Masters Thesis Report
Forklift and Human Pose Estimation for Safety and Risk Analysis in Industrial Environments

Aniruddha Pal*
Matrikel Nr.: 9029755

B-IT Master Studies in Autonomous Systems

Bonn-Rhein-Sieg University of Applied Sciences

Sentics GmbH

Advisors:
Prof. Dr. Sebastian Houben**
Prof. Dr. Paul G. Ploeger††
MSc. Ayan Roy Chowdhury‡‡

April 15, 2025

*aniruddha.pal@smail.inf.h-brs.de
**sebastian.houben@h-brs.de
††paul.ploeger@h-brs.de
‡‡a.chowdhury@sentics.de

Confidentiality Notice

To the examiner:

This Master Thesis project report contains confidential information from Sentics GmbH. Please note that the author of this Master Thesis project report has fully agreed to comprehensive confidentiality obligations towards Sentics GmbH, which must be strictly observed.

Declaration

I, **Aniruddha Pal**, hereby declare that this work has not previously been submitted to this or any other university and that unless otherwise stated, the content is entirely my own work.

15/4/2025

Date

Aniruddha Pal

Acknowledgement

I would like to acknowledge the extended support of my advisors, Prof.Dr. Sebastian Houben, Prof.Dr. Paul G. Ploeger and MSc. Ayan Roy Chowdhury, towards the completion of this work. I am grateful to them for their timely inputs and ever-welcoming discussions through our progress meetings.

I would like to acknowledge my colleagues at Sentics GmbH for helping me throughout my masters thesis project. I would especially like to mention Sai Bhai and Aw Thura foe their constant support and valuable insights during ths work.

I would like to thank all the staffs of Hochschule-Bonn-Rhein-Sieg for providing me with the necessary resources. Last but not the list, I would like to thank my mother, Aparna Pal, my brother, Anabadya Pal, and my girlfriend, Priyanka Das, for their constant motivation during this time.

Abstract

Ensuring safety in environments where humans and heavy machinery, such as forklifts, share space is critically important. This research investigates pose recognition models for human and forklift detection, focusing on enhancing dataset quality and optimizing detection algorithms aimed at boosting safety compliance within industrial settings and minimizing workplace incidents. We perform a comparative analysis of YOLO based pose estimation frameworks, assessing their ability to deliver real-time monitoring and risk assessment. A significant portion of the project is allocated to developing techniques for generating high-quality synthetic data for scenarios where real-life data collection is challenging. Experimental evaluations in real images of industrial environments are carried out to confirm the system's efficiency, accuracy, and effectiveness in detecting safety risks. While heavier models provided better accuracy in keypoint estimation, such as the Nano version, were 30-43% faster in detection of Human and Forklift poses. mAP@[.50:.95] scores of 0.68-0.8 were acheived for Humans and more than 0.98 for Fprklift across models. This work also investigated the advantage of incorporatig synthetic data into the dataset. The models were able to predict more instances and robustness in pose estimation on the test set.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

mAP	mean Average Precision
GPU	Graphics Processing Unit
DCNN	Deep Convolutional Neural Network
YOLO	You Only Look Once
FoV	Field of View
ANN	Artifical Neural Network
AI	Artificial Intelligence
IoT	Internet of Things
ML	Machine Learning
AP	Average Precision
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
DL	Deep Learning
IoU	Intersection over Union
PR	Precision Recall
CNN	Convolutional Neural Network
R-CNN	Region based Convolutional Neural Network
COCO	Common Objects in Context
PCP	Percentage of Correct Parts
PDJ	Percentage of Detected Joints
PCK	Percentage of Correct Key-points
OKS	Object Keypoint Similarity
RTL	Real Time Localization
RTLS	Real Time Localization System

1 Introduction

Industrial Environments are dynamic and involve humans and modern machinery in close proximity. It is important to value safety and risk management to ensure a productive workplace. Forklifts have proved to be an excellent machine for handling and transporting materials in production environments. However, interaction between humans and forklifts poses a significant occupational hazard, especially to pedestrian workers. Crashes involving forklift trucks and unattentive workers can result in severe injuries, even fatal in some cases [39]. It is imperative to design and integrate robust monitoring framework to mitigate such serious risks.

Pose Estimation of Humans and Forklifts can be used to track their movements and identify scenarios where potential accidents can take place. Based on this, preventive measures can be designed. For example, alarms with various coloured LEDs can be implemented to signal hazardous scenarios and the level of risk. Based on the detection of positions of forklifts and humans, automatic braking can be implemented in moving Forklifts in cases where they are close to a pedestrian worker beyond a threshold value.

In industrial settings, accurately estimating the positions and movements of both humans and vehicles is highly valuable. It enhances safety and boosts productivity. Worker movements can be monitored through pose estimation, which supports ergonomic evaluations and highlights potential safety risks or inefficiencies in industrial processes. Similarly, vehicle pose estimation facilitates real-time tracking and navigation of autonomous systems like automated guided vehicles (AGVs), improving logistics and transportation efficiency. Precise monitoring and analysis of human and vehicle poses allow organizations to optimize operations, reduce risks, and foster a safer and more productive industrial environment.

Advancements in artificial intelligence (AI)-driven pose estimation have been propelled by breakthroughs in computer vision, especially deep learning. Convolutional Neural Networks (CNNs) and recurrent models have been instrumental in achieving high accuracy in detecting and tracking human and vehicle poses [68]. Complex spatial patterns are learned by leveraging large-scale datasets in combination with advanced algorithmic frameworks, enabling reliable pose estimation even in dynamic and challenging industrial environments. By leveraging AI-driven pose estimation, industries can achieve greater efficiency, safety, and automation.

This research work investigates recent developments and implementations of AI-powered pose estimation in industrial environments, focusing on human and vehicle tracking. It examines the challenges of pose estimation in dynamic and cluttered settings, discusses cutting-edge methods and frameworks, and evaluates the benefits and limitations of implementing these systems in industrial contexts.

Although there is a significant amount of publicly accessible data for human pose estimation, data for forklift pose estimation is very limited. Besides, among the human data available, there is a lack of data specific to human pose estimation in warehouse environments. Warehouses generally have quite a cluttered environment, and this results in various occlusions. It is a big challenge for pose estimation models to accurately estimate pose in these scenarios. In this study, we aim to address these two challenges with the help of the generation of synthetic data for forklifts and humans. The development of this data-generation pipeline is another major contribution of this study.

