung von Kanten erfolgt durch das DiffusionEdge-Netzwerk, welche aus der Szene entsprechend der Region of Interest extrahiert werden. In der Folge werden die Kantenbilder an das Netzwerk zur Posenschätzung übermittelt, welches die 3D-Koordinaten auf Basis der in den Bildern dargestellten Kanten prognostiziert. Die Vorgehensweise ist vergleichbar mit der von CDPN, Pix2Pose und DPOD, wobei die 3D-Koordinaten des Modells zunächst in RGB-Farbwerte umgewandelt und anschließend durch das Netzwerk vorausgesagt werden. Die finale Pose-Schätzung erfolgt durch die Generierung von 2D-3D-Korrespondenzen und deren nachfolgender Berechnung mittels PnP/RANSAC-Algorithmus.

Die Resultate der Experimente, welche mit unterschiedlichen Datensätzen (RT-Less, T-Less und MP-6D) durchgeführt wurden, demonstrieren, dass die implementierte Methodik eine valide Vorgehensweise zur Posenschätzung von metallischen, reflektierenden Objekten darstellt. Des Weiteren demonstrieren die Resultate, dass besagte Methodik insbesondere für Szenarien vorteilhaft ist, in denen die Kamera durchgängig auf die Szene fokussiert ist, wie es beispielsweise bei Pick-and-Place-Operationen der Fall ist.

Glossary

ADD Average Distance of Model Points.

ADD-(S) Symmetrized Average Distance of Model Points.

AR Augmented Reality.

AR Average Recall.

BOLD Binary Object Line Descriptor.

BOP Benchmark for 6D Object Pose Estimation.

CAD Computer Aided Design.

DoF Degrees of Freedom.

EMA Exponential Moving Average.

IoU Intersection over Union.

MAE Mean Average Error.

mAP Mean Average Precision.

MSPD Maximum Symmetry-Aware Projection Distance.

MSSD Maximum Symmetry-Aware Surface Distance.

NOCS Normalized Object Coordinate Space.

PnP Perspective-n-Point.

RANSAC Random Sample Consensus.

RoI Region of Interest.

VSD Visible Surface Discrepancy.

Contents

CHAPTER 1

Introduction

Precisely estimating object poses is a key component in various computer vision applications, such as industrial robotics [1–4]. Despite significant advancements and the development of highly effective methods for textured objects, a pressing need remains for solutions that are capable of dealing with textureless objects and, in particular, with the challenges faced by metallic objects [1, 5–9].

1.1 Motivation

The rapid advancement of intelligent manufacturing has made pose estimation of industrial objects a pivotal technology for robotic grasping, unit assembly, and human-machine collaboration [1, 4–9]. Despite notable advancements in pose estimation through numerous methodologies, these approaches remain partially applicable to metallic objects due to their reliance on surface-based features [2–4]. The surface properties of metallic objects often prevent extracting distinctive features, resulting in significant deviations in position estimation. Due to their high reflectivity, metallic objects are sensitive to environmental influences and varying lighting conditions, further complicating the process of matching features [1, 10–12]. To successfully address these challenges, it is necessary to establish a reliable and robust technique that can accurately and precisely estimate the pose of metallic objects. A promising approach is utilizing the object's geometric properties, such as contour and edges. [11–19]. The object's contour is not affected by surface texture and is less sensitive to lighting effects. Contour representation offers a promising method for accurately determining the orientation of metallic objects.
The objective of this master's thesis is to develop a reliable methodology for extracting the edges of objects. This approach aims to utilize these edges to estimate poses, thereby enhancing the accuracy of pose estimation specifically for metallic objects and capitalizing on their properties.

1.2 Problem Statement

Most current research focuses on the object's surface. The handling of metallic objects is of particular importance in industrial contexts. While several solutions are available for object pose estimation of textured objects, research addressing reflective objects is limited. Depth information is effective for estimating object poses, but specialized hardware is needed to capture this information. Cameras have already achieved widespread integration within industrial applications and offer a more efficient data processing advantage over depth images [1, 8]. As a consequence, RGB has emerged as a dominant technology for object pose estimation over the past few years [1, 5–8].

The application of metallic objects presents a considerable challenge in the 6D pose estimation. The optical and physical properties of metal parts require different approaches than algorithms for textured objects. One of the main difficulties with metallic objects is their high reflectivity. In contrast to textured surfaces, which show uniform light scattering and thus provide reliable texture information, metallic surfaces reflect light sources and environments. The specular reflectivity of metallic surfaces presents a significant challenge, as the surface's appearance highly depends on light incidence and camera position. Specular highlights and specular reflections cause artifacts in RGB images that can introduce errors in both classical feature-based algorithms and modern neural network-based methods. These effects lead to visual distortions that are often not sufficiently represented in training data [11, 14, 16, 20, 21].

Another aspect is the lack of texturing in many metallic objects, which complicates the application of methods that rely on detecting texture-related features. The metallic properties lead to strong specular effects that complicate the recognition of relevant object features and restrict the generation of robust image features for pose estimation. Low-texture surfaces provide a few visually consistent points that algorithms can use for reliable detection and estimation. This lack of features hinders pose estimation algorithms and methods like correspondence matching [22–31].

In addition, the spectral reflectance of metallic objects often leads to varying appearances depending on the illumination and viewing angle. Especially in industrial applications, where lighting conditions usually cannot be kept constant, pose estimation can be severely affected as the visible object features change dynamically with each change in ambient lighting. Even with steady lighting, small changes in the angle of the camera or object can cause substantial variations in image captures, resulting in reduced reliability of pose estimations [10]. The instances in Figure 1.1 illustrate the complexities of imaging metallic objects. These challenges include textureless surfaces that lack distinctive features, viewpoint-dependent reflections, surface mirroring effects, and susceptibility to overexposure and underexposure in varying lighting conditions.

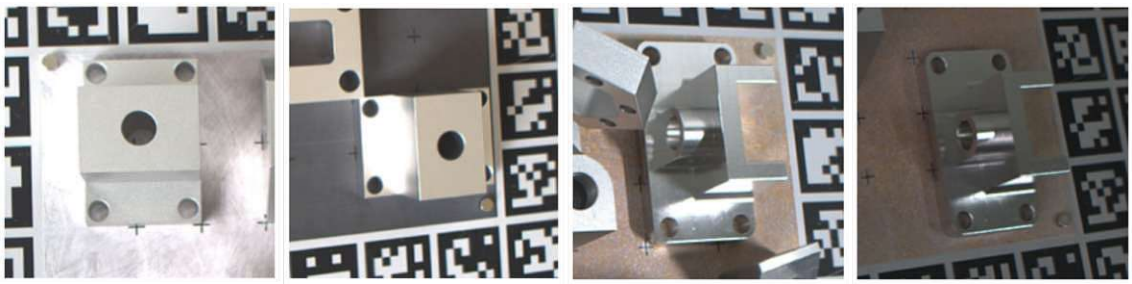

Figure 1.1: Example of metallic objects an their challenges [10]

Based on the preceding discussion of challenges and approaches, this work will address the following research questions:

- **RQ 1:** How can a synthetic training dataset be designed to accurately replicate the visual properties of metallic objects, such as reflections and gloss?
  - The generation of synthetic datasets for metallic objects represents a significant challenge, given the intricate light reflections and reflective surfaces that are inherently difficult to replicate. This leads to the question of how synthetic training data can be created to meet the actual requirements of pose estimation.
- **RQ 2:** How to extract edges and contours of metallic objects accurately and robustly?
  - The objective is to develop a robust edge extraction technique that is not dependent on illumination conditions or surface properties.
- **RQ 3:** To what extent can existing pose estimation methods be leveraged by introducing contour-based features?
  - A key question is whether existing pose estimation methods can be enhanced by incorporating contours and to what extent this allows for more precise estimation of metallic objects.

1.3 Approach and Contribution

This thesis aims to develop a methodology to estimate the 6 Degrees of Freedom (DoF) poses of reflective metallic objects for industrial applications. This method does not rely on the characteristics of the object surfaces but rather utilizes the contour of the objects as a source of information for position determination. Edges are detected using a diffusion model, and the resulting images are subsequently leveraged to establish 2D-3D correspondences. This process results in estimating the final 6DoF object pose. This approach is tested on an industry-related metallic object dataset, providing various scenarios and challenges. The scope of this work is divided into three main components: data preparation, edge detection, and pose estimation 1.2.

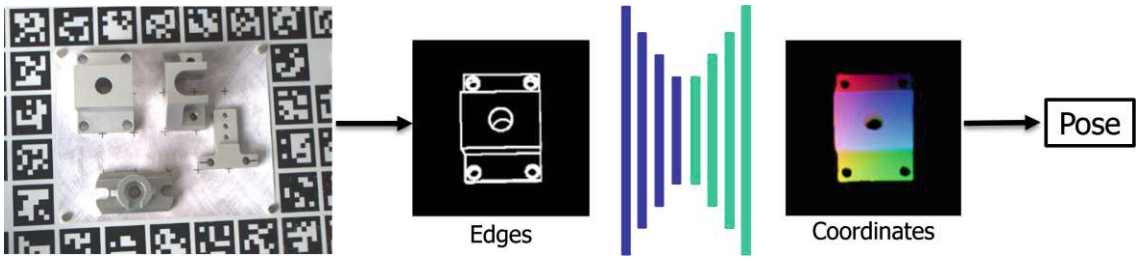

Figure 1.2: Overview of the proposed approach: The scene is converted into an edge-detected image, from which the object is extracted. The resulting contour establishes 2D-3D correspondences, which are essential for estimating the object's pose.

A dataset is required to train a pose estimation algorithm. Since the benchmark datasets in this area (Linemod, YCB, etc.) [2, 3] do not fully meet the specific requirements, the RT-Less [10] dataset is utilized as well as T-Less [32] dataset to further extend the proposed method onto another state of the art dataset. The RT-Less dataset comprises Computer Aided Design (CAD) models and real test scenes with respective ground-truth pose annotations. The data required for training and evaluating the method of this work is created with the introduced rendering pipeline. This pipeline enables photorealistic scene image creation and provides ground truth edge images and segmentation masks. This novel rendering pipeline leverages the Blender [33] software and is inspired by the functionalities of BlenderProc [34] and the RT-Less toolkit [35].
The method presented in this thesis is split into two main sections: Edge Detection and Pose Estimation. The edge detection stage comprises the task of object detection and edge detection. YOLO [36] algorithm is used to perform the first task. In particular, the latest YOLOv8 [37] model is applied. This procedure accomplishes the dual purpose of object detection and segmentation. Based on the bounding boxes and segmentation masks derived in this step, the object in the scene is cropped the Region of Interest (RoI), similar to the techniques in [25, 28, 31]. A diffusion model is used to perform the secondary task in this main section, edge detection. The so-called DiffusionEdge [38] has been specially designed to generate precise and accurate edge images. The training of this network and YOLOv8 [37] for object detection is conducted with the previously rendered scene images and edge images.
The second main stage of the method contains a novel network, which receives the extracted edge image of the object as input to predict the demanded coordinate maps. This network is based on the approaches of GDR-Net [28], CDPN [31], Pix2Pose [25] and DPOD [29], whereas a new decoder for edge images is introduced into a U-Net structure. This network

is trained on data generated with the new rendering pipeline. For this purpose, coordinate maps of the individual objects are created together with corresponding edge images. The normalized coordinates of each vertex of the model are transferred directly to the red, green, and blue values of the color space. The model predicts the color-coded coordinate maps based on the given edge images, thus enabling the creation of 2D-3D correspondences. The final pose is estimated by applying the Perspective-n-Point (PnP) [39] algorithm with further use of Random Sample Consensus (RANSAC) [40] to improve the estimation results. The contributions of this thesis can be summarized as follows:

- Introduction of a rendering pipeline for creating photorealistic scenes with ground truth edge images and generating color-coded coordinate maps enabling the transfer of 2D-3D correspondences from color values.
- Implementation of object detection and edge detection of industrial, reflective, and metallic objects utilizing a state-of-the-art object detection network and a state-of-the-art diffusion model.
- Application of a 6DoF pose estimation pipeline for industrial, reflective, and metallic objects based on their edge representation before coordinate prediction.

1.4 Organization

The organization of this thesis can be described as follows: First, chapter 2 provides a detailed explanation of the technical background of this work. Additionally, this chapter presents a fundamental analysis of the relevant work and carefully examines methods that specialize in applying metallic objects. In the following chapter 3, the methodology implemented in this thesis is presented. Subsequently, chapter 4 describes the concept's practical implementation and the resulting findings. The final discussion of the results and possible subsequent applications can be found in chapter 5.

CHAPTER 2

Background and Related Work

This chapter provides essential background information and concepts relevant to the thesis. It outlines the fundamental methodologies and approaches while exploring the specific topic of metallic object pose estimation to enhance understanding of the core subject. Initially, the chapter discusses the background of pose estimation and the metrics used for evaluation. Following this, it evaluates related and relevant works that have previously tackled this issue and assesses their significance concerning the research question of the thesis.

2.1 Background

"Object pose estimation" refers to precisely determining an object's position within its three-dimensional space. This process involves finding the relative pose between the camera's coordinate system and that of the object. The "6 Degrees of Freedom" pose captures the spatial orientation of an object in a reference coordinate system. The pose is composed of both a three-dimensional translation and a three-dimensional rotation. "Translation" refers to moving an object's coordinates along a coordinate system's x, y, and z axes. At the same time, "rotation" refers to the circular motion of the object around each of these three axes. Understanding these concepts is crucial for accurately assessing an object's position and orientation in any given environment. As illustrated in Figure 2.1, in an industrial application scenario, the camera is mounted on the robot's front end while the object is positioned on a reference table. The camera's coordinate system is fixed, with the z-axis aligned with the view axis. The coordinate system of the external environment is fixed to the center of the observed scene; thus, the z-axis is oriented vertically upward [41–43].
Numerous visual computing tasks necessitate a comprehensive understanding of scenes and the manipulation of objects. 6DoF pose estimation is essential for providing detailed information on both positional and orientational parameters, enabling robotic systems to perform tasks such as object recognition, localization, and grasping with enhanced precision and accuracy. Furthermore, 6DoF pose estimation is critical in various aspects of autonomous driving, including environmental perception, obstacle detection, traffic condition prediction, and decision-making. An additional significant application of 6DoF pose estimation is in the realms of augmented reality (AR) and virtual reality (VR), where it supports the development of spatial mappings of environments and provides essential data for the effective integration of AR/VR content [1, 5, 6, 8].

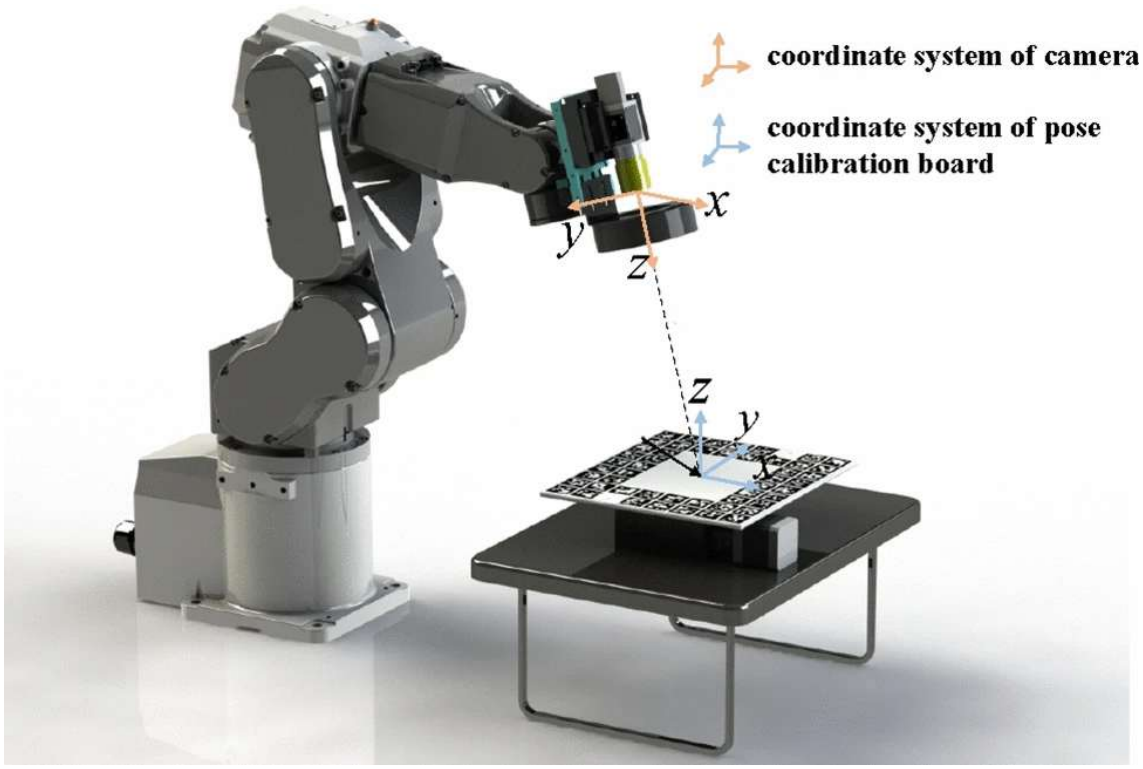

Figure 2.1: An industrial robot with the camera attached to the front end of the arm [10] ©2023 Springer Nature.

2.1.1 Object Representation

Following the preceding discussion, the location of an object within three-dimensional space is defined in terms of its six degrees of freedom (DOF) pose. Each object possesses its local coordinate system, anchored at a specific point on the object. The position and orientation of the aforementioned coordinate system describe the object's pose relative to the said coordinate system. The camera's coordinate system is commonly employed as the reference system in pose estimation. Pose estimation aims to determine the necessary transformation to align one coordinate system with another [6]. The translation vector $t = \left[t_x, t_y, t_z\right]^T$ indicates the position of the origin of the object's coordinate system in the global coordinate system. It describes how far the object is displaced along the x, y, and z axes. Various methods exist for representing rotation in 3D space, such as Euler angles, rotation matrices, and quaternions. However, the most common method uses a 3x3 matrix R, which describes the rotation of the object with respect to the global coordinate system. The orientation of the object can be expressed by combining the translation vector t and the rotation matrix R, using the homogeneous transformation matrix T (2.1). This matrix describes the complete rigid transformation in 3D space [41–43].

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \tag{2.1}$$

In most applications, visual information is used to estimate the pose. The most prevalent approach is to utilize images captured by the camera [1, 5, 6]. The two-dimensional image of a three-dimensional object is generated by the projection of three-dimensional points from the real world onto the two-dimensional image plane of a camera. This projection is based on the principles of perspective projection, whereby each three-dimensional point in space is mapped to a two-dimensional coordinate in the image through the lens of a camera [10]. A camera matrix is used to model this projection, comprising the intrinsic parameters of the camera, including the focal length and the optical center 2.2. In the PnP method, these projected 2D points are used together with the known 3D coordinates of the object to calculate the 6DoF pose relative to the camera. The PnP method [39] solves an optimization problem in which the task is to find the rotation and translation (i.e., the pose) that positions the object in 3D space so that the projected 2D points match the observed 2D coordinates as closely as possible [41–43].