2 Problem Formulation

In this age of cutthroat market competition and constant innovations in production process, it has become increasingly important to ensure improvement of logistical and Supply chain management [65]. The link between various operations in the production process is termed as Intralogistics [66]. There are increased requirements regarding the flexibility and reliability of intralogistics, most importantly, safety for both human workers and machinery involved.

Workers in industrial settings may face various hazardous situations while working with machinery in close proximity. Unsafe conditions in the industry often arise due to the following factors:

- Employee negligence and irresponsibility
- Ignoring safety measures
- Inadequate training and awareness for employees
- Shortage of staff
- Poor maintenance of equipment
- Lack of order and cleanliness
- Ineffective management of occupational health and safety within the company

In this context, there is a high risk of accident associated with the use of forklifts during work. Collisions of Forklifts and humans can cause serious injuries and can sometimes even be fatal. From retrospective analysis of forklift accidents from 2004 to 2019, it is observed that, complex lower limb injuries are caused by these incidents, often leading to complications requiring multistage treatment. Recovery is prolonged, with many returning to work after an extended period, though lasting impairments frequently remain [78]. Accordingly, the development of more robust safety measures is warranted to ensure smooth operations, protect employees, and prevent accidents. However, despite efforts to prevent and minimize workplace accidents, data from the German Social Accident Insurance indicate that achieving this goal remains challenging. The 2020 occupational accident statistics reveal that a significant number of workplace accidents in Germany occurred within intralogistics, particularly in forklift operations, transport, and storage facilities. In total, over 31,000 occupational accidents were recorded in these areas, including 10 fatal incidents involving material handling trolleys and forklifts. [18]. It represents a comparison of occupational accidents in the first 6 months of 2020 and 2021, that the number of occupational accidents, including fatal accidents, will increase significantly, namely by more than 7 percent [18].

Many safety systems have already been put into operation to protect people and other goods in intralogistics and to reduce the risks [1] [28] [17] [35]. Optical object recognition is a crucial component. However, optical detection technologies are susceptible to errors, as the accuracy of position determination is influenced by factors such as light intensity, ambient brightness, image distortion from camera lenses, object occlusion within the field of view, and the perspective relative to the object's direction of motion [15] [16] [17]. Accurately estimating the positions of humans and forklifts is crucial for identifying potential hazards, maintaining safe distances, and preventing accidents. However, achieving this level of precision in dynamic and complex environments is highly challenging.

The availability of labeled data to train pose estimation models for specific cases in industrial environments is a tricky issue, as it is not available too much. Collection of data is also not always

possible, especially for specific edge cases, for example accidents and some specific poses that are not easy to recreate in real life. To solve this issue an extensive synthetic data generation approach has been designed. Apart from real data collected from industrial environments, synthetic data is used to train human and forklift pose estimation models.

The challenge of human pose estimation in production settings involves precisely identifying a body configuration of human [69], including joint positions and orientations. This data is essential for ergonomic evaluations, safety monitoring, and customized support. However, obstacles like occlusions, inconsistent lighting, and fast movements complicate accurate pose estimation. To enhance worker safety and streamline workflow efficiency, current techniques must overcome these difficulties and deliver reliable, precise solutions.

Forklifts are crucial vehicles in material handling and transport within production settings. Precisely estimating their pose is essential for tracking movements, preventing collisions, and ensuring safe distances from people and other objects. However, achieving real-time pose estimation in dynamic environments is challenging due to factors like cluttered backgrounds, occlusions, and limited sensor data. To enhance accuracy and reliability, a robust solution is required to ensure the safe and efficient operation of forklifts and reduce the risk of accidents in industrial environments.

Tackling the challenges of human and forklift pose estimation is vital for improving safety and streamlining production processes. Developing precise and reliable pose estimation algorithms tailored to production environments is key to protecting workers, reducing accidents, and enhancing operational efficiency.

2.1 Research Question

1. Identify which pose estimation model is best suited for security surveillance scenario?
2. What is the impact of Synthetic data in pose estimation performance?

3 Current State of Research

Industrial environments like warehouses are hazardous and dynamic environments where machines like forklifts and human workers function in the vicinity. As reported by the Occupational Safety and Health Administration (OSHA), accidents caused due to unsupervised forklift and human interactions, contribute to nearly 34,900 significant injuries and about 85 fatalities annually in the United States alone [25]. Safety measures used traditionally, such as, physical barriers, training programs, are often not sufficient enough for accident prevention in crowded and fast-paced industrial settings. Lack of attention to occupational health and safety (OHS), leads to lasting expenses, such as those related to workplace accidents, delays, and reworking, also damaging the reputation of the involved organization [80]. Thus, it is of primary importance to ensure workplace safety. This chapter presents the scientific basis for the development of person and forklift recognition and their localization in the industrial environment.

The integration of advanced technologies, data-driven approaches, proactive safety measures, and a strong safety culture characterizes the state of the art in logistics safety. By adopting these advancements, logistics companies can improve safety performance, prevent accidents, protect human workers, and enhance overall operational efficiency. In this section, existing systems for safety based on the real-time localization of vehicles and humans in production and intralogistics are discussed. The pros and cons of the respective systems are examined. Artificial intelligence (AI) is being increasingly integrated into existing systems. Pose estimation in computer vision requires identification and linking key semantic points on the human body, such as the ”right eye”, ”left hip”, ”right ankle”, and ”left elbow” [77]. Based on the spatial reference system, Human pose estimation (HPE) can generally be divided into two types: 3D and 2D posture estimation [10]. 3D pose estimation necessitates identification and analysis of X, Y, and Z coordinates of the joints from images or videos. The task is to detect and evaluate the X and Y coordinates of human body joints using an RGB image. Alternatively, in 2D posture estimation, the goal is to map the X and Y co-ordinates of the human body or an object like a vehicle using the same kind of image. A Human Pose Skeleton represents the pose of a human graphically, likewise, a vehicle pose skeleton represents the crucial points of the vehicle. Skeletons consist of a set of connected points that represent the stance of the object. Each edges of the skeleton is termed as joint, component, or keypoint. When two such points are connected, they comprise a limb or functional pair. It is important note that not all components are valid.

3.1 A Summary of Contemporary Safety Systems in Intralogistics

Industrial environments and warehouses pose significant challenges to forklift operators due to it's dynamic and congested nature, which can cause general safety risks. To mitigate this challenge, industries incorporate various logistic safety measures using advanced technologies and approaches that are developed with the objective of enhancing safety in logistics operations. Some key areas that highlight the forefront of logistics safety are discussed below:

1. **Advanced Automation**: Technologies in automation, including robotics, and automated guided vehicles (AGVs) are transforming logistics safety [**?**]. These technologies minimize human involvement in potentially dangerous tasks, reduce human errors, and enhance operational safety [88].

2. **Sensor Technologies**: Sensors are crucial in logistics safety, providing real-time data to monitor and prevent accidents. Technologies like proximity sensors, vision systems, LiDAR, and RFID (Radio Frequency Identification) extensively untilized to detect obstacles, track vehicle movements, and maintain safe distances [53].

3. **Predictive Analytics**: Machine inference and data evaluation contribute to proactive logistics management and prevention of potential safety risks. Predictive models interpret past record, extract patterns, and offer actionable insights for proactive safety measures.
4. **Real-time Monitoring and Telematics**: Such systems enable the continuous tracking of logistics operations. GPS tracking, fleet management systems, and video surveillance systems provide real-time visibility and allow for timely interventions in the event of safety concerns [36].
5. **Safety Training and Education**: Prioritizing safety training and education is vital for logistics workers. Effective training programs, simulation-based training, and virtual reality modules equip employees with the knowledge and skills necessary to handle safety risks.
6. **Safety Culture and Management**: Promoting safety as a core value and cultivating a strong safety culture is essential for logistics companies. Effective safety management systems, protocols, risk assessments, and safety audits ensure a comprehensive approach to safety throughout the organization.
7. **Collaboration and Standards**: Industry collaboration and adherence to safety standards are key to enhancing logistics safety. Organizations and regulatory bodies work together to develop and implement safety protocols, standards, and recommended procedures.
8. **Internet of Things (IoT) and Connectivity**: IoT devices and connectivity solutions facilitate real-time data exchange and facilitate communication between various components within the logistics system. This enhances situational awareness, supports predictive maintenance, and improves safety coordination.
9. **Human-Machine Collaboration**: As collaborative robots (cobots) and human-robot interaction increase, ensuring safe and efficient collaboration between humans and machines is critical to logistics safety. Standards and guidelines for safe human-machine interaction are being developed to prevent accidents and injuries.

The first type of security systems used in logistics involves ultra-wideband (UWB) technology. UWB refers to wireless technology used for communication and designed for personal area networks (PANs), characterized by low poIt is well-suited for use cases that require high-quality service and can be applied in areas like wireless personal area networks (WPAN), home networks, and short-range radar. Unlike continuous sine wave systems, UWB uses narrow pulsed signals with low transmission power (0.5mW) in the nanosecond to picosecond range, providing a bandwidth of at least 500 MHz [16] [42].

Because of the very short pulses, UWB devices transmit at high speeds using only a fraction of the power required by conventional continuous wave systems. UWB has a transmission range of at least 10 meters and supports supports communication speed of at least 480 Mbps, which is 240 times faster than Bluetooth and 20 times faster than Wi-Fi. With a one-second delay for 100 queries, UWB enables near-continuous positioning. Its resistance to interference and high data rates make it ideal for ultra-precise positioning in industrial environments, achieving accuracy within 10-30 cm [16]. In contrast to modulated signal systems, UWB systems do not rely on carrier waves, making them simpler and more cost-effective [42]. Object positioning is determined using the time difference of arrival (TDoA) time of flight (ToF) or methods, which, when combined with the velocity of UWB signals, enable precise distance calculations [42]. UWB devices consist of receivers that capture data from UWB transmitters and calculate propagation time or phase differences. UWB tags, attached to moving objects like people or vehicles, send

signals to UWB receivers [42]. Furthermore, UWB tags can connect to mobile devices via Bluetooth or USB to display the position data of the tracked objects.

Based on UWB technologies, there are many applications for person recognition and collision prevention of industrial vehicles in production. For example, the companies INSOFT and ELOKON have developed warning systems with UWB and launched them on the market

A UWB tag is emitted from each person and vehicle so that they can be positioned and detected from each other in the industrial environment. If the distance between two objects falls below the predefined adjustable safety distance, there can be various reaction mechanisms, e.g. sending of optical and acoustic warning signals or automatic vehicle braking can be used. The safety distance is marked as yellow and red. The speed of the vehicles is adjusted according to this color. UWB enables detection even if people and vehicles cannot see each other directly due to obstacles. Furthermore, the positions are documented, where often warnings or accidents happen.

In addition to systems with ultra-wideband (UWB), Radio Frequency Identification (RFID) is commonly used for object detection [19]. Although the operating principles of RFID and UWB are alike, their main differences lie in the wavelength of the signals and the detection range. RFID systems, along with their corresponding transponders (RFID tags), communicate using high-frequency electromagnetic waves over short distances. RFID technology is used to locate and trace individuals and mobile platforms. However, both low and high frequency RFID systems have a significant limitation in terms of detection range, as both the receivers and RFID tags are typically only detectable within distances of less than 1 meter. This limited range makes them impractical for situations where avoiding vehicle collisions is critical. To address this, the range can be extended by increasing frequencies and using active transponders.

ZoneSafe, utilizing RFID technology, has developed a proximity warning system to protect workers from vehicle accidents. An antenna on the vehicle detects RFID tags worn by pedestrians or workers from a distance of up to 10 meters. When the antenna identifies a tag, the vehicle's control unit activates an alarm, alerting the driver that a person is nearby. In addition, the RFID tag worn by the employee vibrates, notifying them of the approaching vehicle. This system helps reduce the risk of dangerous interactions between vehicles and pedestrians [92].

Another approach to improving safety is through optical warning systems. For example, Linde has developed warning systems such as BlueSpot and TruckSpot [47] [48]. These systems aim to prevent accidents by making employees aware of the proximity of nearby vehicles through a visual signal. Both TruckSpot and BlueSpot project a light signal or warning sign on the ground via an LED light mounted on the vehicle's overhead guard frame. This alerts employees and nearby vehicles to an approaching vehicle. However, optical warning systems alone are not sufficient to prevent accidents, as employees can be inattentive, distracted, or focused on their work, causing them to overlook visual signals and potential danger.