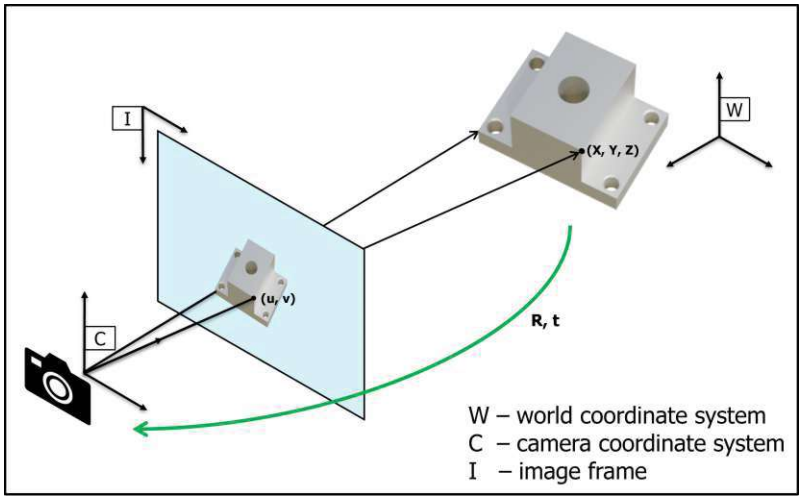

Figure 2.2: Visualization of PnP [39] and object imaging

2.1.2 Evaluation Metrics

The accuracy and precision of six degrees of freedom (6 DoF) object pose estimation are evaluated by comparing the predicted poses with the known ground truth poses of the depicted object. The error metrics Average Distance of Model Points (ADD) and Symmetrized Average Distance of Model Points (ADD-(S)) are utilized to assess the accuracy of the pose estimate. These metrics differ in their applications and calculation procedures, especially when dealing with symmetric objects [2, 3]. The ADD metric calculates the average distance of model points between an object's estimated and ground-truth pose. For each pair of points (a point in the estimated model and the same point in the ground truth model), the Euclidean distance is calculated, and the average of these distances equals the ADD value. The equation 2.2, which is defined as the average distance of the pairwise distances between the 3D model points transformed with the ground truth and estimated poses, where M is the set of 3D model points, m is the number of points, and (R_{GT}, T_{GT}) (R_{EST}, T_{EST}) are the rotation and translation of the ground truth pose and predicted pose.

$$\text{ADD} = \frac{1}{|\mathbf{P}|} \sum_{p \in \mathbf{P}} \|p(R_{\text{GT}} + t_{\text{GT}}) - p(R_{\text{EST}} + t_{\text{EST}})\| \tag{2.2}$$

ADD-(S) enhances ADD to effectively handle symmetrical objects as well. Since symmetrical objects can appear identical from certain perspectives, directly pairing points may result in errors. Therefore, ADD-(S) identifies the nearest point in the ground truth model for each point in the estimated model. The distances to these nearest points are then averaged, accounting for the object's symmetries 2.3.

$$\text{ADD-S} = \frac{1}{|\mathbf{P}|} \sum_{p_1 \in \mathbf{P}} \min_{p_2 \in \mathbf{P}} \|p_1(R_{\text{GT}} + t_{\text{GT}}) - p_2(R_{\text{EST}} + t_{\text{EST}})\| \tag{2.3}$$

A pose is deemed correct if the ADD-(S) error is less than 10 % of the object's diameter. The proportion of all accurately predicted poses is referred to as the ADD-(S) recall [44]. Furthermore, to address the exact precision of the estimated poses, we use the mean error of rotation R_{mean} and translation T_{mean} between the estimated pose and the ground truth.

$$R_{\text{mean}} = \frac{1}{n} \sum_{i \in S} \text{avg}\left(|\alpha_i - \alpha_i'| + |\beta_i - \beta_i'| + |\gamma_i - \gamma_i'|\right) \tag{2.4}$$

$$T_{\text{mean}} = \frac{1}{n} \sum_{i \in S} \text{Euclidean}\left((x_i, y_i, z_i), (x_i', y_i', z_i')\right) \tag{2.5}$$

Both mathematical expressions can be seen in equations 2.4 and 2.5, where S is the set of correct estimated poses according to ADD-(S). Further, n is the number of images in S. The values of the rotation angles in the respective axes are represented by $(\alpha_i, \beta_i, \gamma_i)$ for the estimated pose and by $(\alpha_i', \beta_i', \gamma_i')$ for the ground truth pose. Similarly, the translations for the predicted pose (x_i, y_i, z_i) and the ground truth pose (x_i', y_i', z_i') are represented [10]. Another set of metrics is relevant when comparing against the results of the Benchmark for 6D Object Pose Estimation (BOP) Challenge: Visible Surface Discrepancy (VSD), Maximum Symmetry-Aware Surface Distance (MSSD) and Maximum Symmetry-Aware Projection Distance (MSPD) [2, 3, 44]. These metrics measure the distance or deviation between the estimated pose of an object and the ground truth pose, taking into account different aspects such as visibility, symmetry, and projection. Based on depth images, the VSD metric evaluates an object's visible surface correspondence between the estimated and

ground-truth pose. This metric is beneficial when the object is partially occluded, as only the visible points are considered. The VSD metric calculates the error e_{VSD} as the average depth difference between the pixels visible in both poses.

$$E_{\text{VSD}} = \frac{1}{|P|} \sum_{p \in P} \delta\left(|d_{\text{est}}(p) - d_{\text{gt}}(p)|, \tau\right) \tag{2.6}$$

$$\delta(x, \tau) = \begin{cases} 1, & \text{if } x > \tau \\ 0, & \text{else} \end{cases} \tag{2.7}$$

Where P is the number of visible pixels in both poses, $d_{est}p$ and $d_{gt}p$ are the depth values of the pixels p in the estimated and ground truth poses and τ the misalignment threshold, which specifies the maximum tolerance for the depth difference.
The MSSD metric measures the maximum deviation of the model surface points between the estimated pose and a symmetrically adjusted ground truth pose to account for possible symmetries of the object. The error e_{MSSD} is calculated by taking the maximum distance between the point clouds of the estimated and the symmetrically adjusted ground truth pose.

$$e_{\text{MSSD}} = \min_{\text{sym} \in \text{S}} \max_{p \in \text{P}} \|p_{\text{EST}} - p_{\text{GT, sym}}\| \tag{2.8}$$

Where p_{est} is a point on the surface of the model in the estimated pose, $p_{gt,sym}$ the corresponding point in the symmetrically transformed ground truth pose, and S the set of all possible symmetries of the object.
The MSPD metric evaluates the accuracy of the pose based on the projection of the model points into the image plane. It calculates the maximum 2D distance between the projection points of the estimated and the symmetry-aware ground-truth pose. The MSPD metric is defined by the following formula:

$$e_{\text{MSPD}} = \min_{\text{sym} \in \text{S}} \max_{p \in \text{P}} \|\pi(p_{\text{EST}}) - \pi(p_{\text{GT, sym}})\| \tag{2.9}$$

Where π is the projection function that projects 3D points into the 2D image plane.
The Average Recall (AR) is calculated by measuring the recall for each of these metrics (VSD, MSSD, MSPD) over a range of thresholds and then averaging them. AR indicates how well a model predicts poses at different levels of accuracy and provides a summary assessment of performance across different error tolerances.

$$\text{AR} = \frac{e_{VSD} + e_{MSSD} + e_{MSPD}}{3} \tag{2.10}$$

2.2 Related Work

This section provides an overview of the most widely utilized methods and significant works in pose estimation. It starts by discussing general methods and their foundational concepts, which have yielded impressive results in recent years. Following that, an in-depth exploration of techniques specifically applicable to textureless or metallic objects is presented.

2.2.1 General Pose Estimation Approaches

In computer vision, two main approaches to pose estimation are instance-level and category-level. Instance-level pose estimation detects and estimates a known object's pose. This method estimates the pose of known three-dimensional objects. Category-level pose estimation predicts an object's pose without precise information about the object. This approach addresses the general pose of an object within a category. This thesis examines instance-level pose estimation [1].
The initial step in categorizing these methods is to classify them according to the underlying data type. They can be divided into RGB-based methods, point cloud or depth-based methods, and RGB-D-based methods [45]. While depth measurements have proven to be reliable for estimating object poses, they often require specialized hardware. In contrast, RGB sensors have seen widespread integration in industrial applications and provide more efficient data processing than depth images. As a result, RGB technology has become the preferred choice for object pose estimation in recent years [1]. This thesis delves deeper into pose estimation using RGB images and focuses on relevant studies. These RGB-based methods can be further classified based on functionality. In this context, Guan et al. [5], and Marullo et al. [7] categorize the methods into three distinct groups: regression-based methods, template-based methods, and feature-based methods.
Across the different approaches to pose estimation, several methods use RoI or crop the object out of the image to reduce the computational load and improve accuracy. Such methods [22, 24, 25, 28, 31] usually employ segmentation or detection networks [36, 46] to localize the object and thus focus the image section that is then used for pose estimation.

Regression-Based Methods

In recent years, deep learning-based methods have shown their ability to handle object pose estimation. One of the most straightforward approaches is the direct regression of the 6DoF pose, where the estimation is performed directly on the RGB image without intermediate steps such as segmentation or keypoint extraction. These methodologies are typically implemented as end-to-end applications, which feature a neural network trained and used to regress the 6DoF poses directly from the input image. To simplify pose determination, the primary phase of these methods usually involves an object detection process that locates the object within the scene image in advance.

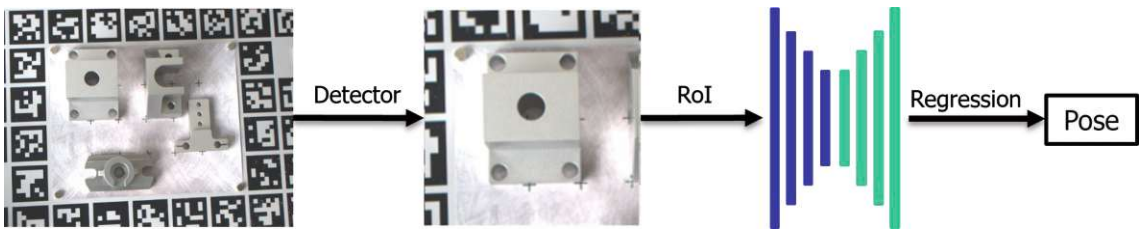

Figure 2.3: Typical workflow of direct-regression-based methods

Early methods such as PoseNet [47] and PoseCNN [24] use Convolutional Neural Networks (CNNs) to regress the pose parameters directly. In contrast, PoseCNN introduces multi-task learning and splits the pose regression into translation and rotation. To further improve accuracy, Deep-6DPose [48] and 6D-VNet [49] combine CNNs with Mask R-CNN-like [46] structures and extend them with additional branches specifically designed for pose estimation. Hu et al. [50] estimating pose via direct regression of 3D correspondences. This method combines the PnP algorithm with neural networks to efficiently generate correspondences from key points obtained by PVNet [22], while the iterative RANSAC step is embedded in the network. DeepIM [51] and CosyPose [26] rely on iterative improvements in pose estimation by successively reducing the difference between the rendered model and the input image. CosyPose also integrates symmetry detection and multi-view information to refine the estimates. GDR-Net [28] proposes employing geometrically guided regression methods and using dense correspondences to make pose estimation particularly accurate and stable.
Although the current method of predicting dense key points achieves superior performance for the pose estimation of ordinary objects, the surface of reflective and textureless metal parts can provide little semantic information.

Feature-Based Methods

Feature-based methods are a commonly used approach in 6DoF pose estimation that extracts distinctive image features and matches them with a 3D object model to establish a 2D-3D correspondence. The general process involves detecting characteristic features in the image that can be matched with corresponding features on the 3D model. The object's pose is then determined using the PnP [39] algorithm in conjunction with RANSAC [40] to increase accuracy by iteratively filtering out mismatches.

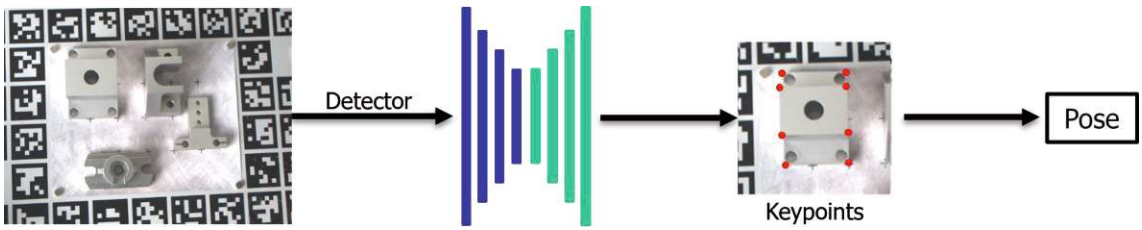

Figure 2.4: Typical workflow of feature-regression-based methods

Methods such as Pix2Pose [25], DPOD [29], and PVNet [22] rely on pixel-wise predictions to compute 2D-3D correspondences. Pix2Pose uses an autoencoder architecture to predict the 3D coordinates of individual object pixels without needing a textured model. These methods are based on the approach that the 3D coordinate of each model vertex can be transferred to the 2D RGB color space by an appropriate color coding and then predicted. These then enable the establishment of 2D-3D correspondences for pose estimation. Additional methods extend the classical feature-based approaches with additional representations to extract geometric information in a more targeted way. HybridPose [15] builds on PVNet and integrates edge vectors and symmetry correspondences as additional intermediate representations, enabling detailed geometry analysis in the image. These hybrid representations improve accuracy but require a higher computational effort. Liu et al. [52] combine RGB image information with a depth map created by a U-Net architecture and apply the DenseFusion [53] network to generate dense correspondences for each pixel. This enables detailed pose estimation and improves accuracy by combining global image features with

pixel-wise poses. To estimate the pose of symmetrical objects, EPOS [54], Pix2Pose [25], and Mei et al. [55] rely on unique symmetry treatments. EPOS segments the object into symmetry-invariant fragments and calculates a probability distribution for each fragment to determine the pose. Mei et al. use the spherical correlation method to learn a latent spherical feature representation that is rotation invariant and robustly estimates the pose of symmetric objects. Pix2Pose introduces a special loss function, the "Transformer-Loss," to consider all possible symmetries while training.

Template-Based Methods

Template-based pose estimation methods utilize templates created from various viewpoints of an object to determine the object's pose by finding the template that best matches the input image. This process consists of two main phases. In the offline phase, a database of templates is built by synthesizing a three-dimensional model of the object from multiple positions and orientations. During the online phase, the input image is compared with the templates to identify the best match, allowing for the estimation of the object's 6D position.

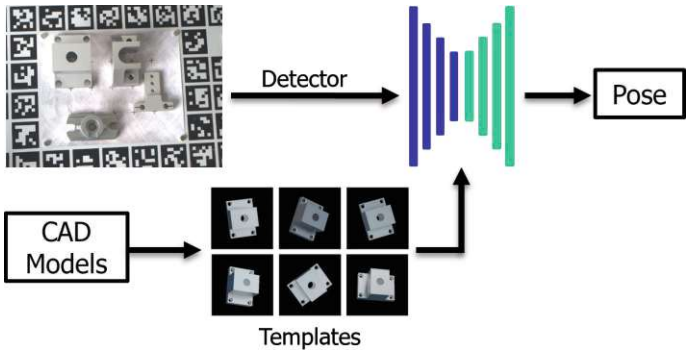

Figure 2.5: Typical workflow of template-based methods

SSD-6D [30] extends 2D object detection (SSD) to a 6D pose estimation system and uses an end-to-end architecture for fast and robust object pose determination. DPOD [29] also follows a template approach but combines template detection with a dense matching approach that establishes 2D-3D correspondences between image pixels and the 3D model of the object. DPOD does not require perfect object segmentation and demonstrates robustness to occlusions and light changes. Nguyen et al. [56] exceed this by generating a large collection of templates of new objects and retrieving the pose via color template detection. For each new object, numerous views are rendered around the 3D model, later compared in real-time with the input image to recognize the object and determine its pose. Template-based methodologies offer numerous advantages, such as their inherent simplicity, rapid processing capabilities, and adaptability to variations in appearance. Furthermore, these methods demonstrate enhanced effectiveness in handling weakly textured objects, making them valuable in various application domains [5, 6, 9]. Template matching is a straightforward and intuitive method that enables swift detection and localization of objects. However, it can encounter difficulties in complex situations involving occlusions, lighting variations, or objects lacking distinctive features [5, 6, 9].

2.2.2 Metallic Object Approaches

The reflective nature of metal parts results in significant variation of the object's pixel colors depending on the lighting conditions and shooting angle. In addition, the surfaces of metal parts are less distinguishable than those of textured objects. Therefore, traditional methods based on texture features, color gradients, or clear geometric features have little guidance for reliable detection and pose estimation, often resulting in inaccurate or erroneous results. Due to this limitation, several methods have been developed specifically for applying metallic, reflective objects. A fundamental understanding that various studies [11–19] have concluded is that the application of edge information provided by objects is a promising technique to improve the pose estimation of textureless objects. The extent to which edge information is used varies across these methods. Some leverage edges directly for iterative optimization, while others extract semantic features from the edges based on deeper geometric relationships.

Methods such as ContourPose [11] or ER-Pose [16] use edge information directly to determine the pose of objects and then adjust it further in an optimization step. ContourPose by He et al. [11] combines a two-stage pipeline consisting of a neural network (ContourNet) and an iterative pose optimization algorithm, as can be seen in Figure 2.6. In the first stage, key points are predicted with implicit constraints on contour. In this stage, the key points and contours of the target object are predicted. The second stage consists of pose estimation using contour as a prior. In this stage, the contour predicted in the previous stage is used as geometric priors to eliminate outlier poses and output the optimal pose in the result set. ContourNet generates a heatmap with 2D nodes and contours of the object, which implicitly constrain the nodes. The optimization algorithm then leverages the contour information to calculate an optimized 6D pose. ER-Pose [16] is a two-stage framework for pose estimation of reflective, textureless objects. In the first stage, the edge representation of the object is extracted from an RGB image, where the direction and distance to specific key points within the object's edges are determined. This information is used to generate 2D-3D correspondences. In the second stage, the pose is optimized using a PnP/RANSAC [39, 40] algorithm that calculates the 6D position and orientation of the object. The approach is particularly robust against disturbances and reflections as it is based on stable edge features. Druskinis et al. [57] use a hybrid architecture that combines Mask R-CNN [46] for object detection with edge-based pose estimation, where the pose estimation part is inspired by the work of Choi and Christensen [17]. The edges of the previously segmented object are extracted using the Canny [41] algorithm and matched with an edge database, which enables robust matching for 6D pose estimation.

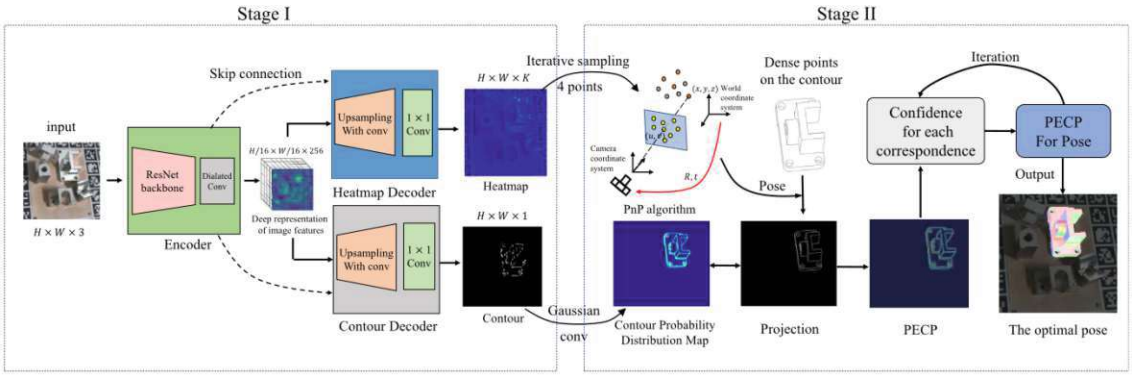

Figure 2.6: Overview of ContourPose [11] ©2023 IEEE

Rather than directly using edges, other methods rely on semantic features, such as pairs of nodes or line segments that appear along the object's edges. The method proposed by Liu et al. [12] is based on semantic-level line matching. Unlike existing low-level feature methods, this approach relies on semantic-level line features. The semantic-level line features are identified by the line detection network L-CNN [18] and incorporated with a segmentation network to extract object-level line descriptors. By matching the descriptors in a sparse template set, the 2D-3D mapping from the actual image to the model of the metal part is realized. Inspired by the BOLD operator [58], they propose an object-level line descriptor to describe each feature of the line related to the object. Binary Object Line Descriptor (BOLD) [58] features are descriptors for textureless objects based on short line segments (instead of points). They were developed to provide stable features for object detection and pose determination without surface structures. BOLD features use line fragments described in a binary format, which makes them robust against light changes and reflections. The work by He et al. [14] also leverages the contours of metal objects and extracts high-level features for matching between real and template images. The LSD [59] method for line detection is further improved to complete and extract straight contours of the objects. Again, the BOLD [58] feature is used for object detection, which describes the correlation of a set of neighboring short-line fragments. After selecting the most matched template from the CAD template database, the absolute pose is calculated using the EPnP [39] algorithm. Alternatively, hybrid approaches combine classic edge detection with the advantages of neural networks. Here, networks can pre-process features that are then further processed by classic algorithms. Such methods offer the robustness of neural networks and the efficiency of classical algorithms. Chen et al. [20] proposed a three-step process: object detection, feature extraction, and pose estimation. They utilize Mask R-CNN [46] to detect objects and HRNet [60] to extract the corresponding features. This method does not depend on continuous contours but examines distinct combinations of dense discrete points along the edges. Hu et al. [61] proposed a cascaded neural network architecture similar to SSD6D [30], which utilizes the size of the predicted bounding box to provide a depth estimation. Subsequently, a two-stage rough-to-fine pose model provides a pose estimation. The approach by Chen et al. [21] employs a three-phase framework of object detection, feature detection, and pose optimization. They leverage the contour information for pose estimation. They use the dense discrete points along the edges of the metal part as semantic key points for contour detection. Afterward, the 6D pose is calculated by exploiting both keypoint information and the CAD model. He et al. [13] propose a generative feature-to-image framework based on generative models, whose pipeline is a reverse mapping from feature to image. In other words, given a feature representing a pose, this method generates an image of the object in the same pose. They also apply an edge-based approach by regressing edge images, which are compared to templates, resulting in the estimated pose. De Roovere et al. [19] proposed CenDerNet, which features a three-stages-framework for 6D pose estimation from multi-view images based on center and curvature representations. A convolutional neural network is trained to predict center and curvature heat maps. This step eliminates task-irrelevant variations by converting images into center and curvature representations. They detect objects and estimate their approximate locations using center heatmaps, representing the likelihood of object center points. In addition, they use curvature heatmaps to emphasize local geometric features, making it easier to compare images with rendered models.