Another approach involves the integration of a camera-based system into logistics. Cameras capture images of the environment and objects within the industrial setting, with the resolution determined by the individual pixels that store the information. After identifying the 2D positions of the objects within the image, their corresponding contact points in 3D space are determined through camera calibration [89]. In the event of a potential collision, warning signals or other responses can be triggered. Additionally, recorded images can be used for post-accident analysis in case of a collision. The use of cameras in safety systems offers significant advantages, including cost-effectiveness, robustness, and reliability. However, their ability to detect objects is significantly affected by environmental factors. For example, variations in color properties, such as brightness and color saturation during the day or evening, as well as occlusion or the

obstruction of objects by obstacles, can result in detection errors [3] [38].

As an example of how camera detection theory is applied and commercialized, VIA Technologies offers products and services such as the VIA Mobile360 D700 AI Dash Cam [79]. This product is primarily used in traffic and transportation applications. Equipped with two 1080p interior and front cameras, the VIA Mobile360 D700 AI Dash Cam captures high-definition video of both the driver and the surrounding conditions simultaneously. Micro SD cards can be used to store the recordings locally, and may be stored in the cloud via 4G or Wi-Fi. Custom alerts can be configured to trigger real-time collision warnings.

Additionally, by utilizing the CAN bus, it is possible to collect extensive vehicle metrics like speed, distance traveled, idle duration, and fuel usage. This data can then be uploaded to the cloud for analysis, aiding in uncovering ways to enhance both fleet performance and safety. By tagging G-sensor events such as hard acceleration, hard braking, and sharp cornering, the device offers fleet operators valuable insights to improve driver behavior and reduce vehicle wear and tear [5].

3.2 Visual safety in intralogistics: Enhancing protection through optical sensors.

Arcure Group has developed *Blaxtair*, an on-board camera system designed for industrial and logistics vehicles to enhance workplace safety. Blaxtair is an intelligent vision-based system capable of differentiating humans from other obstacles in real-time and alerting the operator in case of potential hazards [4].

The system is composed of three main components:

- A sensor head with a stereo camera,
- A computing unit,
- An alarm generator, which includes an LCD screen or a warning LED light.

The stereo camera captures environmental data, which is then processed by a recognition algorithm. Should a threat be detected, the system sends a visual and acoustic alert within 200 to 300 milliseconds. Operators can also monitor the situation live via the control screen [86]. Despite its effectiveness, Blaxtair cannot fully eliminate industrial accidents. Individuals obscured by opaque objects or located beyond the camera's viewing range may remain undetected. Moreover, the system's accuracy can degrade if the cameras are damaged or contaminated.

3.3 Real-Time Localization Systems (RTLS)

Real-Time Localization Systems (RTLS) enable the tracking and positioning of individuals and assets in real-time using various integrated technologies. Unlike GPS, which is designed primarily for outdoor use and suffers from signal attenuation indoors, RTLS is specifically tailored for indoor environments. These systems handle object detection and localization, where minimizing latency in data acquisition, processing, and transmission is critical for responsive control [74]. RTLS technologies are widely employed in production and logistics. Products from manufacturers such as Elokon, Safezone, Linde, and VIA are either fully or partially based on RTLS infrastructure. Additionally, technologies like ultrasound and infrared can complement RTLS deployments.

The startup *Sentics* offers an RTLS-based solution for collision prevention and navigation in autonomous transport vehicles. The system uses optical sensors to detect people and vehicles,

and processes positional data to track them in real-time. This data not only enhances safety analytics but also supports the creation of a digital twin of the workplace, enabling remote supervision of safe workflows [50].

3.3.1 Components of an RTLS System

To achieve efficient and real-time performance, an RTLS system requires the seamless integration of both hardware and software components:

Hardware Components

1. **Optical sensors:** Responsible for capturing image data used by computer vision algorithms.
2. **Central processing server:** A server equipped with powerful GPUs processes frames from all connected cameras simultaneously. It runs AI models to detect and track objects, then relays positional data to the network.

Software Components

1. **AI Models:** Object and keypoint detection algorithms that determine the spatial location of tracked entities.
2. **Accelerated computing platforms:** High-performance hardware (e.g., NVIDIA platforms) that enable real-time image processing and inference.
3. **Data fusion algorithms:** These algorithms combine data from multiple cameras, eliminating redundancy and generating a unified view of object locations.

3.4 Optical object detection and computer vision in industrial environments

In this work, artificial intelligence (AI) is employed to detect people and vehicles in industrial environments using optical sensors and digital images. In most cases, an image contains multiple objects of interest, each of which must be correctly classified and localized. In the field of computer vision, this process is referred to as *object detection*. With the advancement of powerful hardware, such as GPUs and TPUs, and continuous improvements in detection algorithms, object detection has become increasingly important in both computer vision (CV) and safety-related systems [24].

The underlying principle of computer-based image recognition closely mirrors human visual perception. In humans, image recognition is based on categorizing visual input based on distinct features. The brain compares the current image to stored memory patterns, identifying similarities to previously seen features in order to recognize the object.

Likewise, machine-based image recognition relies on the extraction and classification of relevant features while discarding redundant data. These features can range from highly distinctive to more general characteristics, directly impacting the recognition performance of the system. In general, image content is described through *image features*. Every image possesses unique features. Studies on human eye movements reveal that our eyes tend to focus on areas with the most visual information where contours curve sharply or abruptly change direction. The scanning path of the eye continually shifts from one feature to another. This behavior suggests that the perceptual system prioritizes relevant details and discards unnecessary information, integrating them into a complete perceptual image.

As image recognition techniques for machines are modeled after human visual perception, the stages of the recognition process are quite similar. Typically, the image recognition pipeline includes the following steps:

1. **Information acquisition**: Capturing raw image data from sensors.
2. **Preprocessing**: Enhancing the image resolution and structuring data for analysis.
3. **Feature extraction and selection**: Identifying and isolating meaningful image features.
4. **Classifier development**: Training models to distinguish between object classes.
5. **Classification**: Assigning detected objects to predefined categories based on extracted features.