2.2.3 Datasets

Datasets for pose estimation are central to developing and evaluating algorithms for precise object localization. They contain various annotated images or 3D models that are used to prepare neural networks and other methods for realistic scenarios. While general posed datasets mostly contain everyday objects with clear textures and shapes, methods encounter significant challenges when recognizing industrial objects, especially metallic and reflective parts. Industrial pose datasets for such objects are characterized by scenarios with complicated lighting conditions, reflective surfaces, and often textureless, smooth structures that can overwhelm conventional approaches. These specialized datasets, therefore, often include synthetic renderings and multi-modal data such as RGB and depth images to develop and test robust algorithms for accurate pose estimation in industrial environments. With the development of computer vision and deep learning, increasingly diverse pose estimation datasets have been proposed, which can be categorized into containing non-industrial objects and industrial objects [1].

Non-Industrial Datasets

Non-industrial datasets for pose estimation provide an important basis for training and testing algorithms and include various everyday objects in different scenarios. One of the most important resources is the BOP Challenge [2, 3], which combines several widely used datasets and offers standardized comparison options. The BOP datasets include LINEMOD, LINEMOD-Occluded [62] and YCB-V [2].
LINEMOD is one of the most frequently used datasets and contains images of various textured everyday objects with different backgrounds. It offers easily recognizable objects thanks to precise edges and textures and is, therefore, suitable for algorithms that rely on texture-based features. The extension to this dataset, LINEMOD-Occluded, introduces scenes that meet the requirement for different occlusions among objects. YCB-V [2] contains a variety of household objects in multiple views and positions, with complex interactions and partial occlusions. It is particularly valuable for pose estimation tasks that require robust algorithms in highly realistic scenarios.

Figure 2.7: Overview of BOP datasets [2]

Industrial Datasets

Metallic and textureless industrial datasets for pose estimation aim to address the particular challenges of industrial contexts where smooth, reflective, and low-detail objects are often found. Unlike datasets with everyday objects, these industrial data collections place exceptionally high demands on pose determination algorithms, as metallic surfaces usually lead to strong reflections and light distortions, while textureless objects offer little to no visually distinctive features that classical algorithms rely on.
The two industrial datasets T-Less [32] and ITODD [63] are included in the collection

of the BOP Challenge [2, 3]. The T-LESS [32] (Texture-Less Object Dataset) includes objects largely made of plastic and recorded in various scenarios, some containing strong overlaps and occlusions. T-LESS [32] allows algorithms to be trained and evaluated for realistic production environments in which the visual characteristics of the objects are minimal. ITODD [63] (Industrial Textureless Object Dataset and Benchmark) is another specialized dataset focusing on textureless and reflective industrial objects. The use of metallic materials further increases the challenge for the algorithms, as reflections and light interferences make pose determination more difficult.
In addition to the BOP Challenge, other datasets expand the application possibilities for textureless and metallic objects. These are based on the general format of the BOP datasets but are not included in the collection. These include MP6D [64], RT-Less [10] and the proposed work by De Roovere [65]. MP6D (Metallic Parts 6D Pose Dataset) [64] is an industrial dataset that includes detailed scenes with complex shaped metal parts and provides realistic, industrial scenarios for pose determination, taking into account the strong reflections caused by metallic surfaces. MP6D [64] is particularly relevant for applications in the manufacturing and automotive industries where such objects are common. The RT-Less [10] dataset contains a variety of metallic objects captured in different lighting conditions, reflection ratios, and from different angles. These variations simulate realistic industrial scenarios in which strong reflections from the point or directional light sources occur, and the object's visual appearance changes significantly depending on the camera's perspective. De Roovere [65] proposed another dataset focusing on industrial objects. The authors present a diverse dataset of industrial metal objects characterized by symmetry, texturelessness, and high reflectivity, offering valuable insights for materials science and object recognition research.

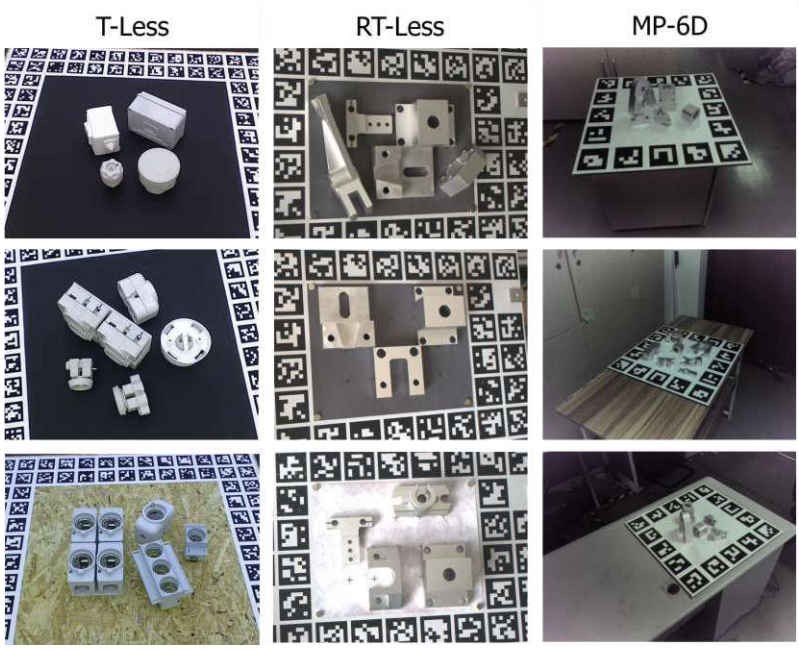

Figure 2.8: Overview of industrial datasets [10, 32, 64]

CHAPTER 3

Metallic Object Pose Estimation

This chapter presents the concept employed in estimating the poses of metallic, reflective objects. Based on the aforementioned related work and the methods presented therein, a workflow is developed that combines the strengths of general methods with the application-specific characteristics of techniques designed for metallic objects. First, the procedure is explained conceptually, resulting in a detailed explanation of the processes involved in data preparation.

3.1 Concept

Most of the presented methods for pose estimation 2.2.1 are based on extracting and matching features, which are extracted from the image of the object surface and compared with the known counterparts. The resulting estimates of 3D coordinates and subsequent 2D-3D correspondences lead to accurate pose estimates. However, these approaches fail in cases where the materials of the objects do not include suitable textures that provide such distinctive features. Therefore, the works that have specialized in metallic or textureless objects 2.2.2 employ an alternative approach: They utilize the contour of the objects for their pose estimation. The edges of an object promise to be a constant source of information due to its independence from texture and environmental influences. Since this approach has led to good results in various applications, the question arises of how using object contours can benefit the establishment of 2D-3D information for pose estimation.
This work proposes a model that predicts 2D-3D correspondences using edge images before addressing the challenge of pose estimation for metallic objects. It assumes that the entire scene is considered an edge image, and only this information is sufficient for estimating the object's pose. The underlying assumption is that the supposedly more straightforward edge detection process bypasses the difficulties of metallic surfaces. The necessary steps include object recognition, edge detection, and the subsequent prediction of 2D-3D correspondences with pose estimation.

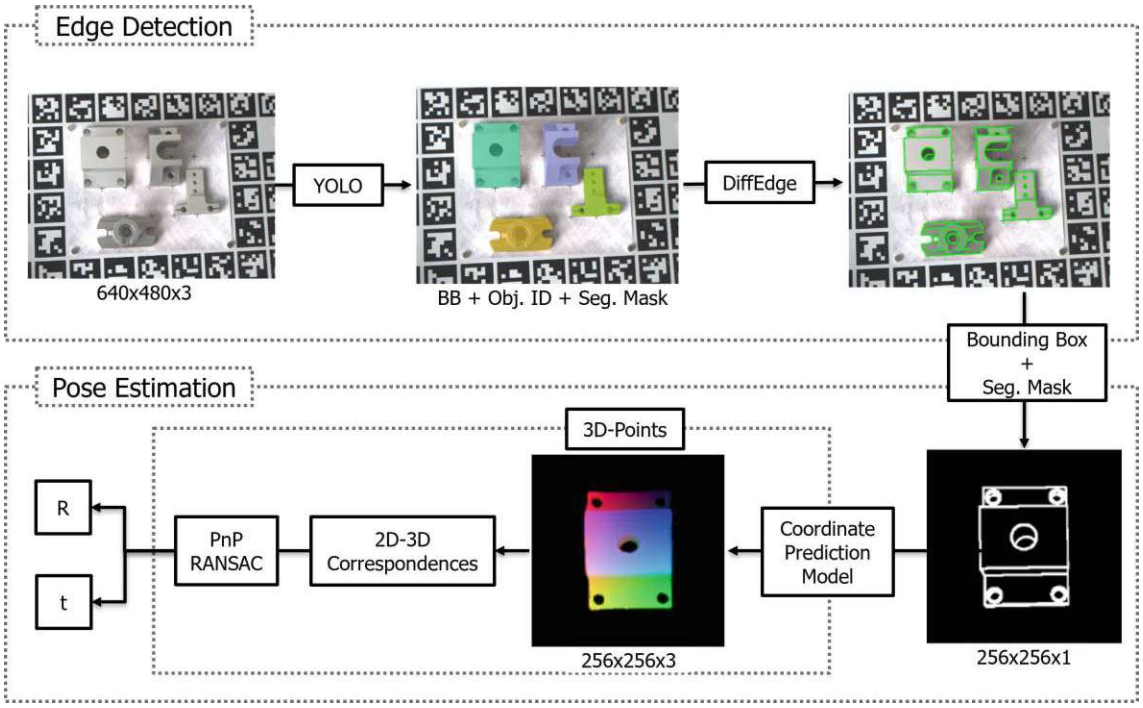

Figure 3.1: Pipeline for pose estimation of metallic objects. Edge Detection: The first phase is to detect objects and extract edge images. Pose Estimation: The second phase predicts the coordinate maps from the edge images and generates the 2D-3D correspondences with subsequent pose estimation.

The resulting pipeline for pose estimation of metallic, reflective objects is illustrated in figure 3.1. The entire workflow can be divided into two main components: "Edge Detection" and "Pose Estimation." The Edge Detection section covers the first phase of the pipeline, in which object detection and subsequent edge detection take place. In the first step, the 2D position of the object and its visibility within the obtained scene are determined. To do this, YOLOv8 [37], a method for object recognition and segmentation is applied. In addition to the 2D bounding box, this YOLOv8 [37] model also predicts the segmentation mask and ID of the targeted object. Next, the edge detection step is performed, for which DiffusionEdge [38] is utilized. This diffusion model generates an edge image based on the entire scene. The following steps isolate the object from the scene using the RoI principle. This process involves extracting the contour of the target object for focused analysis. The object is cropped along its scaled bounding box in a fixed size of 256 x 256 from the edge image. Only the contour of the visible object based on the segmentation mask is used to reduce unwanted edges. This cut-out is passed to the 3D coordinate prediction model. Its architecture is inspired by the methods of CDPN [31], DPOD [29], and Pix2Pose [25], in which the 3D coordinates of the model vertices are normalized to RGB color space values and subsequently predicted for pose estimation. The network provides the 3D coordinates based on the edge image of the object and, together with the 2D information from the previous steps, establishes the correspondences that are finally processed by PnP/RANSAC [39, 40] to estimate the object pose.
In light of the conceptual framework and approach delineated, the developed methodology will be designated as Edge2Pose within the context of this thesis. This term underscores the principle that estimating the object's pose is derived from edge-based imagery.

3.1.1 Edge Detection

This pipeline's first phase comprises object recognition and edge detection tasks. In this stage, information about the 2D object position, the visible part, and the object contour is provided for the subsequent pose estimation task. In this subsection, the YOLOv8 [37] and DiffusionEdge [38] methods applied are examined in detail, and their implementation within the pipeline is explicitly discussed.

Object Detection

To precisely estimate an object's pose, it is necessary to identify its location within the scene image. Instance segmentation comprises a more sophisticated methodology than traditional object detection. This technique is dedicated to determining individual objects within an image and outlining them from their surrounding environment. The output of an instance segmentation model consists of a collection of masks or contours that define the boundaries of each object, along with class labels and confidence scores for each identified object. This technique is particularly valuable when detailed information about the location and shape of objects within an image is essential [37, 66].

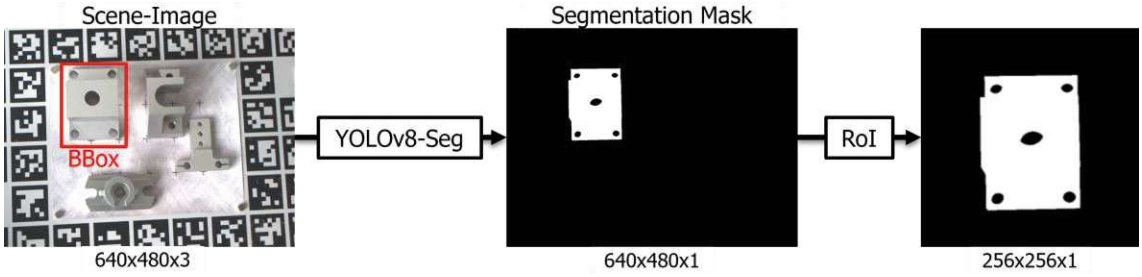

Figure 3.2: Overview of the Object Detection workflow

YOLO is widely known as a robust object detection algorithm. The latest version, YOLOv8 [37], represents the newest advancement in the YOLO series of object detectors, delivering exceptional performance in both accuracy and speed. The YOLOv8 [37] series offers a diverse range of models, each specialized for specific tasks in computer vision. These models are designed for object detection to more complex tasks like instance segmentation. The methodology adopted for this task involves first detecting the object within the scene and subsequently segmenting it. A specialized model from the YOLOv8 [37] catalog is utilized, specifically designed for object instance segmentation. The 'YOLOv8s-seg' [66] model is employed. The architecture of YOLOv8 [37] consists of two primary components: the backbone network and the detection head. The backbone network is designed to extract various rich features from the input image at multiple scales. Meanwhile, the detection head integrates these features to predict bounding boxes. The backbone network of YOLOv8 [37] is built upon EfficientNet [67], a cutting-edge neural network architecture recognized for its remarkable efficiency and performance across a range of computer vision tasks [37]. EfficientNet [67, 68] utilizes a concept known as compound scaling, which effectively balances the scaling of the network's width, depth, and resolution. The detection head of YOLOv8 [37] utilizes NAS-FPN [69], a search-based neural architecture method that automatically creates feature pyramid networks for object detection tasks. Feature pyramid networks integrate features from various levels of the backbone network to generate predictions across multiple scales.

The object detection model is trained on a synthetic dataset that showcases various objects under different lighting conditions and scenarios. This dataset encompasses the RGB scene images and segmentation masks highlighting the respective objects within those images. Only visible portions of the objects are included in these masks, ensuring that only the actual parts of the target objects are captured for later analysis. Applying the trained YOLOv8 [37] model to actual test data involves two primary tasks: detecting the object and generating segmentation masks, as illustrated in Figure 3.2. The object is extracted using the bounding box from the edge image based on the region of interest principle. The contour is subsequently refined using the segmentation mask, which exclusively considers the overlapping entity. The size of the extracted object is fixed at 256 x 256 pixels, with the object centered within this frame.

Edge Detection

The complex material properties of metallic objects often lead to reflections and textureless surfaces, significantly complicating pose estimation [1, 5–8]. This thesis addresses these challenges by outlining the contours of the objects and leveraging this information for pose estimation. The initial step involves transforming the scene image into an edge image using an advanced edge detection algorithm. Given that the subsequent algorithm for generating 2D-3D correspondences for pose estimation is critically dependent on the accuracy of the detected contours, it is crucial to utilize a highly robust and precise method during this phase. Conventional edge extraction techniques cannot reliably produce highly accurate and clear edge maps. Consequently, an approach called DiffusionEdge [38] is employed. This study illustrates that diffusion probabilistic models (DPMs) are particularly advantageous for edge detection, as the noise reduction process is applied directly to the original image, resulting in enhanced sharpness and accuracy of the edges detected.

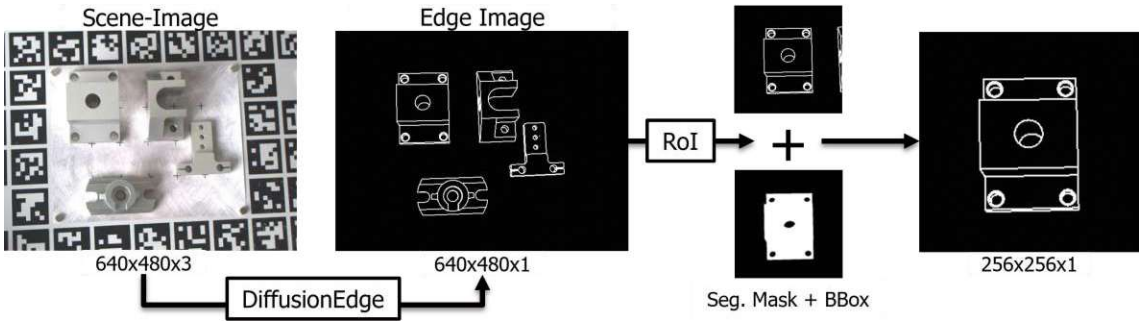

Figure 3.3: Overview of the Edge Detection workflow

DiffusionEdge [38] is based on a diffusion model initially developed for generative tasks. The network learns to gradually remove the noise in the image data, fitting the target distribution. Unlike conventional CNN-based edge extractors based on encoder-decoder architectures, which tend to have thicker edges, DiffusionEdge [38] performs noise reduction directly on the original image size. Most calculations are conducted in latent space to minimize computational effort. The model operates on images reduced to one-quarter of their original size. An adaptive Fourier filter [38] is added to analyze and adjust the frequency components of the image data. This technique improves edge extraction by filtering unwanted noise components in the frequency domain. A distinctive distillation methodology has been implemented since edge annotations frequently encompass uncertainties due to different annotators' marking variations. This method transfers the uncertainties to the latent space

and optimizes the gradients directly to provide a more stable and accurate prediction. DiffusionEdge [38] can generate edge images that are both accurate and sharp without relying on costly post-processing, such as non-maximum suppression. Combining these techniques makes it possible to train DiffusionEdge [38] with limited computational resources and still produce crisp and precise edge maps.
The Edge Detection section builds upon the previous Object Detection section. The DiffusionEdge network processes the RGB scene image as input to perform edge detection, systematically outlining the boundaries of all objects. As illustrated in Figure 3.3, the target object is analyzed using the predicted segmentation masks and bounding boxes, enabling its extraction from the edge image. The edge mask of the object is subsequently forwarded to the pose estimation phase. Furthermore, the DiffusionEdge [38] network is trained on a synthetic dataset that comprises a diverse array of rendered scenes. This dataset consists of scene images alongside their corresponding ground-truth edge images, in which white lines distinctly represent the boundaries of all objects. Moreover, this representation focuses on the observable edges of the objects, disregarding any hidden or occluded edges.

3.1.2 Pose Estimation

This section of the proposed methodology is dedicated to estimating the object pose. The selected approach integrates the principles of general pose estimation techniques 2.2.1 with insights from specialized research on textureless, metallic, and reflective objects 2.2.2. Using an object's contour has been shown to enhance pose estimation significantly. This concept comprises a critical component of the methodology presented in this thesis. As discussed previously, the edge image of an object is extracted from the scene image according to the RoI principle, as demonstrated in the works of GDR-Net [28], CDPN [31] and Pix2Pose [25]. The subsequent procedure for estimating the pose is likewise based on the general methods and primarily employs the concepts of the methods proposed [31], [25] and [29].
A general method for integrating the 3D points of a CAD model into machine learning processes involves encoding the model's surface coordinates in the RGB color space. Initially, the coordinates of each vertex (x, y, z) are normalized to a uniform value range of [0, 1]. This normalization is achieved by dividing the coordinates by the maximum extension of the object along each respective axis. The normalized values are then mapped to the color channels of an RGB image. This mapping process results in a 'color-coded image' where each pixel value corresponds to the 3D position of a point on the object's surface. This representation is advantageous as it allows for the processing of 3D information using standardized image processing tools and neural networks initially designed for RGB images [25]. The primary distinction between color-coded 3D coordinates and Normalized Object Coordinate Space (NOCS) [70] is their coordinate systems. Color coding normalizes original CAD model coordinates, while NOCS [70] uses a standardized unit cube for uniformity, aiding visualization of different object sizes. In contrast, color coding retains geometric details, enhancing pose estimation when CAD models are available [70].
In this approach, the object is extracted from the edge image through the use of its bounding box by using the techniques outlined in previous studies, such as [25], [31] and [28]. This method utilizes only the contained image to estimate the coordinates, ensuring precision and reliability. The dimensions of the bounding box, determined in the pipeline's initial phase, are employed to crop the object into a square mask. Subsequently, the bounding box is scaled to a fixed size of 256 x 256 pixels, and the object is centered within it. This results in a more efficient procedure than observing the entire scene due to smaller image sizes. To ensure that only the visible contours of the object are considered and to prevent inadvertently including overlapping portions of other elements in the scene, the edge image is refined using the segmentation mask. The edges indicated by the area of the segmentation mask are retained in the object section. The cropped section of the object now contains the edge image, which can be utilized as input for the coordinate model. After de-normalization of the RGB values, the predicted coordinate image provides the 3D coordinates of the individual model points of the object. Based on the edge image and the two-dimensional position of the object in the scene image, the model establishes the 2D-3D correspondences and, in conjunction with the camera matrix, calculates the final pose of the object using the PnP [39] algorithm and RANSAC [40].

Model Architecture

Most state-of-the-art methods primarily utilize encoder-decoder architectures, often in U-Net models. In this architecture, an encoder extracts features from the RGB input image while a decoder reconstructs the output as a coordinate image. The encoder used in [25] and [29] is a pre-trained ResNet [71] model. Initially optimized for RGB images, the current decoder has limitations when applied to other input types. A new encoder, which utilizes gray-scale images, is essential to process edge images effectively.
Figure 3.4 shows an overview of the applied model. The network encoder hierarchically extracts features from the input image. A double convolutional block initially processes the input image, followed by a sequence of successive convolutional layers, batch normalization, and ReLU activation functions. These blocks are complemented by max-pooling layers that systematically reduce the spatial resolution of the feature maps while preserving the most salient features, thereby enhancing computational efficiency and resilience to variations in image resolution. Residual blocks enhance neural network depth through internal skip connections, aiding in stable gradient flow and mitigating vanishing gradient issues during training.

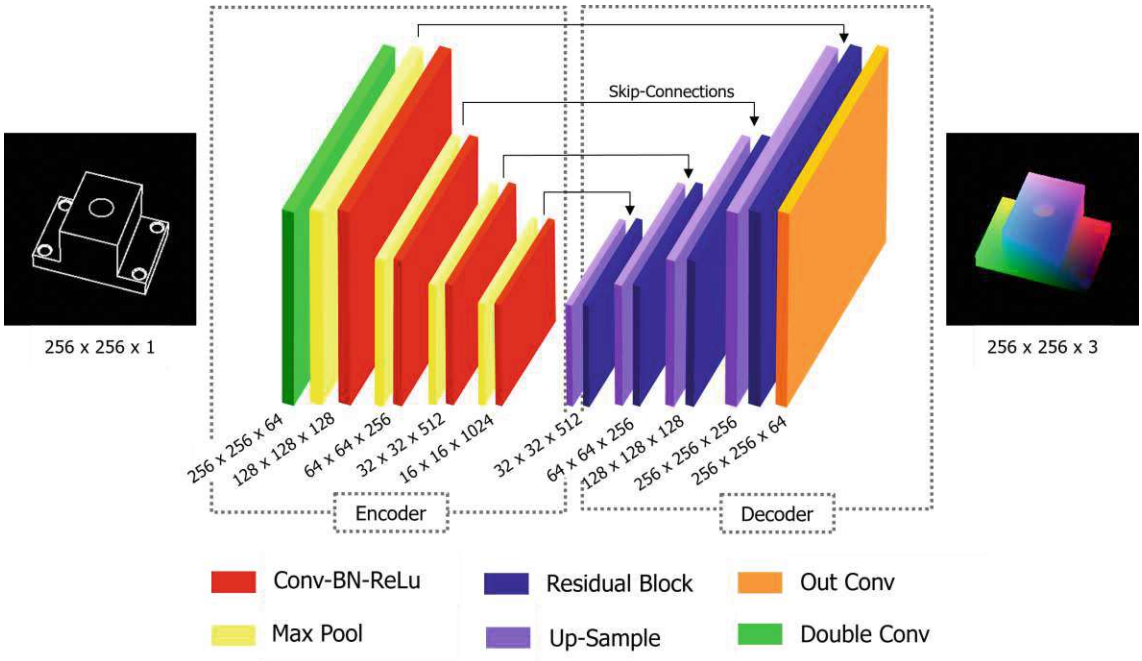

Figure 3.4: The architecture of the coordinate-map prediction network. The encoder consists of double convolution and a sequence of convolutional layers, batch normalization, ReLU activation, and max-pooling blocks, followed by the decoder consisting of up-sampling and residual blocks with a final output convolution.

The decoder architecture is derived from the DPOD model [29], where the RGB color value prediction is divided into three channels, each comprising a range of 0-255 values. The three correspondence heads regress tensors with dimensions HWC, where C is the number of unique colors in the correspondence map. In this context, "H" and "W" refer to the height and width of the input image, respectively. The probability values for the class corresponding to the channel number are contained within each output tensor channel. Subsequently, the tensors are stored as single-channel images. Each pixel represents the class with the maximum estimated probability, generating the correspondence image's

channels U, V, and W. As the authors of [29] have demonstrated, formulating the color regression problem as a discrete color-classification problem has proven to be a practical approach, facilitating faster convergence and improving the quality of 2D-3D matches. The model is trained on image pairs comprising edge images and the corresponding coordinate images of the objects in question. The network parameters are optimized considering the composite loss function:

$$L_{uvw} = \alpha L_u + \beta L_v + \gamma L_w \tag{3.1}$$

where L_u, L_v, and L_w are the losses responsible for the quality of the U and V channels of the coordinate image. α, β and γ are weighting factors. The L_u, L_v, and L_w losses are defined as multi-class cross-entropy functions. The transformer loss, as detailed in Pix2Pose [25], is incorporated into the existing loss function to optimize the recognition of symmetric objects. A set of poses is defined for each symmetric object that is identical in appearance on either side. Rather than calculating the discrepancy between the predicted pose and the actual ground truth pose, the transformer loss computes the difference between the predicted and the symmetric poses, thereby identifying the pose with the smallest error. The transformer loss is defined as follows:

$$L_{\text{3D}} = \min_{p \in \text{sym}} L_(I_{\text{3D}}, R_p I_{\text{gt}}) \tag{3.2}$$

R_p represents a transformation to one of the symmetric poses from the symmetry set *sym*. The loss is calculated for each symmetric pose, and the smallest loss is selected. The total loss is now the combination of the two losses, whereby their respective influence can be set via the weighting:

$$L_{\text{combined}} = \alpha L_{uvw} + \beta L_{\text{3D}} \tag{3.3}$$

3.2 Data Preparation

This thesis focuses on pose estimation of industrial, metallic, reflective objects for which application the RT-Less dataset [10] is used. The provided CAD models are utilized to generate the respective training data for each stage of the pipeline 3.1. The test data and ground-truth pose annotations are taken from the RT-Less dataset [10].

3.2.1 RT-Less Dataset

The RT-Less dataset [10] offers a collection of reflective, metallic objects focusing on industrial parts and scenarios. The dataset contains 3D models of 38 metal parts covering typical features, e.g., large areas, surfaces, chamfers, and circular holes. All objects in the dataset originate from the metal parts processing plants to ensure the authenticity of the industrial attributes of objects. These industrial objects have a strong reflectance and no regular texture. During machining, particular objects undergo chamfering, developing a more intricate structure. Numerous parts were designed with high similarity to replicate the subtle shape variations typical of different components in the manufacturing industry. All items in this dataset are derived from actual production lines and have been machined using standard industry practices.

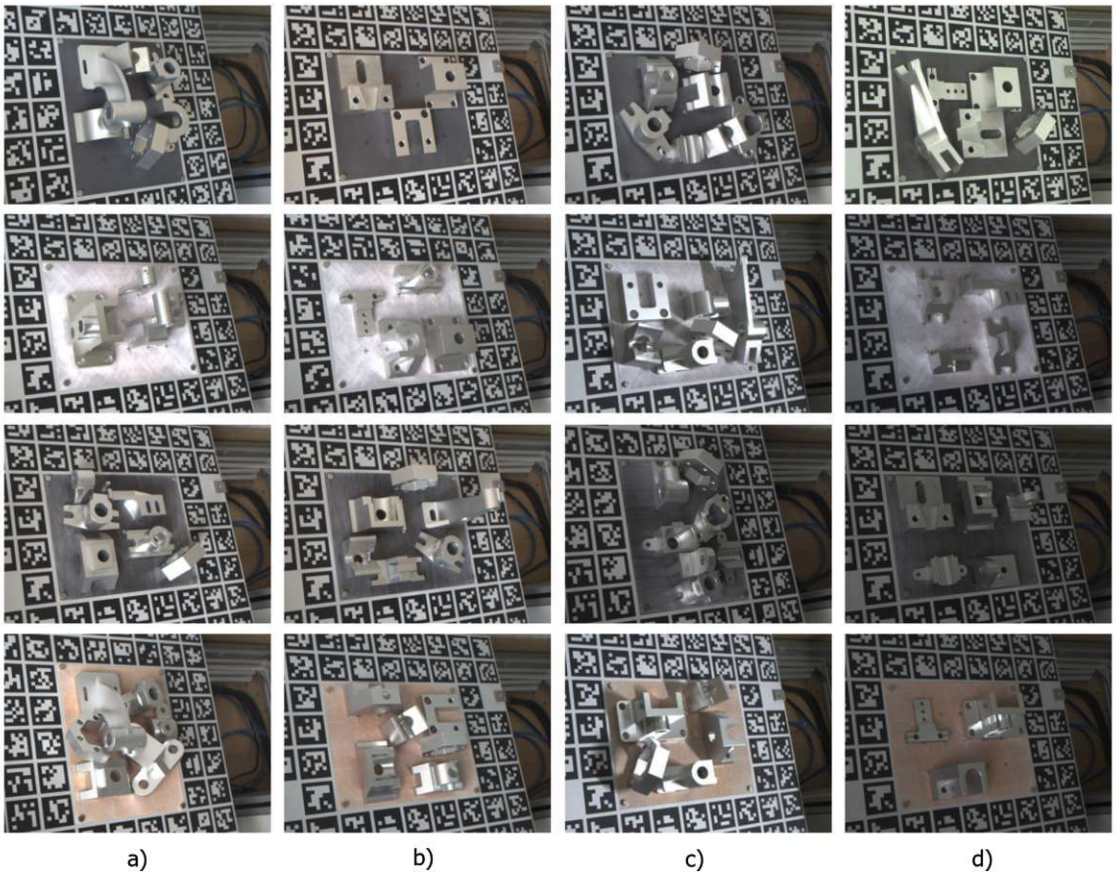

Figure 3.5: Samples of RT-Less [10] test scenes. The images in each row exhibit uniform background textures—matte, reflective, textured, and rusty—while the columns a) through d) vary in lighting conditions.

The RT-Less dataset [10] comprises a collection of reflective metallic objects, explicitly concentrating on industrial components and scenarios. It features 3D models of 38 metal parts that exemplify key characteristics such as expansive surfaces, chamfers, and circular holes. The industrial components of this data set possess high reflectivity and lack regular texture. To ensure the authenticity of the industrial attributes, each in this dataset originates from actual production lines and has been produced using established industry standards. Throughout the machining process, particular objects were chamfered, resulting in more complex structures. Many parts were crafted with close similarities to capture the subtle shape variations in the manufacturing industry. Industrial scenarios are carefully recreated to reflect diverse factors such as part placement, part types and shapes, lighting settings, and backgrounds. Given the relatively straightforward characteristics of machined parts, handling similar parts is a standard application in actual industrial tasks. Consequently, the test set includes several similar parts with identical attributes in certain scenes to simulate realistic conditions closely. As illustrated in Figure 3.5, a selection of the test scenes is presented, all meeting the necessary criteria. The test scenes primarily differ in the following aspects: lighting conditions, the number of sampled objects, and background materials. In addition to natural lighting a) - b), certain scenes were captured with artificial overexposure c) and others with a less powerful light source d). The number of objects varies throughout the scenes, as does the size of the objects collected to simulate overlaps. There are also interfering objects that are not part of the models included in the dataset. Four different background types are used within the scenes to simulate the environment's influences. These are backgrounds with the following properties: reflective, matte, textured, and rusty.
The image acquisition setup comprises an MV-CA050-11UC industrial camera, a MELFA RV13FD 6-DoF manipulator, a turntable, and a pose calibration board positioned above the turntable. The industrial camera is integrated with the manipulator in an eye-in-hand configuration, enabling the capture of realistic images synchronized with the manipulator's motion. This configuration facilitates precise spatial alignment and improves the accuracy of image data acquisition in dynamic environments. To ensure that all target parts are captured within the camera's field of view, the manipulator is controlled to enable the camera's center to traverse along a quarter-section of a spherical space, with the turntable's rotation center serving as the sphere's center. Three spherical spaces are established, with diameters of 750 mm, 800 mm, and 850 mm, with a spherical angle of 130°. During the acquisition of test images, the manipulator's motion space comprises 104 points distributed across different layers of the quarter-spherical spaces. The turntable rotates around the sphere's center to complete these layers, resulting in a comprehensive 360-degree view.

3.2.2 Data Generation

As described in the earlier concept, specific datasets are required for the two stages of the pipeline to train the respective methodologies. As a result, a comprehensive rendering pipeline is developed to generate the requisite ground-truth data. This pipeline leverages the capabilities of BlenderProc [34] alongside the RT-Less Toolbox [35]. The integration of these tools enables precise simulation of scenarios, ensuring the reliability and accuracy of datasets crucial for analytical investigations. This methodology utilizes Blender's 3D modeling software [33] and Python API to create photorealistic scenes and generate coordinate maps with corresponding ground-truth edge masks. This advancement enhances visual accuracy and provides a solid computer vision and graphics application framework.

Edge Detection

The RT-Less [10] dataset provides synthetic training data for object detection. However, it is unsuitable for training edge detection methods due to the absence of ground truth edge masks. The primary objective of the "Edge Detection" phase is to locate objects and identify edges in a scene accurately. This task requires developing suitable training and testing data with synthetic images representing various real-world conditions to enhance detection. The rendering pipeline must effectively simulate realistic material properties for objects and background elements, considering lighting configurations and object arrangements. This objective requires diverse setups for photorealistic imagery to emulate light and material interactions.

Figure 3.6: Samples of synthetic training data with RT-Less [10] models. From left to right, the RGB scene image, edge images, and segmentation masks. The individual masks are color-coded for improved clarity.

The image generation process closely mirrors the image acquisition techniques utilized in the RT-Less [10] dataset. During this process, the Blender camera moves along predefined points arranged on a hemisphere. The center of this hemisphere is strategically positioned so that, at each point, the virtual camera consistently focuses on the origin of the Blender world coordinate system. The number of images required influences the number of points evenly distributed across the hemisphere per the Fibonacci Theorem. The sphere has a radius of 1 meter relative to the origin of the coordinate system. The camera matrix and image dimensions are derived from the intrinsic camera parameters of the RT-Less [10] dataset. The scenes are created using an internal environment in Blender, which includes a base plate for the background, light sources, and the objects themselves. The type, position, and intensity of light sources are randomly generated and change with each new camera angle. This setup accommodates both overexposure and underexposure, as well as steep and subtle illumination angles. Consequently, the shadows cast by the objects, along with reflections and mirroring, also vary accordingly. The materials used for the background and models are sourced from the BlenderProc texture catalog, ensuring realistic surface properties. Similarly to the lighting, the textures of the background and objects are randomized every time the camera is repositioned. The arrangement of objects in each scene is randomized. Several models are selected and arranged on the base plate at the start of each rendering cycle. These models are then repositioned for each camera view within the current cycle, with random rotations and translations applied. The distance between objects is controlled to ensure occlusions occur within the scene.
In addition to generating photorealistic scene images, edge images, and segmentation masks are created. Consecutive images are generated for each camera position to maintain consistency within the current scene. The Blender [33] module "Freestyle" captures the edges in the scene. This tool highlights edges with distinctive colors by defining a threshold value. This threshold value delineates the minimum angular separation required between two surfaces to identify their intersection as a distinct edge. Edge images are created by applying a black texture to all models and the background while highlighting visible edges with white lines. This technique effectively outlines the contours of objects within the camera's field of view. Segmentation masks are generated by alternately applying white and black color shaders to objects during the rendering step. The mask image corresponds to the object highlighted in white. This methodology enables the assessment of model visibility in the current scene through segmentation masks. Figure 3.6 displays selected rendered scenes, along with their edge images and segmentation masks. For clarity, all segmentation masks for the corresponding objects are compiled into a single image.

Pose Estimation

The second component of the method, pose estimation, requires an additional dataset for training. This approach builds upon the methodologies presented in [25, 29, 31], where the 3D coordinates of all the vertices of the model are transformed into the RGB color space and subsequently mapped onto an image. To implement this approach, training images consisting of edge-coordinate image pairs are generated. The objective can be achieved by modifying the established rendering pipeline to adapt an alternative workflow.
In contrast to creating scene images, this dataset treats individual objects separately. In addition, the use of backgrounds and lighting is not required. Unlike the distribution of camera viewpoints on a hemisphere, this approach utilizes the entire surface of a sphere. The sphere is generated with its center at the origin of the Blender world coordinate system, and the desired number of viewpoints is evenly distributed across its surface. The respective object is then loaded into the Blender scene with the model centered in the origin of the coordinate system. The 3D coordinates are applied to the model through a customized vertex shader designed for it. This shader normalizes the x, y, and z coordinates of each point on the model and converts them into an RGB color value. As a result, each model vertex is assigned a distinct color, allowing it to be identified by its position. The object is captured from every angle, and a coordinate map is generated for each viewpoint. Blender settings are configured to utilize a lossless format during the image rendering. The shader is devised to guarantee no discrepancy between the colors assigned to the vertices and the final rendered colors of the image. In addition, edge images are produced using the Freestyle module to align with the corresponding viewpoint.

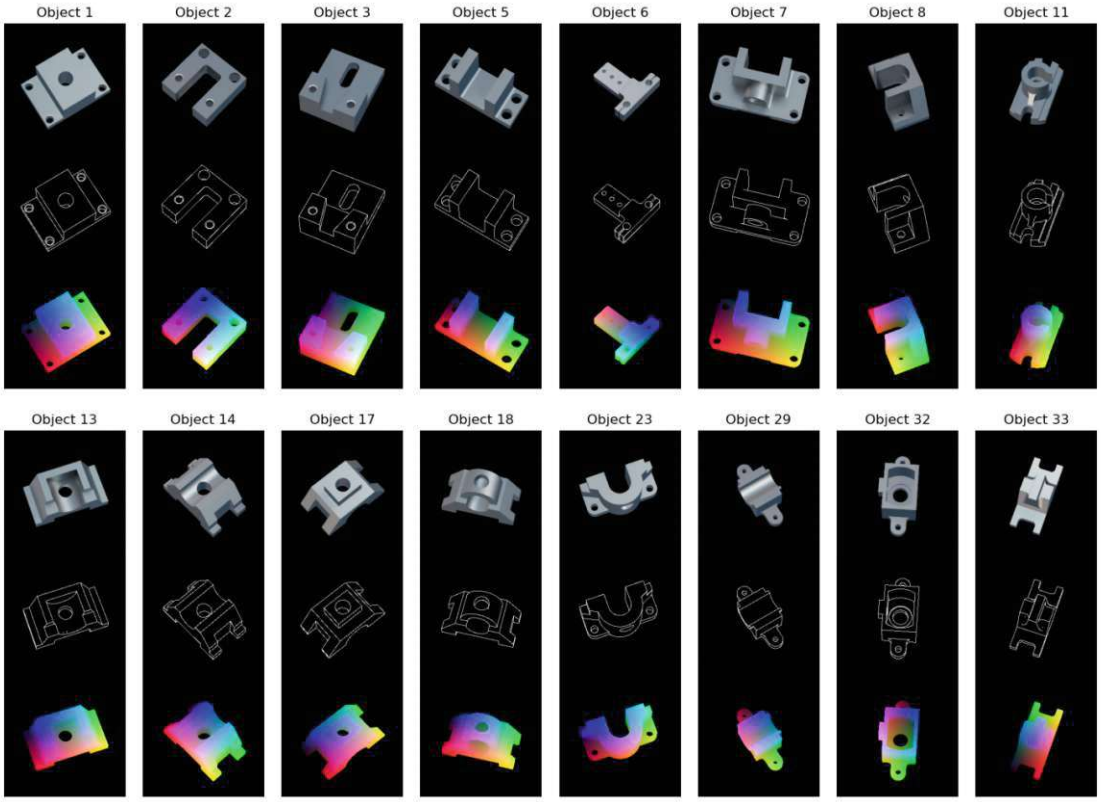

Figure 3.7: RT-Less models [10] rendered as edge-coordinate image pairs. A model processes an edge image alongside a coordinate map of color-coded 3D values for each viewpoint.