This structured approach enables robust and efficient recognition, essential for real-time safety systems in industrial applications [7].

3.5 Object Recognition Image Processing Pipeline

Information acquisition refers to the the conversion of environmental stimuli, including light and sound, into electrical signals via sensor devices [27]. In the context of digital object recognition, this means collecting fundamental data about the target object and converting it into machine-readable information. Optical sensors play a key role by capturing light and converting it into digital images composed of individual pixels. Each pixel is represented by three numerical values ranging from 0 to 255, corresponding to the red, green, and blue (RGB) color channels. The resulting pixel color is determined by the combination of these three values. For instance, a pixel with RGB values of (0, 0, 0) is black, whereas (255, 255, 255) yields white [45].

A standard high-resolution image, for example, may have a resolution of 1920x1080 pixels. Each pixel is assigned a unique coordinate in a two-dimensional space, denoted by (x, y). In computer vision, the origin (0, 0) is located at the top-left corner of the image. The x-axis increases horizontally to the right, reaching a maximum of 1920, and the y-axis increases vertically downward, with a maximum value of 1080 [67].

Preprocessing involves applying image enhancement operations such as denoising, smoothing, and format conversion to improve the visibility of important features and prepare the data for further analysis [71].

Feature extraction and selection are essential steps in the image recognition pipeline. Feature extraction refers to the process of identifying relevant characteristics that can distinguish one type of object from another. Since not all extracted features may contribute positively to recognition performance, feature selection is employed to extract the most relevant attributes. This step is vital for improving accuracy and efficiency in pattern recognition tasks [43].

Classifier development involves training models or recognition rules that can categorize objects based on their extracted features. The **classification decision** stage assigns the object to a specific class within the feature space, enabling the system to reliably detect and interpret the object under observation [49].

There are multiple methods for object recognition and localization in images. Among these, the use of *bounding boxes* and *keypoints* for pose estimation is particularly effective. For example, when detecting humans, their position can be determined using defined keypoints such as joints, providing both spatial and structural context for further analysis.

3.5.1 Bounding-box detection of objects in industrial environments

Object detection using bounding boxes is essential not only for identifying whether an object is present in an image but also for accurately locating its position. This process involves drawing bounding boxes around multiple objects within the image to distinguish and classify each target region [3].

The task of object detection and localization comes with several challenges, as outlined below [37]:

1. **Variable Number of Objects:** The number of objects present in an image may vary, making it difficult to represent this information using fixed-size vectors, which are often required by machine learning models. Since the exact number of objects is unknown beforehand, additional post-processing steps are necessary, increasing the overall complexity.
2. **Varying Image and Object Sizes:** Even identical objects can appear at different scales in different images. This variation in object size makes detection more difficult. While some algorithms attempt to address this using sliding window techniques, they are often inefficient [60].
3. **Model Definition and Metrics:** Object detection involves both classification and localization. Most models use multi-task loss functions that penalize both incorrect classifications and inaccurate bounding boxes. However, the dual nature of such loss functions can lead to suboptimal performance in both tasks.
4. **Limited Data:** A significant limitation in object detection is the scarcity of annotated datasets. The lack of sufficient labeled training data poses a major challenge to developing robust models.
5. **Detection Speed:** For real-time applications such as video analysis, detection algorithms must be not only accurate but also extremely fast. Videos typically run at 24 frames per second, making it difficult to develop models that can maintain high precision while operating at such speeds [63].

In this section, the term *bounding box* is introduced and explained. Subsequently, some object detection algorithms that utilize bounding boxes will be discussed, along with their advantages and disadvantages. In object recognition, a bounding box is typically used to mark the objects and describe their positions. The bounding box is a rectangular frame that completely contains the object in the image. It is represented by four values: the x and y coordinates of the upper left corner, and the x and y coordinates of the lower right corner. When an object is identified by a bounding box, its point of contact can be derived from it. However, determining the contact point depends on various factors, such as the camera angle, image distortion, and object coverage. Typically, the contact point is simplified to the center of the lower horizontal frame line of the bounding box.

In tasks involving object recognition, annotations within the dataset serve as the reference positions for detected entities. These are termed as *ground truth boxes* (e.g., (3, 3, 5, 5)). Conversely, the coordinates predicted by the neural network model define a *prediction box*. The model estimates the probable location of a particular entity, alongside the predicted label L and associated confidence score P, which indicates the likelihood that the detected object corresponds to the specified class. One input frame can yield multiple predictions. A major challenge in such detection frameworks, whether involving anchor regions or proposed segments is the repeated identification of the same entity through several bounding rectangles [87]. The objective then becomes selecting the best-fitting box with the greatest confidence.

There are two principal strategies for object recognition using convolutional networks, each with distinct strengths and limitations, depending on the number of stages involved [21]. These techniques are broadly categorized into single-stage and two-stage models. In the two-stage approach, candidate regions are generated first, which are then refined through classification and localization steps using CNNs. Though precise, this method is computationally expensive. By contrast, single-stage methods bypass proposal generation and directly infer object classes and locations in one step using a unified CNN. This significantly reduces computational load but often compromises on precision [90].

Regional Convolutional Neural Networks (R-CNN) was introduced by Girshick et al. [32]. This algorithm provides a solid basis for subsequent developments, and belongs to the group of two-stage detectors. In order to mitigate the challenge of selecting a large number of regions, they have developed a method in which a selective search is used to extract only 2000 regions from the image that are likely to contain objects. Then, the features in each region are extracted by CNN, and a Support Vector Machine (SVM) is used to classify the objects in the image. For each class, an SVM must be trained. In the end the object in the image is localized by a final bounding box regressor (a linear regression model), which assists in obtaining a more reliable bounding box for each detected object [32]. Given the CNNs for the feature extraction, the linear SVM classifier for target object detection, and the regression model for bounding box, a particularly obvious drawback of R-CNN is that it runs slowly and computationally expensive [32].

To decrease computational overhead, Girshick et al. introduced the Fast R-CNN framework [31], offering substantial improvements over the earlier R-CNN design. This approach revises how features are extracted, object categories are predicted, and bounding boxes are generated, resulting in a significant drop in inference time. Specifically, Fast R-CNN processes a single image in approximately 2.2 seconds [30]. In Faster R-CNN, the Region Proposal Network (RPN) generates candidate regions using a more efficient method than the selective search used in previous models [31]. Because it produces fewer candidate areas, the overall detection pipeline becomes more suitable for near real-time usage. Even so, Fast R-CNN still faces limitations when handling large-scale image datasets, primarily due to its reliance on selective search for identifying regions of interest. Thus, powerful GPUs are often necessary for adequate performance.

These techniques are part of the two-stage detection category. Though more accurate, they tend to be slower than their single-stage counterparts. For instance, You Only Look Once (YOLO) [64] was developed to address these speed challenges, enabling real-time object detection without dependency on high-end GPUs. YOLO splits the input image into an $s \times s$ grid and assigns bounding boxes to each grid cell. For every bounding region, it computes both classification probabilities and positional adjustments. Only bounding boxes with confidence scores exceeding a preset threshold are retained, which then serve to localize objects within the image. This method processes roughly 45 frames per second, making it significantly faster than traditional detectors. However, YOLO tends to underperform when identifying tiny elements, such as birds grouped together. This shortfall stems from the spatial constraints of the grid-based structure. Nonetheless, YOLO remains an essential advancement in fast object detection and is widely adopted in real-time systems.

3.5.2 Utilization of Bounding Box-Based Object Localization in the Existing Real-Time Localization Framework

For a localization framework to function effectively, identifying target items is essential. This task is accomplished using YOLO [64], one of the most advanced algorithms for object recognition. As of the year 2023, YOLOv8 represents the most recent and refined iteration among its predecessors. Over time, YOLO has become a pivotal component in fast object recognition and

is widely adopted in robotics, driverless systems, and automated video monitoring. YOLOv8, introduced by Ultralytics, the organization also behind YOLOv5, was officially released in January 2023 [70]. At that time, no academic publication had detailed the architecture of YOLOv8, necessitating a comparison-based understanding with earlier variants.

This version operates without predefined anchors, resulting in fewer bounding box hypotheses and accelerating the non-maximum suppression (NMS) stage. YOLOv8 also utilizes mosaic augmentation during model training. However, to avoid degradation in learning, this technique is deactivated for the final ten training epochs. Users may initiate YOLOv8 via a command-line interface (CLI) or install it through a PIP package. It supports a wide array of tools for labeling, model training, and deployment workflows.

Five distinct versions of YOLOv8 exist, namely nano (YOLOv8n), small (YOLOv8s), medium (YOLOv8m), large (YOLOv8l), and extra-large (YOLOv8x) [**?**]. On the MS COCO dataset (test-dev 2017), YOLOv8x reached an average precision (AP) score of 53.9% with input dimensions of 640 pixels. In comparison, YOLOv5 recorded 50.7% under the same conditions. YOLOv8x also achieved processing speeds of up to 280 frames per second (FPS) using NVIDIA's A100 GPU with TensorRT acceleration.

3.5.3 Bounding-Box Detections at Sentics GmbH RTLS

The Sentics GmbH system currently employs bounding box detection as the primary method for real-time object localization. The AI model at Sentics is based on YOLOv8s, a fast and accurate single-stage detector. This model has been trained on diverse instances of humans and various types of forklifts, making it well-suited for industrial environments where forklifts are commonly encountered.

The training dataset consists of 90,000 images, which are divided into a 70% training set and a 30% validation set. This split ensures the model can generalize effectively, striking a balance between fitting the training data and validating its performance on unseen samples. During the training process, the YOLOv8s model detects and localizes objects by creating bounding boxes around the target items, facilitating real-time object detection and localization across different environments.

By leveraging YOLOv8s, Sentics GmbH achieves high speed and accuracy in detection, which are critical for real-time applications, especially in industrial settings. This enables the system to detect and track forklifts and other objects, contributing to safer and more efficient operations in dynamic environments.

3.5.4 Limitations of Bounding Box-Based Detection in Real-Time Localization Systems

In the current industrial use case, an object detection algorithm is employed to detect objects in the image plane and subsequently localize them in the global plane. Several steps are involved, starting from acquiring image data, passing through multiple processing stages, and ultimately transforming the object's calculated position from the image plane to the global plane.

While bounding box detections are fast and work well, there are limitations, especially given the complexity of industrial environments. These environments contain a mix of static and dynamic objects, which can obscure the view of other objects of interest. Since the bounding box is determined by enclosing the object, an incomplete bounding box, particularly around the bottom of the object, caused by partial occlusion of the object, can lead to an inaccurate estimated foot-point. When this projected foot-point is transferred to the global plane, it becomes shifted

by certain factors. To address this issue, this work investigates an alternative algorithm. For example, consider a situation where a human in a production environment is obscured by other objects, and only the upper part of the body is visible to the camera. The object detection algorithm detects the visible features and produces a bounding box with a high confidence level. This bounding box will enclose the visible portions of the human's body, with the foot-point in the image plane being positioned at the center of the torso. When this foot-point is projected onto the global plane, the resulting point is shifted, depending on the height of the bounding box's bottom side relative to the ground. Although the actual foot-point lies elsewhere, the bounding box detection will incorrectly place it.

To overcome this issue, a feature-based computer vision task known as pose estimation can be used. Pose estimation allows machines to identify and track body parts in images and videos, such as locating the position of a human knee. In cases where a body part is not visible, pose estimation can infer its position based on other visible keypoints. Unlike object detection, which identifies objects and provides coarse localization via bounding boxes, pose estimation models focus on estimating the precise locations of key points associated with an object, providing finer granularity for localization.

3.6 Estimation of Pose

Pose estimation is defined as a deep neural network based regression problem which helps in recognizing body joints [77]. These body joints are labeled by keypoints and each keypoint may have some relation to other keypoints which are annotated by connections. In the use case of this thesis, pose has to be estimated for two different types of objects, namely humans/persons and forklifts.

Human pose estimation (HPE) has garnered notable attention in the field of computer vision. With the advent of deep learning, significant advancements have been made in solving this challenge, mirroring the developments observed in other visual recognition tasks. Since the early work introduced in [76] and [77], performance on the MPII benchmark [**?**] has significantly improved within three years, rising from approximately 80% [77] to over 90% with newer architectures [11, 13, 56, 84]. Progress on the more recent and demanding COCO dataset [46] has been even more accelerated.