CHAPTER 4

Experiments

This chapter explores the implementation and testing of the method discussed in section 3 for estimating poses of metallic reflective objects. The components of the proposed pipeline are applied to test scenes from the RT-Less [10] dataset, with results thoroughly examined. This methodology is extended to include other datasets, such as T-Less [32] and MP-6D [64], highlighting its versatility.

4.1 Evaluation on RT-Less Dataset

The RT-less [10] dataset introduced in Chapter 2 serves as the foundation for the primary application of this method. In the subsequent sections, the components for object detection, edge detection, and pose estimation are trained using the synthetic training data outlined in 3.2. Their performance is assessed on the actual test scenes from the RT-Less [10] dataset. A comparative analysis is conducted to enhance the understanding of the results using current state-of-the-art methodologies.
A comprehensive analysis is performed on a subset of objects from the RT-Less [10] dataset to evaluate the effectiveness of Edge2Pose in real-world environments. The chosen objects are depicted in Figure 3.7. This selection captures the wide range of challenges posed by the diverse array of objects. It includes items with multiple symmetries, such as objects 1, 14, 17, 18, and 33, and those with distinctive reflections along their curved surfaces, including objects 7, 13, 14, and 29. The selection features objects with flat elements and textureless surfaces, such as objects 3, 5, and 8. Similar geometric characteristics are represented in objects 14, 17, 29, and 32. In addition, the selection encompasses objects with more complicated and complex geometric properties, like those seen in 7, 8, 11, 13, and 23, alongside simpler geometric shapes found in 2, 3, 6, and 33. The objects exhibit significant size variability. Models 7 and 8 have larger diameters of 130 to 145 mm, while items 2, 6, and 29 are notably smaller, ranging from 90 to 115 mm.
The evaluation of the efficacy of the Edge2Pose is structured into three sections, each focusing on a specific task within the pipeline. First, the performance of object recognition and segmentation processes is assessed. This is followed by an analysis of the network's edge detection capabilities. Finally, the network's ability to generate coordinate maps and perform precise pose estimation is evaluated.

4.1.1 Edge Detection

The evaluation of the first stage, "Edge Detection," focuses on the accuracy of object detection and segmentation, along with the precision of the detected edges. Additional evaluation metrics are introduced to fulfill this purpose. The Intersection over Union (IoU), along with Precision, Recall, and the resulting F1-Score, are commonly used metrics for assessing the outcomes of segmentation or classification tasks [2, 36, 38].
The IoU is a crucial image segmentation and object detection metric. It measures the extent of overlap between predicted and actual classifications concerning their union. In object detection, IoU is used to assess the accuracy of a predicted bounding box against the ground truth. In the context of semantic segmentation, it quantifies the similarity between a model's segmentation mask and the ground truth. Typical thresholds, such as 0.5 or 0.75, are usually employed to determine a "valid" prediction. The IoU metric accounts for false-positive and false-negative pixel values, offering a more comprehensive evaluation [2, 36]. The IoU is calculated using

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{4.1}$$

where $|P \cap G|$ is the number of pixels marked as part of the target class in both the ground truth G and the predicted mask P. $|P \cup G|$ is the number of pixels labeled either in G or in P as the target class.
Precision quantifies the accuracy of positive predictions, specifically defined as the ratio of correctly predicted positive pixels to all predicted positive ones. A high precision value suggests the model is reliable and generates fewer false positives. However, precision can be a somewhat misleading metric if the model adopts a conservative prediction approach, identifying only a limited number of positive pixels [2, 38].

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Therein, TP denotes the accurate positive predictions, and FP the false positive predictions. Recall measures a model's ability to identify valid positive pixels, calculated as the ratio of correctly predicted positives to the total positives in the ground truth. A high recall value indicates that the model is sensitive and capable of recognizing the majority of relevant pixels. However, it should be noted that the recall metric can be misleading if the model tends to make an excessive number of positive predictions, which can subsequently result in a high false positive rate [2, 38].

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

This equation contains FN as false negative predictions. The F1-Score is defined as the harmonic mean of Precision and Recall. This relation is employed to achieve a balance between the two metrics. The F1-Score ensures that neither Precision nor Recall is entirely neglected. The F1 score ranges from 0 to 1, with 1 indicating a perfect match [2, 38].

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.4}$$

Quality of Object Detection and Segmentation

This proposed pipeline uses the YOLOv8 [37] model for object detection and segmentation tasks. Synthetic training data is generated to implement this model, as detailed in the previous chapter 3.2. This dataset consists of 30,000 scene images, each with a 640 x 480 pixels resolution, containing various RT-Less [10] objects arranged in diverse configurations. To enhance the diversity of the dataset, metallic materials, and backgrounds are randomly selected for each image. The position, intensity, type, and position of the light source are varied based on predefined criteria. Segmentation masks are created for individual objects along with RGB scene images. These masks highlight only the visible parts of the objects within each scene. To fulfill the requirements for training the YOLOv8 [37] model, label files containing polygons, bounding boxes, and unique identifiers are generated simultaneously for all scene images. The 'yolov8s-seg' model is selected from the provided YOLO repository and trained on the synthetic data. The dataset is enhanced through data augmentation techniques, precisely vertical and horizontal flips, resulting in 90,000 images. An 80:10:10 split divides the dataset into training, testing, and evaluation subsets. Data augmentation incorporates adjustments to the brightness and contrast of the scene images. The training process spans 50 epochs, employing the AdamW optimizer with an initial learning rate of $1e^{-4}$ and a batch size of 16.

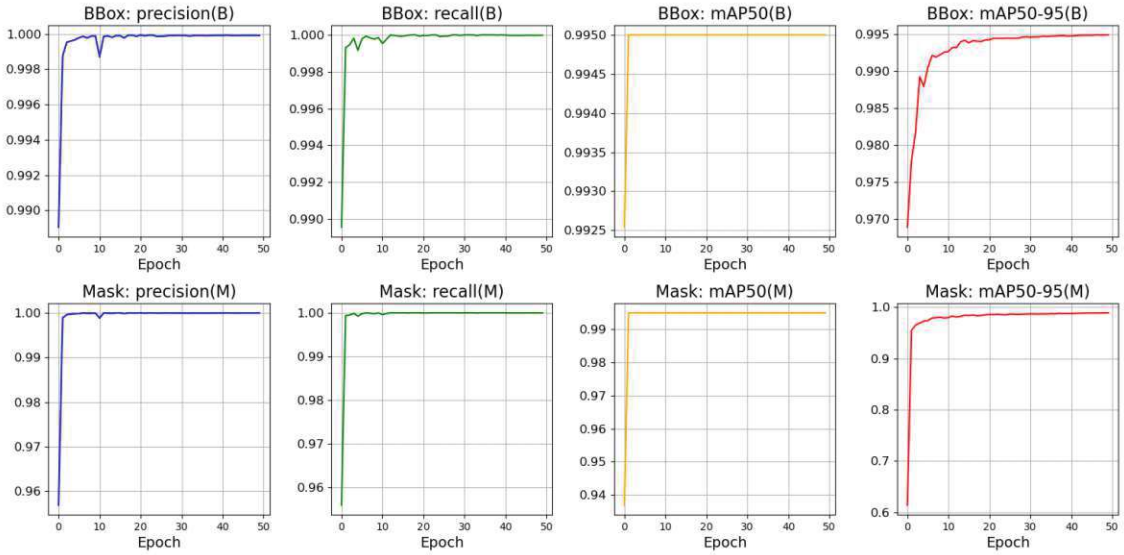

Figure 4.1: Metrics for YOLOv8 [37] training on synthetic data. The metrics for bounding boxes and segmentation masks use color coding: blue for precision, green for recall, yellow for mAP50, and red for mAP50-90.

These metrics include general evaluation parameters such as Precision, Recall, and the Mean Average Precision (mAP). Precision measures the certainty of the model's predictions concerning the actual presence of objects in the scene, while Recall evaluates the model's ability to identify relevant objects. The most crucial metric for YOLOv8 is the mAP, calculated in two variants: mAP and mAP@0.95. The former evaluates the mean accuracy at an IoU threshold of 0.5, which reflects the proportion of correctly localized objects with sufficient overlap. The calculation begins with generating a precision-recall curve for each object and class. Following this, the mean precision values across all recall levels are computed. This procedure is repeated for each class and obtained by averaging the values across all classes, resulting in the overall mAP. The mAP@0.95 extends the mAP metric by varying the IoU threshold from 0.5 to 0.95 in increments of 0.05. This approach enhances performance evaluation by establishing stricter match validity criteria. As a result, it offers more profound insights into model effectiveness in object detection tasks. When interpreting the results shown in Figure 4.1, it is essential to recognize that the mAP metric evaluates the model's ability to recognize and localize objects. In contrast, the mAP@0.95 metric assesses its ability to recognize and segment objects precisely.
The segmentation model is trained and evaluated using synthetic data, and its performance is further assessed with actual test data from the RT-Less [10] dataset. Predictions for these image scenes are generated and verified against the provided segmentation masks. The results of this assessment are shown in Table 4.1. In addition, Figure 4.7 showcases an example of the predicted segmentation masks overlaid on a real RT-Less [10] test scene. In contrast, Figure 4.2 illustrates the individual predicted masks for the same scene alongside the ground-truth masks.

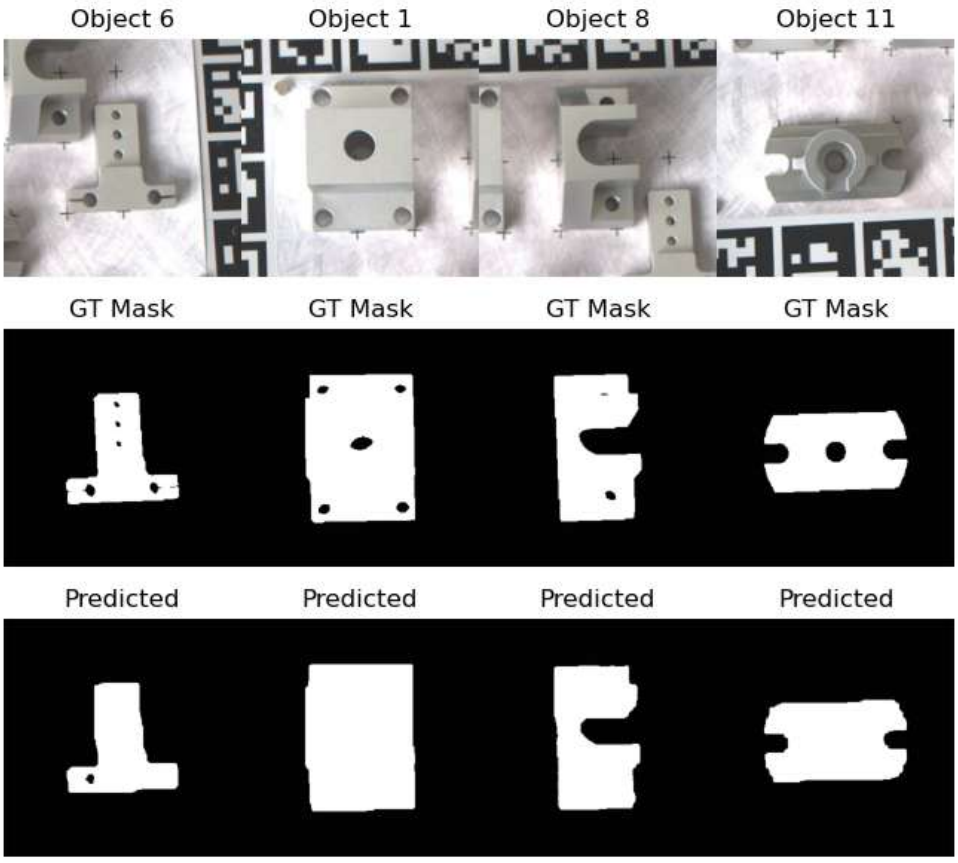

Figure 4.2: Comparison between ground truth and predicted masks

Table 4.1 presents the test results of the trained object detection and segmentation model. The evaluation is conducted using genuine test scenes sourced from the RT-Less [10] dataset. Each object is individually evaluated to assess the model's performance across various scenarios. Subsequently, the mean value of each metric is computed to provide an overall performance assessment.
The results demonstrate a high level of accuracy, with scores that closely correspond to those obtained during the synthetic testing phase of the training process. One possible explanation for the minor discrepancies in the observed values is that real scenes generally possess a higher density of objects, which can lead to increased occlusion for individual objects. This fact results in slight variations in the precision of the predicted segmentation masks compared to the ideal outcome. As illustrated in Figure 4.2, the predicted masks exhibit a notable limitation in accurately representing the holes and bores within the objects, which results in lower precision values. However, this shortcoming has a minimal effect on subsequent processes, as the segmentation masks are primarily used to refine the outer contours of the object within the RoI. It can be concluded that the test produced positive results and that neither the diverse appearances of materials nor the variations in lighting and backgrounds significantly impacted the outcomes.

Obj ID	Precision	Recall	F1-Score
1	0.907	0.941	0.924
2	0.902	0.957	0.929
3	0.881	0.987	0.931
5	0.914	0.962	0.937
6	0.886	0.917	0.901
7	0.928	0.933	0.931
8	0.910	0.972	0.940
11	0.884	0.896	0.890
13	0.902	0.983	0.940
14	0.881	0.967	0.922
16	0.920	0.931	0.925
17	0.910	0.934	0.922
18	0.901	0.922	0.912
21	0.935	0.941	0.938
23	0.895	0.903	0.899
29	0.896	0.943	0.919
32	0.929	0.953	0.941
33	0.901	0.939	0.919
AVG	0.904	0.943	0.923

Table 4.1: Results of predicted segmentation masks on real RT-Less [10] scenes

Quality of Detected Edges

As the conceptual framework outlines, accurately predicting the contour of an object is essential for pose estimation. To achieve this, the DiffusionEdge model [38] is used. The training of DiffusionEdge [38] utilizes the generated synthetic training data. This dataset extends the previously discussed dataset by incorporating edge images instead of segmentation masks. It is divided into training and test sets, following an 80:20 ratio to ensure a robust evaluation of the model's performance. The AdamW optimizer is applied during training, with a damped learning rate varying from $5e^{-5}$ to $5e^{-6}$ across 40,000 iterations, employing a batch size of 8. An exponential moving average (EMA) addresses unstable model performance during training. Image patches measuring 240 x 240 pixels are processed during inference, and overlapping regions are averaged to produce the final results.

The accuracy of the predicted edge images is determined using the Precision, Recall, and F1-Score, as previously described, to evaluate the segmentation masks. Test scenes incorporating the segmentation masks and edge images of real RT-Less [10] scene images are utilized to evaluate the system's performance. The RT-Less dataset [10] lacks ground truth edge images for real scenes. A custom rendering pipeline arranges objects within the scenes according to their corresponding ground truth poses. This process generates crucial ground truth data needed for evaluation.

The edge images obtained during the 'Edge Detection' phase are compared to those of the corresponding ground truth images. Each target object is evaluated individually across all test scenes in the dataset, and the relevant assessment metrics are calculated. The mean values of these recorded metrics are presented in Table 4.2.

Obj ID	Precision	Recall	F1-Score
1	0.793	0.862	0.826
2	0.842	0.846	0.844
3	0.777	0.819	0.798
5	0.860	0.874	0.867
6	0.766	0.814	0.789
7	0.801	0.843	0.822
8	0.769	0.789	0.779
11	0.863	0.867	0.865
13	0.759	0.872	0.812
14	0.845	0.872	0.858
16	0.898	0.881	0.889
17	0.858	0.871	0.864
18	0.822	0.832	0.827
21	0.838	0.854	0.846
23	0.781	0.784	0.782
29	0.835	0.843	0.839
32	0.873	0.878	0.876
33	0.827	0.859	0.843
AVG	0.822	0.847	0.834

Table 4.2: DiffusionEdge results real RT-Less scenes

The findings presented in this table demonstrate that the DiffusionEdge [38] model, trained on synthetic scenes, performs remarkably well in edge detection within real RT-less [10] test scenarios. The average values across all scenes indicate that the predicted edge images closely align with the ground truth masks. It can be inferred that the model effectively distinguishes between different objects and navigates the challenges posed by metallic properties, allowing for stable and precise edge prediction. The gap between the training data and real-world scenarios underscores a fundamental limitation of this testing methodology. As it is impossible to encompass the entire spectrum of potential outcomes within the dataset, some deviation from an ideal result is unavoidable. However, this training method can still produce satisfactory outcomes within the model's context. The following section offers a comprehensive analysis of edge cases. Figure 4.3 presents an example of the objects identified within this scene through their cut-outs, comparing them with the ground truth edges alongside the edges predicted by DiffusionEdge [38].

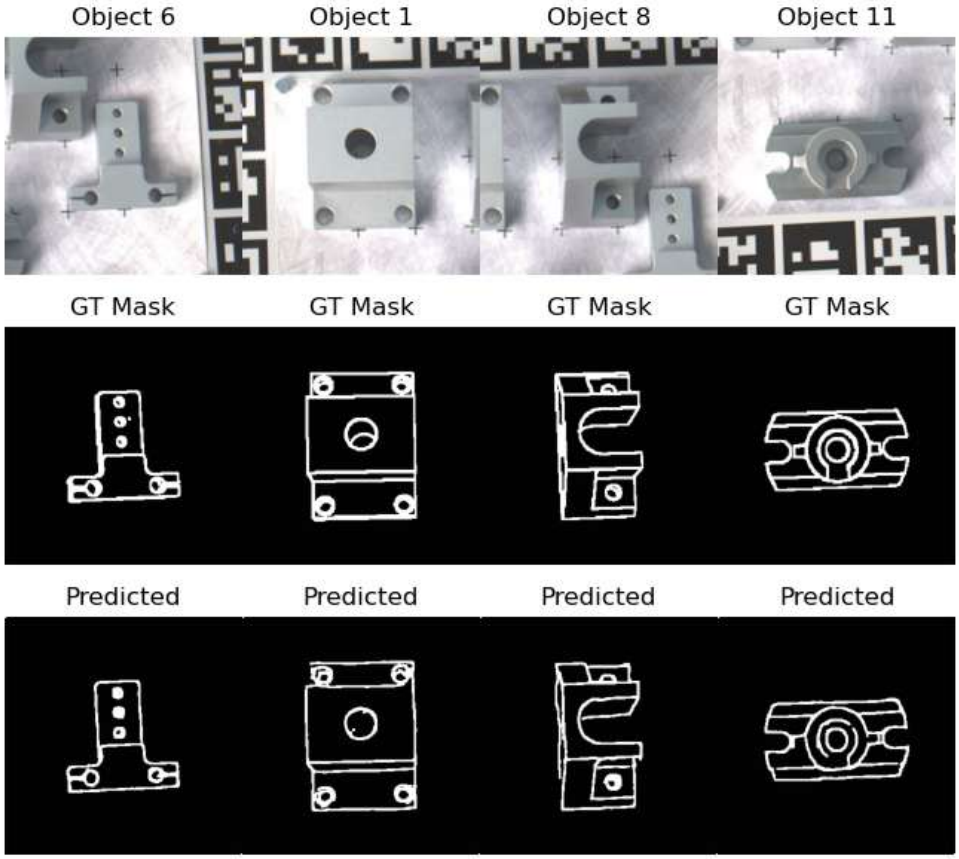

Figure 4.3: Ground truth and DiffusionEdge predicted edges on RT-Less [10]

4.1.2 Pose Estimation

After thoroughly analyzing the pipeline's initial phase, this section focuses on the second phase: "Pose Estimation". First, the performance of the proposed network in predicting coordinates is evaluated. Next, the capability of Edge2Pose in estimating poses is assessed. Finally, the results are compared with those of the state-of-the-art methods.