Moreover, the intricacy of architectural configurations and trial frameworks has grown, which complicates both assessment and benchmarking processes. With the rapid growth of intelligent IoT applications, there is a heightened need for compact and efficient multi-person pose estimation systems [54]. Traditionally, the detection of humans has played a central role in general object detection tasks. Thanks to contemporary machine learning methods, it is now possible for computers to interpret body movements by analyzing poses and tracking motion. The current precision levels and technical infrastructure now support deployment in real-world systems.

Furthermore, in the recent past COVID-19 pandemic has accelerated the demand for real-time posture recognition technologies, which could drive innovation in visual intelligence systems. For instance, combining pose estimation with spatial projection techniques allowed the implementation of social distancing tools, helping people maintain physical separation in crowded settings. In autonomous driving, this technology has already shown considerable value. Using live pose detection, systems can track and anticipate pedestrian actions more effectively, thereby enhancing the safety and reliability of navigation systems.

In general, human pose estimation enables the creation of models that represent body posture, such as skeletal frameworks, by precisely identifying joint locations through visual data. This

modeling process plays a crucial role in extracting meaningful features from input imagery. A model-based strategy is often adopted to infer 2D or 3D poses, supporting representation in either dimensional space. Pose estimation methods are typically divided into three fundamental model categories for visualizing the human figure in 2D or 3D formats [91].

- **Skeleton-based model:** Often termed a kinematic representation, this model is applicable to both 2D and 3D pose prediction. It captures the structure of the human form using joint coordinates and limb orientations. Its strength lies in the clear, interpretable way it models the interconnections among body parts. However, it falls short in conveying details like shape or surface texture.
- **Contour-based model:** Also known as a planar approach, this technique is generally used for estimating 2D poses. It focuses on representing body appearance and shape using simplified outlines, typically formed by rectangular blocks corresponding to body segments. A notable example is the Active Shape Model (ASM), which utilizes principal component analysis (PCA) to model silhouette changes and the body's overall geometry [15].
- **Volume-based model:** This approach is geared toward 3D pose inference. Volumetric methods rely on established 3D body templates, which are integrated into deep learning systems to predict joint locations and reconstruct the body mesh. One widely adopted model, SMPL (Skinned Multi-Person Linear), allows for realistic deformations that mimic muscle and soft-tissue motion.

Most modern methods rely on a rigid kinematic configuration involving N joints, representing the body as a connected network of limbs and joints. This model incorporates structural and shape-related details of the human figure.

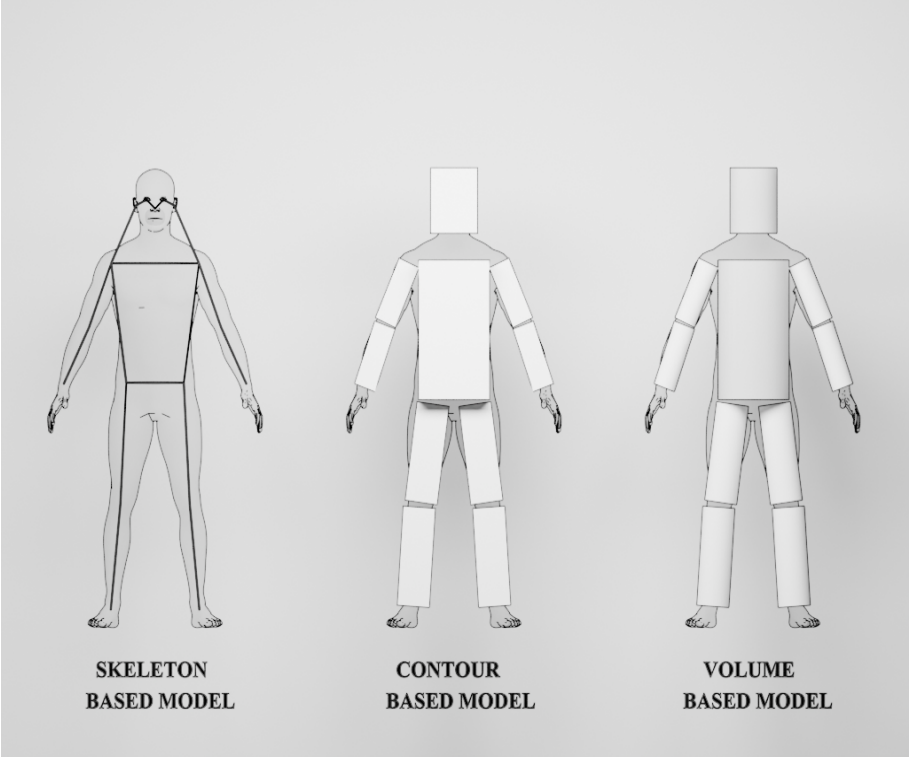


Figure 1. Visual Representation of Human Body Models a. Skeleton Based Model b. Contour Based Model and c. Volume Based Model Adapted from [12]

Human pose estimation remains a challenging task due to the constantly changing visual charac-

teristics of the body. Factors such as clothing variations, occlusions, changes in perspective, and complex background environments make the process particularly difficult. For a pose estimation system to function effectively in real-world conditions, it must be resilient to external variables like lighting conditions and weather changes. These factors complicate the model's ability to accurately locate joint positions, especially when it comes to detecting smaller or less prominent joints.

Just like estimating human posture, pose estimation for non-human objects involves identifying meaningful keypoints based on the object's shape or structural features. In the context of this research, the focus is on determining the pose of a forklift. This requires consistent labeling of keypoints across a variety of forklift models and image instances. While forklifts typically have fewer keypoints compared to the human body, the task introduces its own complexity. The visual similarity and repetitive patterns across different parts of a forklift increase the difficulty for keypoint detection algorithms, which may require larger datasets to learn robust representations and generalize effectively.

This thesis will center on the training and evaluation of models designed for both human and forklift pose estimation tasks.

3.6.1 Keypoint Detection based Pose Estimation

Keypoint detection involves not only identifying objects within an image but also precisely localizing their significant points. These keypoints, also known as interest points, are distinct locations within an image that highlight visually important features. They remain stable under various transformations such as rotation, scaling, translation, or geometric distortions. In the context of anatomical keypoint detection, most prior work has focused on identifying specific body components [6,22,62,82]. Typically, the input to a pose estimation model is a preprocessed image captured by a camera, while the output consists of a set of keypoints. Each detected keypoint is associated with a body part ID and a confidence value, usually ranging between 0.0 and 1.0, which indicates the certainty of the model about the presence of a point at that specific location.