Evaluation of Coordinate Prediction

The original network proposed by [29] utilizes a ResNet [71] model for feature extraction within the encoder layer. This architecture is also observed in the works of [25, 28, 31]. These studies highlight the model's dependence on RGB images as input. However, since the proposed approach does not depend on RGB image data of the object of interest but instead uses edge images, it is necessary to adjust the encoder layer of the network accordingly. As discussed in Section 3.1.2, the decoder layer remains specify unchanged compared to the original structure.
As an ablation study, the encoder is kept in its original configuration, utilizing a pre-trained ResNet-34 model for feature extraction. However, skip connections between the encoder and decoder are incorporated, following the approach outlined in [25], to enhance information flow and mitigate gradient vanishing issues. In addition, the transformer loss proposed by [25] is integrated into the existing loss function. The data loader has been adapted to preprocess gray-scale edge images for input into the ResNet architecture [71]. This is accomplished by tripling the channels of the gray-scale images to conform to the expected input format of the model. The training uses the rendered single-object dataset described in Section 3.2, comprising 6,600 edge-coordinate image pairs for each object. Several augmentation techniques are employed, including random zoom-ins, cut-outs, noise along the edges, and variations in edge intensity to increase the diversity of the dataset. The AdamW optimizer is applied with an initial learning rate of $1e^{-4}$ and a batch size of 8, running for 25 epochs.

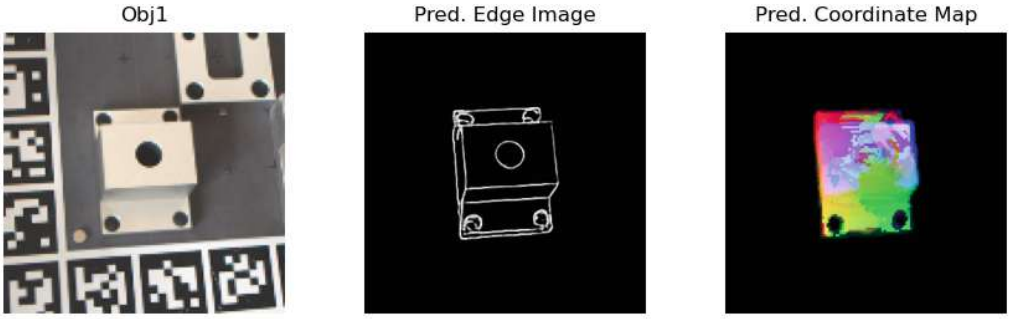

Figure 4.4: Exemplary results of coordinate prediction with ResNet [71] encoder

Upon testing the model with RT-Less test scenes, it became clear that consistently reliable predictions could not be achieved despite the promising results observed during the training phase. These results indicate that utilizing a ResNet model as the encoder is ineffective and inappropriate for the proposed approach. This limitation primarily arises because these models are specifically designed to extract features from RGB-format images. Without the required input, the feature extraction process encounters significant challenges, leading to outcomes that fall short of expectations.

Modifications are applied to the encoder layer of the network, while the decoder layer for predicting coordinate maps remains unchanged. Figure 3.4 illustrates the refined encoder in alignment with the model architecture. The updated model is trained on the dataset of rendered edge-coordinate images, each measuring 256 x 256 pixels, using the AdamW optimizer with a learning rate of $1e^{-4}$ and a batch size of 8. Augmentation techniques are implemented to enhance the dataset further, including random zoom-ins, cut-outs, edge noise, and variations in edge intensity. The dataset is divided into a training set and a test set in an 80:20 ratio. The training process is completed after 25 epochs.

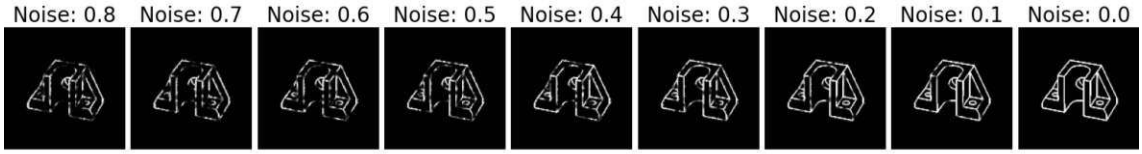

Figure 4.5: Qualitative display of noise factors on input images for MAE evaluation

The quality of the predicted edges is assessed using the Mean Average Error (MAE) metric. This metric quantifies the mean absolute difference between two images, allowing an evaluation of the discrepancies between the predictions. The MAE serves as an indicator of the average variation in pixel values between the two images. In this context, the pixel color values can be understood as a representation of the three-dimensional coordinates of the model's vertices. New unseen pairs of edge coordinate images are generated for each object to evaluate the network's performance. The input images contain noise along the edges and black spots that obscure the edges. The noise factor indicates how much noise or overlapping elements influence each edge pixel in the image. When the factor value is set at 1.0, every edge pixel is impacted by noise or overlap 4.5. The experiment begins with a noise factor of 0.75, decreased by 0.05 at each subsequent iteration until reaching an image with minimal disturbance, corresponding to a factor of 0.05. The results are illustrated in Figure 4.6 for all objects.

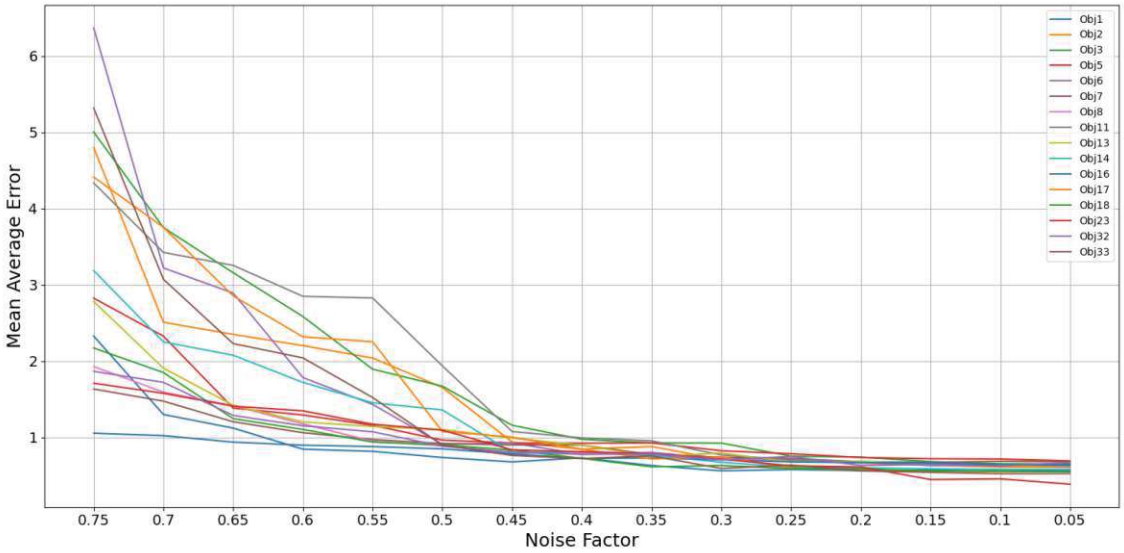

Figure 4.6: MAE of predicted images over noise factors

As depicted in Figure 4.6, all observed objects exhibit an MAE of 1.0 or lower, even with a noise factor of 0.4. The aggregate mean averages all MAEs for the evaluated objects, calculated to be 0.55 when analyzing test images with a noise factor of 0.25 and lower. The error is influenced by the complexity of the geometry and the symmetry of the object in question. More significant disturbances naturally lead to increased error. Notably, the prediction accuracy for objects 11 and 29 is adversely impacted by increased disturbances. This behavior can be attributed to their smaller sizes, which results in the loss of essential edge information, thus hindering accurate predictions. The predictions for these objects are consistent with the overall trends observed in other objects under minimal disturbance. In contrast, the more prominent objects (7, 8, 16, and 21), characterized by more intricate contours, exhibit outstanding performance in this evaluation. Despite the noise and increased overlap, these objects possess sufficient information to facilitate precise predictions.

Results on Objects

After evaluating the model's ability to predict 3D coordinates on synthetic test data, Edge2Pose is evaluated on real test scenes from the RT-Less [10] dataset. All pipeline components are implemented according to the specifications in Figure 3.1. As an initial assessment, each object is evaluated individually. The poses of the target objects are estimated for their corresponding scenes, and the mean value is subsequently calculated from these results. The metrics outlined in Chapter 2.1 are utilized to evaluate the pose estimation error. The findings are summarized in Table 4.3. This table presents the success rate for poses where the ensuing ADD-(S) error is below 10 percent of the object's diameter. Consistent with the methodology described in the RT-Less dataset paper [10], the mean rotation and translation errors are computed for the correct poses.

Obj ID	d/mm	ADD-(S)	R/°	t/mm
1	123.00	86.3	1.08	2.55
2	116.00	**98.2**	0.91	2.47
3	134.20	**98.7**	0.89	2.36
5	131.00	82.5	1.08	4.86
6	91.30	91.4	1.85	4.43
7	143.00	90.9	2.86	5.30
8	134.20	**97.8**	1.02	4.18
11	98.00	78.3	2.94	7.14
13	118.00	**96.7**	1.52	4.92
14	117.00	78.3	2.95	5.17
16	187.00	86.2	2.47	5.88
17	116.70	88.5	3.17	4.75
18	112.25	88.0	3.21	5.60
21	123.00	91.7	2.32	4.81
23	108.12	90.1	3.99	4.25
29	100.50	77.4	2.14	8.57
32	114.00	82.5	1.68	4.86
33	111.00	87.4	3.43	3.39
	AVG	91.4	2.36	4.75

Table 4.3: Results of pose estimation of RT-Less [10] models. Including the model diameter d dependent ADD-(S) error and the R/t error for valid poses.

Objects 2, 3, 8, and 13 have notably high values for the ADD-(S) metric. These objects enclose a clear and straightforward geometric structure, free from intricate symmetries, which enhances edge detection and enables accurate pose estimation. Contrarily, the results for Objects 11, 14, and 29 fall below the average, likely due to the effects of strong reflections or occlusions in the respective scenes, which may have adversely influenced the outcomes for these objects. The following section will examine the results of edge cases to explore this matter further.
The qualitative results of the predictions for an RT-Less [10] test scene are shown in Figure 4.8, which includes the predicted edges and coordinates and the resulting 2D-3D correspondences.

Figure 4.7: Predicted segmentation masks on RT-Less [10] scene

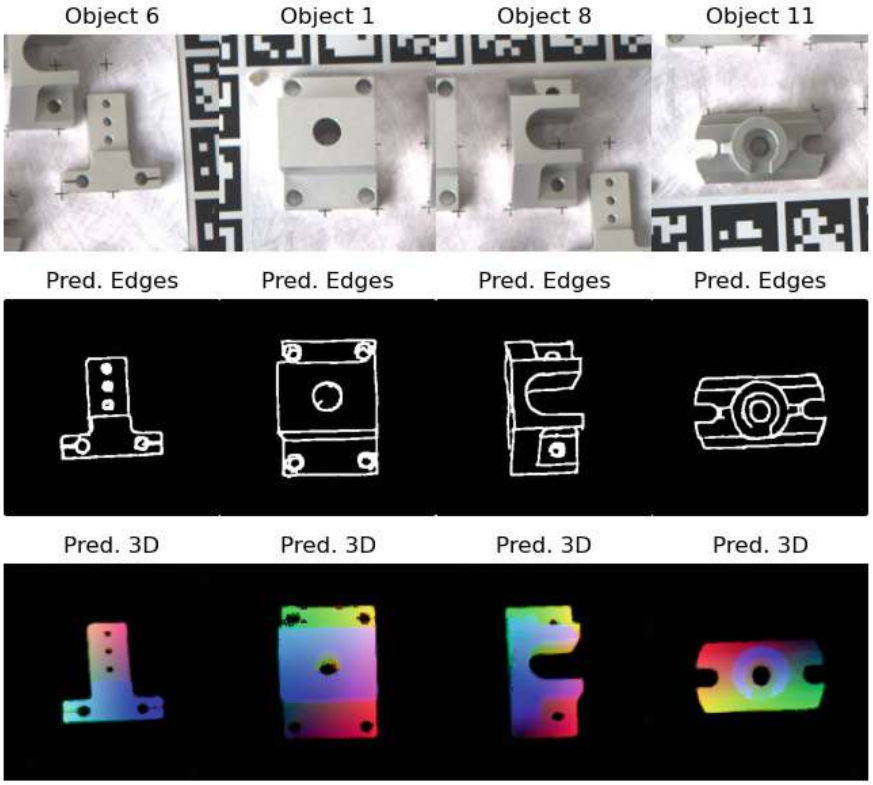

Figure 4.8: Coordinate prediction on RT-Less [10] scene

A comparison is made with recent state-of-the-art methodologies that have also been applied to the RT-Less dataset [10] to validate the method presented in this work. Notably, ContourPose [11] serves as a reliable benchmark, as it has produced the most successful results for this particular RT-Less dataset [10] dataset. This method employs an edge-based approach. The findings detailed in the related state-of-the-art studies and those of Edge2Pose are summarized in Table 4.4.

Obj ID	AAE [72]	STB [14]	PSGMN [73]	GFI [13]	CP [11]	Edge2Pose
1	76.96	64.21	94.23	95.32	**100.00**	86.30
2	76.43	66.49	70.86	96.77	97.54	**98.20**
3	84.32	54.65	82.45	92.16	95.35	**98.70**
6	32.42	48.90	74.95	**91.49**	88.14	91.30
7	64.77	36.48	79.57	87.85	90.70	**90.90**
13	45.32	62.36	84.34	85.03	96.71	**96.80**
16	49.33	29.45	74.11	76.31	**91.82**	86.20
18	72.12	45.49	75.89	84.22	**95.31**	88.00
21	67.09	62.26	79.94	89.92	**93.50**	90.10
32	71.32	59.23	87.30	85.11	**92.30**	82.50
AVG	64.91	52.85	80.36	88.42	**94.14**	90.90

Table 4.4: Comparison with different methods on RT-Less [10] dataset using ADD-(S)

In Table 4.5, the R/t error metric is assessed in detail for valid poses, analyzing the respective errors along their corresponding axes. In addition, these findings are compared with the methods employed in [11]. The two methods that yield the most favorable results are ContourPose [11] and GFI [13].

Method	x/mm	y/mm	z/mm	$\alpha/^\circ$	$\beta/^\circ$	$\gamma/^\circ$
AAE [72]	1.48	1.10	7.92	5.25	4.99	2.23
STB [14]	1.47	0.94	7.49	1.11	1.44	0.85
PSGMN [73]	2.50	1.97	8.45	4.00	3.74	1.52
GFI [13]	2.47	1.94	6.59	1.85	1.91	0.97
CP [11]	0.71	0.79	4.31	1.00	1.10	0.42
This	0.86	0.82	5.04	1.37	1.22	0.65

Table 4.5: Comparison using R/t metric on valid ADD-(S) poses

Results on Scenes

It is essential to examine the various scenes to gain a deeper understanding of the results from the previous experiments. As outlined in previous chapters, these scenarios differ in terms of illumination, reflections, and the level of occlusion. In addition, the backgrounds vary, affecting the accuracy of pose estimation. An overview of these characteristics is illustrated in Figure 3.5.
One of the primary challenges posed by Edge2Pose is handling metallic textures. Reflections on the object's surface may obscure the edges of its geometry, reducing the data available for accurately estimating the coordinates. The influence of these factors depends on the specific lighting conditions, the camera position, and the geometry or position of the object. Figure 4.9 offers a selection of edge cases designed to demonstrate the effects and outcomes clearly and unequivocally. In the initial example of object 1, a notable optical reflection occurs at the interface between two distinct surfaces, resulting in a near-seamless integration of the object's edge with the surrounding background. The edge detection algorithm demonstrates partial efficacy. While some edges are reliably predicted, others exhibit inaccuracies in classification. This example highlights the complexities inherent in boundary identification in varying optical contexts. A similar phenomenon is evident in the case of object 3, where the inner edge is nearly entirely obscured due to reflective interference. The contour can be utilized predominantly for pose estimation. The examples of objects 7 and 14 illustrate that reflections can create the illusion of edges on the surface. However, DiffusionEdge [38] does not consider these and generates only the actual contours. Multiple reflections across various surfaces can lead to the appearance of new edges, which are overlooked mainly during detection.

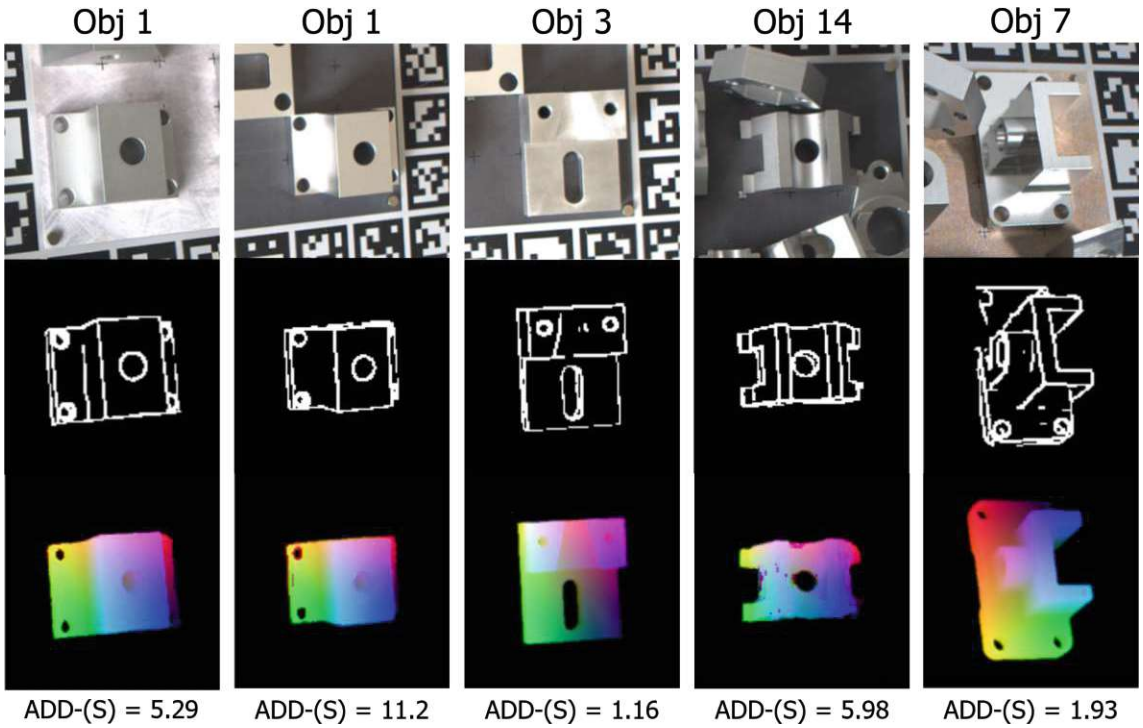

Figure 4.9: Qualitative evaluation of RT-Less objects [10] with high reflections includes the actual image, predicted edges, estimated 3D coordinates, and ADD-(S) error.

In addition, the test scenes feature images that are either underexposed or blurred. As illustrated in Figure 4.10, object 7 is affected by inadequate lighting and the presence of the light source itself. This results in indistinct edges and a textureless appearance on the surfaces. Furthermore, there is a noticeable reflection of the light source. The images of object 2 display a combination of a blurred camera shot and prominent shadows cast on the object. While the outlines of the objects are predominantly captured with minimal distortion, a significant increase in interference and noise along the edges distinguishes this observation from the previous one.

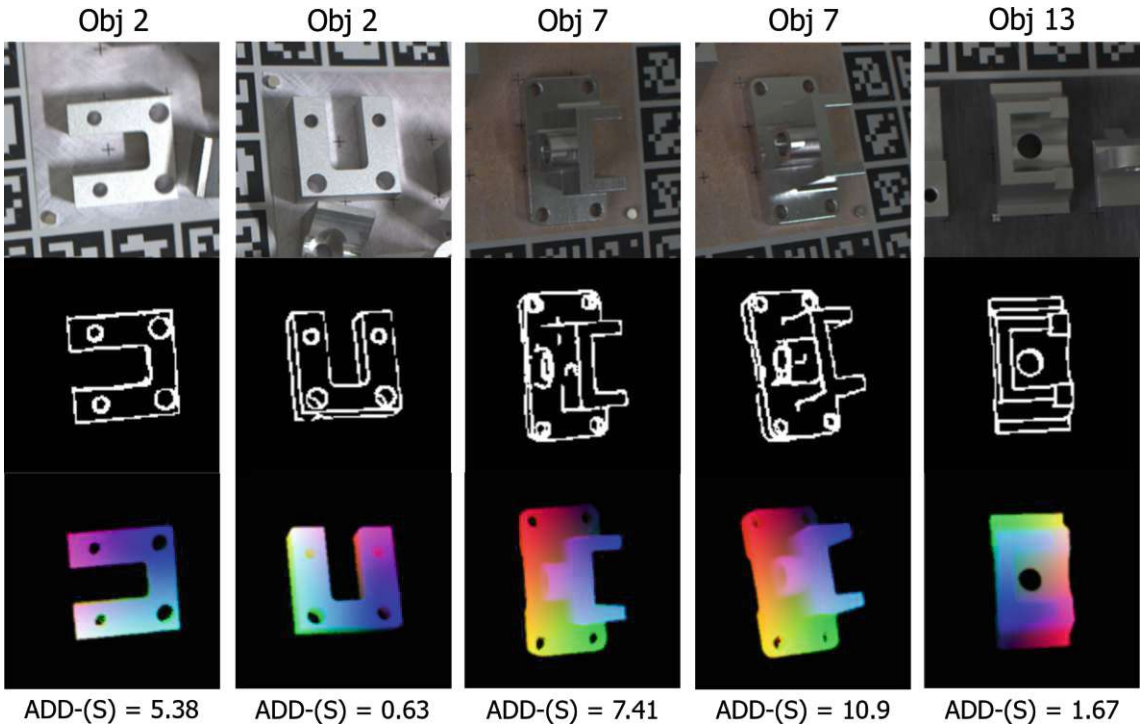

Figure 4.10: Qualitative evaluation of RT-Less objects [10] with illumination changes includes the actual image, predicted edges, estimated 3D coordinates, and ADD-(S) error.

The level of occlusion present within the scene is an essential factor influencing pose estimation. As shown in Figure 4.11, undetected edges are heightened along the objects' actual contours, mainly seen in the examples of object 23. In the image featuring objects 6 and 14, the edges of the overlapping object are mistakenly integrated into the contour of the actual object due to their inadequate recognition as interfering objects.

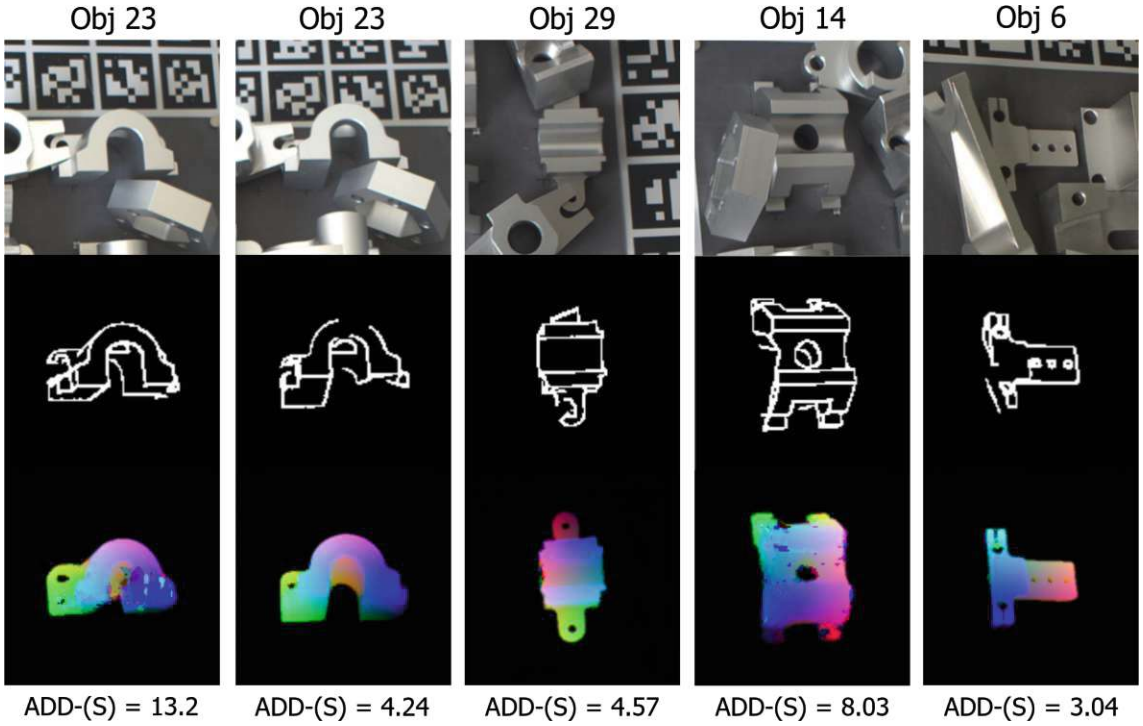

Figure 4.11: Qualitative evaluation of RT-Less objects [10] with occlusion includes the actual image, predicted edges, estimated 3D coordinates, and ADD-(S) error.

4.2 Further Datasets

The testing is extended to additional datasets to ensure a thorough validation of the methodology outlined in this thesis. Specifically, the T-Less dataset [32] from the BOP challenge [2, 3] is utilized for textureless but non-metallic objects and compared against the current state of the art. The proposed pipeline is applied to a different dataset featuring metallic and reflective objects, namely the MP-6D dataset [64].

4.2.1 T-Less Dataset

The use case has been expanded to include pose estimation for objects from the T-Less [32] dataset to provide a more comprehensive evaluation of Edge2Pose. This dataset is a focal point in the primary study, investigating the application of textureless objects in an industrial setting [2, 3]. As mentioned in Chapter 2.1, the dataset mainly consists of non-metallic items. It includes complex objects that lack distinctive textures, which can be observed in various scenarios within the industrial context.
The configuration of the individual components within the pose estimation pipeline remains unchanged. A new set of training data has been generated specifically for detecting and segmenting objects in the T-Less [32] dataset and further edge detection. The rendering pipeline was employed to create a dataset comprising 30,000 synthetic scenes. These scenes were designed with diverse randomized lighting conditions, background environments, and object layout configurations. This systematic variation facilitates comprehensive analysis and modeling of visual perception in complex environments. The main difference between these rendered scenes lies in the textures applied to the models. Instead of metallic textures, textures representing plastic surfaces are utilized.

Figure 4.12: Samples of synthetic training data with T-Less [32] models. From left to right, the RGB scene image, edge images, and segmentation masks. The individual masks are color-coded for improved clarity.

Evaluation of Edge Detection

The training of the YOLOv8 [37] model for object detection and segmentation is conducted using the aforementioned synthetic data. Horizontal and vertical flips and brightness and contrast adjustments are available to enhance training dataset diversity. The dataset is divided into three categories: training, testing, and evaluation, with a distribution ratio of 80:10:10. The training process is carried out over 50 epochs, utilizing the AdamW optimizer with an initial learning rate of $1e^{-4}$ and a batch size of 16.

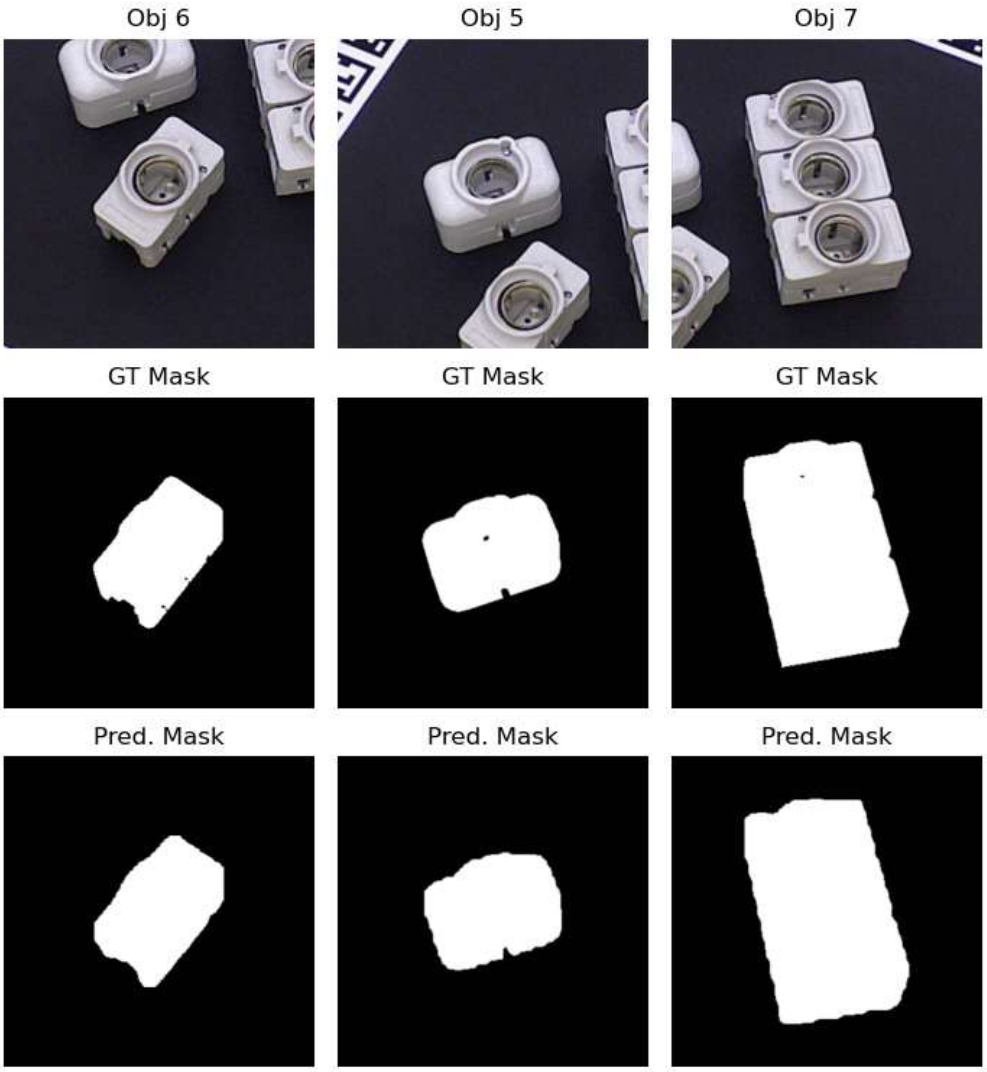

Figure 4.13: Results of YOLOv8 [37] on T-Less [32]

Edge2Pose is evaluated by applying the developed model to authentic testing scenarios derived from the T-Less [32] dataset. The Average Precision metrics from earlier work by [2, 3] are applied to evaluate the estimated poses. The results are detailed in Table 4.6, which differentiates between average precision at various threshold values. Specifically, the metrics AP50 and AP70 indicate the test cases in which at least 50% and 70% of the segmentation masks align, respectively. Moreover, the runtime in seconds required for each method to detect and segment the target object within the scene is also provided. Figure 4.13 presents an example of the ground truth and predicted segmentation masks from T-Less [32] test scenes.

Method	Domain	Det./Seg.	Synth	AP	AP_{50}	AP_{75}	$Time(s)$
CosyPose [26]	RGB	Mask R-CNN	✓	0.886	0.925	0.847	0.080
ZebraPose [74]	RGB	FCOS	✓	0.708	0.790	0.626	0.053
Edge2Pose	RGB	YOLOv8	✓	0.903	0.964	0.897	0.095

Table 4.6: Average Precision, AP50, AP70 and Runtime on T-Less [32] dataset.

The generated dataset, consisting of photorealistic scenes and corresponding ground truth edge images, is further utilized to train the edge detector. The DiffusionEdge [38] model is trained on this dataset with a training and test data split of 80:20. To enhance data augmentation, the scenes are flipped vertically and horizontally. The model is trained using the AdamW optimizer with a damped learning rate ranging from $5e^{-5}$ to $5e^{-6}$ throughout 40,000 iterations, utilizing a batch size of 8. In addition, Exponential Moving Average (EMA) is implemented to maintain stable model performance throughout the training process. During inference, image patches of size 240 x 240 are utilized, and overlapping regions are averaged to produce mean values. Exemplary results are illustrated in Figure 4.14.

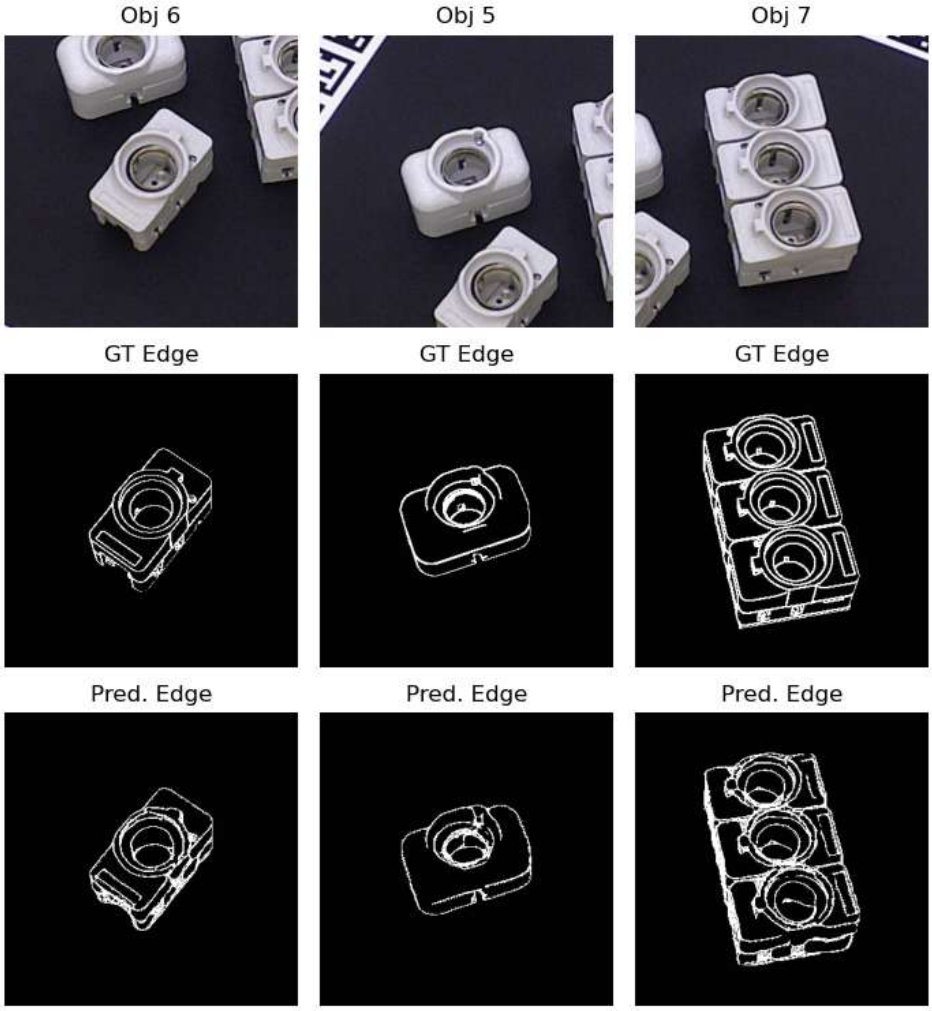

Figure 4.14: Results of DiffusionEdge [38] on T-Less [32]

Evaluation of Pose Estimation

The rendering process for the edge-coordinate training data remains consistent with that of the previous dataset. A total of 6,600 images, each with a size of 256 x 256 pixels, are captured for each object. The network is trained using the AdamW optimizer, with a learning rate set at $1e^{-4}$ and a batch size of 8. Training is conducted over 25 epochs. An additional augmentation process introduces random zoom-ins, cut-outs, noise along the edges, and variations in edge intensity to enhance the dataset further. The dataset is divided into training and test sets, maintaining an 80:20 ratio.
The Augmented Reality (AR) metric is utilized to evaluate estimated poses, enabling significant comparisons with other methods in the BOP challenges [2, 3]. The AR is determined by measuring the recall for each of the VSD, MSSD, and MSPD metrics across a range of thresholds and subsequently averaging these values. This metric indicates how effectively a model predicts poses and assesses performance across various tolerances. Table 4.7 presents the results of Edge2Pose and other approaches evaluated on the T-Less [32] dataset. These methods utilize an RGB-based approach and are categorized based on whether the respective networks were trained using synthetic or real training data.

Method	Domain	Synth	*AR*	*Time(s)*
GDRNPP [28]	RGB	✗	**78.7**	**0.23**
CosyPose [26]	RGB	✗	72.8	0.45
SurfEmb [75]	RGB	✗	73.5	8.89
ZebraPose [74]	RGB	✗	78.6	0.25
CDPNv2 [31]	RGB	✗	47.8	0.94
Pix2Pose [25]	RGB	✗	34.4	1.22
GDRNPP [28]	RGB	✓	**79.6**	**0.28**
CosyPose [26]	RGB	✓	64.0	0.48
SurfEmb [75]	RGB	✓	74.1	9.05
ZebraPose [74]	RGB	✓	72.3	0.50
CDPNv2 [31]	RGB	✓	40.7	0.98
EPOS [54]	RGB	✓	46.7	1.87
Edge2Pose	RGB	✓	74.4	8.40

Table 4.7: Average Recall and Runtime on T-Less [32]

In comparison to the methods outlined in Table 4.7, it is clear that the performance of Edge2Pose on the T-Less dataset significantly declines in this experiment. A more detailed examination of the individual scenes and objects reveals that the primary factor behind this noticeable difference is the quality of the predicted edges. The objects within the T-Less dataset [32] exhibit a greater complexity and variability in their contours, primarily due to a higher edge density when compared to the RT-Less dataset [10]. The density of edges in these objects leads to heightened noise and uncertainties along the predicted contours. This effect increases the chances of errors occurring in pose estimation. Despite these challenges, the AR value achieved in this experiment aligns well with established comparison methods, suggesting that Edge2Pose remains a competitive solution within the evaluated framework.

4.2.2 MP-6D Dataset

Another dataset featuring metallic and reflective objects is the MP-6D [64] dataset. The aim of utilizing this dataset is to expand the versatility of Edge2Pose by incorporating a wider range of conditions. Similarly, this dataset includes metallic objects from an industrial setting, much like the RT-Less [10] dataset. However, the scene images differ with respect to the camera's distance from the objects and the camera's orientation relative to the scene. As a result, the previously examined datasets, RT-Less [10], and T-Less [32], the grouped objects are not necessarily centered in the images. In addition, the increased distance between the camera and the scene leads to a significant portion of the image being taken up by distracting background elements. These factors introduce further challenges related to the rendering pipeline for generating training data and the methodology used for pose estimation.

Evaluation of Object Detection

The training data for the 'Edge Detection' section is generated within the rendering pipeline using the approved procedure. Models of various objects are placed randomly throughout the scene, and images are captured from different camera positions along the hemisphere. The radius of the sphere is calibrated to ensure accurate distances of the actual scenes. The rendering pipeline employs diverse environment textures to create realistic backgrounds. The segmentation model YOLOv8 [37] and the edge detector DiffusionEdge [38] are trained using this synthesized data. However, testing conducted with authentic images from the dataset has revealed that this training approach is ineffective. The DiffusionEdge [38] model struggles to identify the edges of objects accurately.

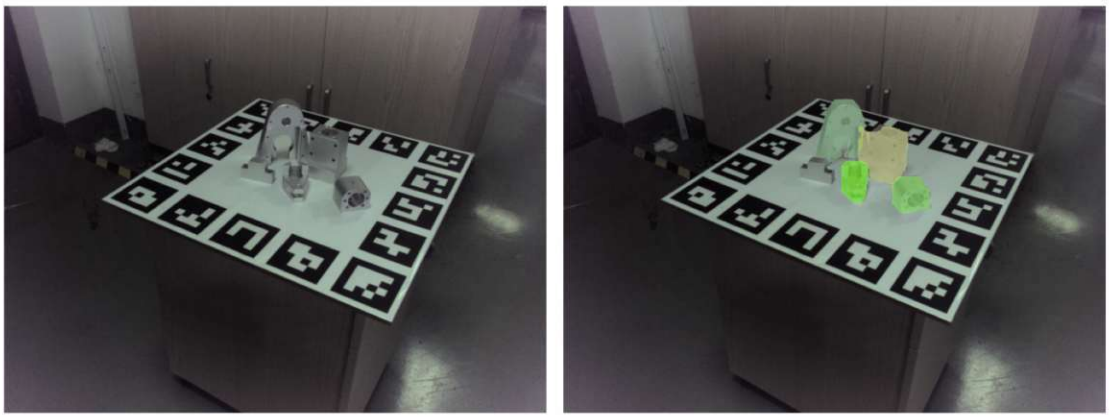

Figure 4.15: Object Detection on MP-6D [64] test scenes

The findings from the previous experiment necessitated modifications to the training data and the edge detection procedure. The inability of DiffusionEdge [38] to accurately detect the edges or contours of objects in the test scenes was linked to the insufficient similarity and diversity within the training data. The training data was aligned such that the camera consistently focused on the center of the scene. However, this does not reflect real test scenarios. Additionally, the size ratio between the objects and the background in the MP-6D [64] test scenes is significantly larger than in the prior datasets.
Consequently, the following adjustments were made: DiffusionEdge [38] is modified to predict edges based solely on a section centered on the target object rather than the entire scene image. This change aims to minimize the influence of the surrounding environment on

the detection process. The focus has systematically shifted to each model within a scene to enhance the effectiveness of the training data in this new approach. This targeted analysis allows for a more detailed representation of the individual objects. Sections were created from the original training dataset, each measuring 320 x 320 pixels. The workflow for edge detection was adapted. Predictions are now made in these object sections to enhance the details within the input data. Based on the detected bounding box, a 320 x 320 pixel section is extracted from the center of the scene image, which is then utilized for edge detection. During this experiment, the size of these cut-outs was adjusted to identify the optimal configuration. The outcomes of this revised method are illustrated in Figure 4.16. Only the 128 x 128 pixel size yielded recognizable edges and contours.

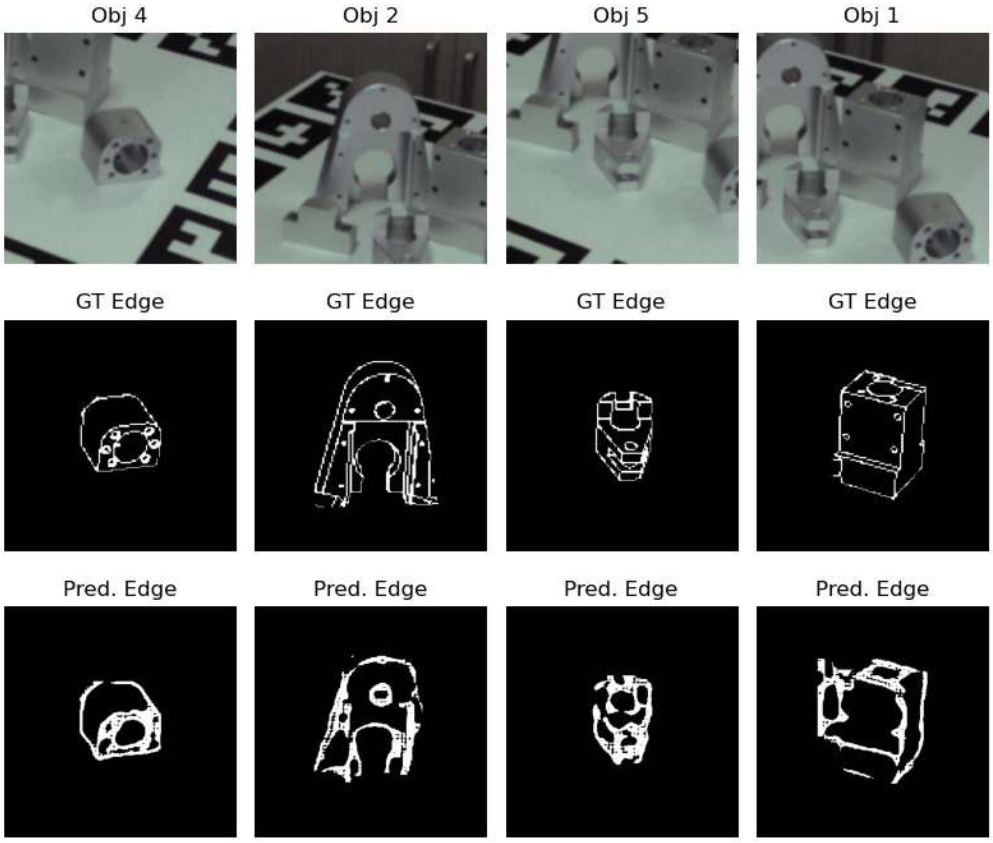

Figure 4.16: Predicted edges on various objects of MP-6D [64]

The findings show that DiffusionEdge [38] has difficulties accurately predicting edges in MP-6D [64] test scenes, even after attempts to reconstruct the workflow and adjust the training data. The outer contours of the objects are detectable, but crucial details are missing. This divergence makes them unsuitable for further pose estimation. One possible explanation for this result is that the objects are too small compared to the rest of the scene shown in the image. This imbalance means that the details of the edges are barely distinguishable from the background and from other objects that overlap. A weak contour can be recognized for individual objects. However, precise detection is no longer possible if several objects are in an image section.

CHAPTER 5

Conclusion

This chapter draws conclusions based on the outcomes of the methodology presented and its relevance to the initial research questions. It addresses how these questions could be answered by the content of this thesis. In conclusion, a proposal is put forth regarding the prospective enhancement of the work.

5.1 Metallic Object Pose Estimation

This thesis aims to develop a pose estimation methodology for metallic, reflective objects. A state-of-the-art review has revealed that object contours are a promising strategy for overcoming texture and high reflectance challenges. In light of this insight, a two-stage pipeline for pose estimation, called Edge2Pose, has been devised and implemented. This approach employs the prior estimation of the contours of the objects to facilitate the subsequent pose estimation process. The initial step is detecting and segmenting objects using the YOLOv8 [37] algorithm. Subsequently, edge detection is conducted using the DiffusionEdge [38] network. In the second stage of the pipeline, the edge images are used to perform pose estimation. A modified encoder-decoder network predicts the 3D coordinates using RGB color coding based on the edge images. The final pose estimation is achieved by establishing 2D-3D correspondences and their subsequent calculation using PnP/RANSAC [39, 40].
In addition, training data generation is a crucial aspect of implementing this methodology. A novel rendering pipeline has been introduced, which allows for the creation of photorealistic scenes alongside corresponding ground-truth edge images and segmentation masks. The data must be appropriately adapted to ensure the practical application of the methods to real-world scenarios, ultimately achieving the desired outcomes. The outcomes of the object recognition and segmentation tasks conducted using the YOLOv8 [37, 66] model and the edge detection results from the DiffusionEdge [38] network demonstrate that using rendered training data has yielded the expected results. The incorporation of synthetic images contributes substantial diversity, allowing the training data to effectively mirror the real-world conditions present in the test images from the datasets.
The issues associated with metallic objects, such as reflections or lack of textures, are addressed in this conceptual approach through the utilization of contours. It has been demonstrated that applying a diffusion model for edge detection is particularly well suited

to this objective. The experiments conducted with the RT-Less [10] dataset demonstrated that DiffusionEdge [38] is capable of robustly detecting precise object edges when being trained on with synthetic training data. Despite varying lighting conditions and reflections, it was proven that the generation of edge images can be accomplished with a high degree of reliability.

The final pose estimation is based on the edge images extracted from the objects. The network was developed using the concepts of CDPN [31], Pix2Pose [25], and DPOD [29] and has demonstrated reliability in experimental settings. The observed results indicate that the method can accurately estimate poses even in the presence of errors in the edge images due to strong reflections or occlusions. The results obtained are comparable to those of other methods, such as ContourPose [11] or SurfEmb [75], and thus provide a robust approach for pose estimation.

In conclusion, this thesis's scope sufficiently addressed all the initial research questions. It significantly contributes to research in the pose estimation of metallic objects. It conceptualizes and implements a contour-based approach, introducing a diffusion model and pose estimation from extracted edge images. The rendered pipeline for generating photorealistic training data allows the contour-based approach to be implemented. Edge2Pose effectively addresses the challenges inherent to the task, facilitating robust pose estimation.

5.2 Further Work

The present section considers the potential for enhancing the quality of the work. The initial observation is the runtime of the pose estimation pipeline. The average time required for detecting the target object in the scene and the subsequent pose estimation is 8 to 10 seconds. The primary factor contributing to this duration is the edge detection component. In its current version, DiffusionEdge [38] takes an average of 7 to 9 seconds to generate the edge image. Nevertheless, deploying a real-time model, which the DiffusionEdge [38] developers had announced at the time of this thesis but had not yet published, is expected to reduce the computation time. Such a model will likely considerably impact the entire pipeline's runtime, potentially reducing it.
A further aspect that warrants enhancement is how Edge2Pose deals with occlusions – a challenge common to most methodologies within this domain. A substantial number of objects in a given scene will increase interference in the edge images along the transition between these objects. As the accuracy of the estimated pose is contingent upon the quality of the detected contour, this can result in incorrect estimates. A potential solution to this issue is to apply edge detection not to the entire scene image but only to the observed sections along the scaled bounding box. This approach could also address the challenge of accurately predicting complex objects with closely spaced edges. By explicitly considering the target object, greater precision could be achieved. The feasibility of these approaches depends on the availability of an appropriate training dataset.
In summary, the pose estimation pipeline presents opportunities for enhancement. As is common in this field, further research is needed to understand the impact of occlusions on the process. The strong correlation between the accuracy of pose estimation and the quality of detected edges indicates potential for optimization. Implementing a real-time model for edge detection is expected to improve runtime efficiency further. In conclusion, Edge2Pose has successfully implemented an approach to pose estimation, demonstrating its effectiveness for metallic and reflective objects. By expanding the application of this method and integrating the proposed improvements, it is clear that this approach can offer substantial added value to the current state-of-the-art.

List of Figures

List of Tables

Bibliography

[1] S. Thalhammer, D. Bauer, P. Hönig, J.-B. Weibel, J. García-Rodríguez, and M. Vincze, "Challenges for monocular 6-d object pose estimation in robotics," *IEEE Transactions on Robotics*, vol. 40, pp. 4065–4084, 2024, Conference Name: IEEE Transactions on Robotics.

[2] T. Hodaň *et al.*, "BOP challenge 2020 on 6d object localization," in *Computer Vision – ECCV 2020 Workshops*, A. Bartoli and A. Fusiello, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 577–594.

[3] M. Sundermeyer *et al.*, "BOP challenge 2022 on detection, segmentation and pose estimation of specific rigid objects," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, ISSN: 2160-7516, Jun. 2023, pp. 2785–2794.

[4] Z. He, W. Feng, X. Zhao, and Y. Lv, "6d pose estimation of objects: Recent technologies and challenges," *Applied Sciences*, vol. 11, no. 1, p. 228, Jan. 2021, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[5] J. Guan, Y. Hao, Q. Wu, S. Li, and Y. Fang, "A survey of 6dof object pose estimation methods for different application scenarios," *Sensors*, vol. 24, no. 4, p. 1076, Jan. 2024, Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[6] A. Papadaki and M. Pateraki, "6d object localization in car-assembly industrial environment," *Journal of Imaging*, vol. 9, no. 3, p. 72, Mar. 2023, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[7] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6d object position estimation from 2d images: A literature review," *Multimedia Tools and Applications*, vol. 82, no. 16, pp. 24 605–24 643, Jul. 2023.

[8] A. E. Doruk, T. E. Ozkaya, F. Gülmez, and F. Uslu, "A comparative study for 6d pose estimation of textureless and symmetric objects used in automotive manufacturing industry," in *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2023, pp. 1–7.

[9] C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim, "A review on object pose recovery: From 3d bounding box detectors to full 6d pose estimators," *Image and Vision Computing*, vol. 96, p. 103 898, Apr. 1, 2020.

[10] X. Zhao, Q. Li, Y. Chao, Q. Wang, Z. He, and D. Liang, "RT-less: A multi-scene RGB dataset for 6d pose estimation of reflective texture-less objects," *The Visual Computer*, Oct. 18, 2023.

[11] Z. He, S. Zhang, and J. Tan, "ContourPose: Monocular 6-d pose estimation method for reflective textureless metal parts," *IEEE TRANSACTIONS ON ROBOTICS*, vol. 39, no. 5, 2023.

[12] Y. Liu and S. Feng, "6d pose estimation method using curvature-enhanced point-pair features," *IEEE Access*, vol. 11, pp. 122 598–122 609, 2023, Conference Name: IEEE Access.

[13] Z. He, M. Wu, X. Zhao, S. Zhang, and J. Tan, "A generative feature-to-image robotic vision framework for 6d pose measurement of metal parts," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3198–3209, Oct. 2022, Conference Name: IEEE/ASME Transactions on Mechatronics.

[14] Z. He, Z. Jiang, X. Zhao, S. Zhang, and C. Wu, "Sparse template-based 6-d pose estimation of metal parts using a monocular camera," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 390–401, Jan. 2020, Conference Name: IEEE Transactions on Industrial Electronics.

[15] C. Song, J. Song, and Q. Huang, "HybridPose: 6d object pose estimation under hybrid representations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 428–437.

[16] X. Yang, K. Li, J. Wang, and X. Fan, "ER-pose: Learning edge representation for 6d pose estimation of texture-less objects," *Neurocomputing*, vol. 515, pp. 13–25, Jan. 2023.

[17] C. Choi and H. I. Christensen, "3d textureless object detection and tracking: An edge-based approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 3877–3884.

[18] Y. Zhou, H. Qi, and Y. Ma, "End-to-end wireframe parsing," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 962–971.

[19] P. De Roovere, R. Daems, J. Croenen, T. Bourgana, J. de Hoog, and F. wyffels, "CenDerNet: Center and curvature representations for render-and-compare 6d pose estimation," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds., ser. Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2023, pp. 97–111.

[20] C. Chen, X. Jiang, S. Miao, W. Zhou, and Y. Liu, "Texture-less shiny objects grasping in a single RGB image using synthetic training data," *Applied Sciences*, vol. 12, no. 12, p. 6188, Jan. 2022, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[21] C. Chen, X. Jiang, W. Zhou, and Y.-H. Liu, *Pose estimation for texture-less shiny objects in a single RGB image using synthetic training data*, Sep. 23, 2019. arXiv: 1909.10270[cs].

[22] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6dof pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2019, pp. 4556–4565.

[23] Y. Zhu, L. Wan, W. Xu, and S. Wang, "ASPP-DF-PVNet: Atrous spatial pyramid pooling and distance-filtered PVNet for occlusion resistant 6d object pose estimation," *Signal Processing: Image Communication*, vol. 95, p. 116 268, Jul. 2021.

[24] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems XIV*, Robotics: Science and Systems Foundation, Jun. 26, 2018.

[25] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 7667–7676.

[26] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6d pose estimation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 574–591.

[27] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *2017 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2017, pp. 3848–3856.

[28] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2021, pp. 16 606–16 616.

[29] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6d pose object detector and refiner," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 1941–1950.

[30] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6d: Making RGB-based 3d detection and 6d pose estimation great again," in *2017 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2017, pp. 1530–1538.

[31] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 7677–7686.

[32] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-d dataset for 6d pose estimation of texture-less objects," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2017, pp. 880–888.

[33] "Blender python API." (), [Online]. Available: https://docs.blender.org/api/current/index.html (visited on 10/16/2024).

[34] M. Denninger *et al.*, *BlenderProc*, Oct. 25, 2019. arXiv: 1911.01911[cs].

[35] transcend-lzy, *Transcend-lzy/RT-less-toolbox*, original-date: 2022-03-22T01:44:44Z, Jun. 4, 2024.

[36] C. Li *et al.*, *YOLOv6: A single-stage object detection framework for industrial applications*, Sep. 7, 2022. arXiv: 2209.02976[cs].

[37] R. Varghese and S. M., "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, Apr. 2024, pp. 1–6.

[38] Y. Ye, K. Xu, Y. Huang, R. Yi, and Z. Cai, "DiffusionEdge: Diffusion probabilistic model for crisp edge detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, pp. 6675–6683, Mar. 24, 2024, Number: 7.

[39] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate o(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, Feb. 1, 2009.

[40] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1, 1981.

[41] R. Szeliski, *Computer Vision: Algorithms and Applications* (Texts in Computer Science). Cham: Springer International Publishing, 2022.

[42] M. Wolnitza, O. Kaya, T. Kulvicius, F. Wörgötter, and B. Dellen, "6d pose estimation and 3d object reconstruction from 2d shape for robotic grasping of objects," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, Dec. 2022, pp. 67–71.

[43] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Second edition. Cambridge: Cambridge University Press, 2003, 1 p.

[44] S. Thalhammer, "Simultaneous object detection and pose estimation under domain shift," Accepted: 2022-11-09T11:55:34Z Journal Abbreviation: Simultane Objekterkennung und Poseschätzung unter Domain Shift, Thesis, Technische Universität Wien, 2022.

[45] C. Du, J. Guo, S. Guo, and Q. Fu, "Study on 6d pose estimation system of occlusion targets for the spherical amphibious robot based on neural network," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, ISSN: 2152-744X, Aug. 2023, pp. 2058–2063.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2017, pp. 2980–2988.

[47] Z. Yang, X. Yu, and Y. Yang, "DSC-PoseNet: Learning 6dof object pose estimation via dual-scale consistency," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2021, pp. 3906–3915.

[48] T.-T. Do, M. Cai, T. Pham, and I. Reid, *Deep-6dpose: Recovering 6d object pose from a single RGB image*, Feb. 28, 2018. arXiv: 1802.10367[cs].

[49] W. Zou, D. Wu, S. Tian, C. Xiang, X. Li, and L. Zhang, "End-to-end 6dof pose estimation from monocular RGB images," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 1, pp. 87–96, Feb. 2021, Conference Name: IEEE Transactions on Consumer Electronics.

[50] Y. Hu, P. Fua, W. Wang, and M. Salzmann, *Single-stage 6d object pose estimation*, Mar. 20, 2020. arXiv: 1911.08324[cs].

[51] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6d pose estimation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 695–711.

[52] J. Liu, W. Sun, C. Liu, X. Zhang, S. Fan, and L. Zhang, "A novel 6d pose estimation method for indoor objects based on monocular regression depth," in *2021 China Automation Congress (CAC)*, ISSN: 2688-0938, Oct. 2021, pp. 4168–4172.

[53] C. Wang *et al.*, "DenseFusion: 6d object pose estimation by iterative dense fusion," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2019, pp. 3338–3347.

[54] T. Hodaň, D. Baráth, and J. Matas, "EPOS: Estimating 6d pose of objects with symmetries," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2020, pp. 11 700–11 709.

[55] J. Mei, X. Jiang, and H. Ding, "Spatial feature mapping for 6dof object pose estimation," *Pattern Recognition*, vol. 131, p. 108 835, Nov. 2022.

[56] V. N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, and V. Lepetit, "Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 6771–6780.

[57] V. Druskinis, J. M. Araya-Martinez, J. Lambrecht, S. Bøgh, and R. P. de Figueiredo, "A hybrid approach for accurate 6d pose estimation of textureless objects from monocular images," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, ISSN: 1946-0759, Sep. 2023, pp. 1–8.

[58] F. Tombari, A. Franchi, and L. Di, "BOLD features to detect texture-less objects," in *2013 IEEE International Conference on Computer Vision*, ISSN: 2380-7504, Dec. 2013, pp. 1265–1272.

[59] R. G. v. Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, Mar. 24, 2012.

[60] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[61] J. Hu, H. Ling, P. Parashar, A. Naik, and H. Christensen, *Pose estimation of specular and symmetrical objects*, version: 1, Oct. 31, 2020. arXiv: 2011.00372[cs].

[62] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 548–562.

[63] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger, "Introducing MVTec ITODD — a dataset for 3d object recognition in industry," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, Italy: IEEE, Oct. 2017, pp. 2200–2208.

[64] L. Chen, H. Yang, C. Wu, and S. Wu, "MP6d: An RGB-d dataset for metal parts' 6d pose estimation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5912–5919, Jul. 2022, Conference Name: IEEE Robotics and Automation Letters.

[65] P. De Roovere, S. Moonen, N. Michiels, and F. Wyffels, "Sim-to-real dataset of industrial metal objects," *Machines*, vol. 12, no. 2, p. 99, Feb. 2024, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[66] M. Zhao *et al.*, "MED-YOLOv8s: A new real-time road crack, pothole, and patch detection model," *Journal of Real-Time Image Processing*, vol. 21, no. 2, p. 26, Jan. 29, 2024.

[67] "Object detection in autonomous maritime vehicles: Comparison between YOLO v8 and EfficientDet | SpringerLink." (), [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-6755-1_10 (visited on 12/15/2024).

[68] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, May 24, 2019, pp. 6105–6114.

[69] K. Wang and Z. Liu, "BA-YOLO for object detection in satellite remote sensing images," *Applied Sciences*, vol. 13, no. 24, p. 13 122, Jan. 2023, Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

[70] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2019, pp. 2637–2646.

[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jun. 2016, pp. 770–778.

[72] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 714–729, Mar. 1, 2020.

[73] C. Wu, L. Chen, Z. He, and J. Jiang, "Pseudo-siamese graph matching network for textureless objects' 6-d pose estimation," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2718–2727, Mar. 2022, Conference Name: IEEE Transactions on Industrial Electronics.

[74] Y. Su *et al.*, "ZebraPose: Coarse to fine surface encoding for 6dof object pose estimation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2022, pp. 6728–6738.

[75] R. L. Haugaard and A. G. Buch, "SurfEmb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2022, pp. 6739–6748.