1. **2D Pose Estimation:**
   This task refers to determining the two-dimensional coordinates of an object's keypoints using visual inputs such as images or video frames. Traditionally, 2D pose recognition relied heavily on manual feature engineering targeting distinct body parts. Earlier computer vision methods often represented human posture using simplified stick figure models. However, the introduction of deep learning has notably improved the accuracy and robustness of 2D pose estimation for both single-person and multi-person scenarios.

2. **3D Pose Estimation:**
   This process focuses on predicting the three-dimensional locations of joints or keypoints in space. It leverages visual data from single-view images or videos to reconstruct the spatial structure of the body. Applications include 3D motion capture, action recognition, virtual environments, and AR/VR systems. 3D pose estimation may integrate information from multiple viewpoints or additional sensors such as IMUs and LiDARs. Fusion techniques are often employed to enhance accuracy. Despite its potential, the task remains highly challenging due to issues such as the high cost and difficulty of creating annotated datasets, the computational demands of the models, challenges in generalization across domains, and sensitivity to occlusions.

Estimating the pose of multiple objects within images, especially in industrial settings, introduces a unique range of challenges. Firstly, varying illumination conditions result in inconsis-

tent visual input, which can hinder the accuracy of keypoint predictions. Secondly, interactions among multiple objects create spatial complexity due to occlusions, physical contact, and articulated structures, making it difficult to correctly associate individual parts [8]. Lastly, with the increase in the number of detected objects, the computational demands rise, posing difficulties for achieving real-time processing and inference speeds.

3.7 Motivation for System Enhancement

Industrial environments such as warehouses and logistics hubs often present unpredictable and highly dynamic scenarios, making it difficult for vehicle mounted sensors to reliably detect hazardous or life threatening situations. A typical example would be a forklift approaching a sharp blind corner, where neither the driver nor the onboard sensors can detect an individual walking on the other side of an opaque obstacle potentially leading to a serious accident.

A viable solution to this limitation involves installing multiple cameras within the indoor industrial space, offering a near bird's-eye perspective of the surroundings. Reconsidering the earlier example, one camera could monitor the path of the forklift on one side of the wall, while another captures the pedestrian's movement on the opposite side. The visual data from these cameras can be transmitted to a centralized processing system, which uses AI-based algorithms to identify both the person and the forklift, and convert their locations into global coordinates. With appropriate post-processing, the movement trajectories of the two entities can be analyzed, enabling the system to issue a timely alert via audio, visual signals, or both, to the forklift operator about the presence of a nearby human.

Multi-target multi-camera tracking (MTMCT) plays a crucial role in several infrastructure and transportation-related applications, such as optimizing traffic light intervals or analyzing vehicle flow [72]. When integrated with reliable AI models, such a multi-camera framework can address the limitations encountered by real-time object localization systems in industrial settings.

This work utilizes 2D pose estimation, a vision-based approach, to enhance workplace safety by enabling real-time visual tracking and analysis of both forklifts and personnel. This is achieved by identifying and following keypoint coordinates corresponding to human joints (17 in total) and forklift structural components (8 keypoints) within 2D image space. The developed system, grounded in pose estimation, is capable of recognizing potentially unsafe behavior and issuing real-time warnings to prevent collisions or accidents. The next sections delve into specific 2D pose estimation strategies for both humans and forklifts, with a focus on their role in industrial safety systems.

3.7.1 2D Human Pose Estimation

Vision-based 2D human pose estimation (HPE) involves the identification and tracking of human keypoints in 2D images or video. In industrial settings, 2D HPE proves useful for detecting unsafe behaviors, monitoring worker movements, and predicting potential accidents. The ability to perform real-time pose estimation makes 2D HPE a valuable tool in improving workplace safety.

Human pose estimation methods can be classified into three categories: bottom-up, top-down, and hybrid, based on the methodology used by the models. The Bottom-Up approach, first introduced by DeepCut [40], involves detecting keypoints first, followed by the identification of the bounding box, indicating the object's position within the image. A well-known system for bottom-up HPE is OpenPose [9], which is especially useful for multi-person real-time pose estimation. OpenPose uses Part Affinity Fields (PAFs), which encode the relationships between

body parts, helping to associate detected joints with individual poses. This approach is robust in multi-person scenarios, such as crowded warehouses, and can handle occlusions effectively.

In contrast, the Top-Down approach first identifies the individuals and their bounding boxes, after which the pose is estimated through a separate network. HRNet [73] is a popular top-down model known for capturing fine-grained human pose details. It excels at detecting subtle unsafe behaviors but is computationally expensive, making it less suitable for real-time multi-person pose estimation. As real-time performance is crucial for this application, HRNet is not ideal.

While the bottom-up approach is efficient for multi-object detection, it struggles with scale variations of humans in images, leading to inaccuracies in keypoint prediction. The top-down approach handles scale variation more effectively by normalizing all human instances to the same scale, but it sacrifices speed. A hybrid model, often used in applications like warehouses where accuracy and speed are both essential, combines the strengths of both approaches. The YOLO-pose network utilizes a hybrid model, directly optimizing the Object Keypoint Similarity (OKS) metric, allowing the joint detection of keypoints and bounding boxes. The major advantage of this method is that no grouping of keypoints is needed, thus accelerating the pose estimation process. 2D HPE systems can provide timely warnings to operators and overseers, supporting proactive safety measures.

3.7.2 2D Forklift Pose Estimation

Forklifts are vital in warehouse and industrial environments, performing various movements such as lifting, positioning, turning, and tilting. Since forklifts are often operated near human workers, their interactions with people or objects can lead to serious accidents. Monitoring forklift movements is therefore essential for safety. However, forklifts can be partially occluded by machinery or other objects, making pose estimation challenging. Here, pose estimation becomes a crucial tool for more accurately tracking forklift positions. Effective pose estimation can trigger real-time alerts to prevent accidents.

2D forklift pose estimation involves detecting and tracking the positions of forklift components, such as forks, masts, and wheels, in 2D images or video frames. This data can be used to analyze forklift movements, detect unsafe operations, and predict potential collisions. Forklift pose estimation plays a key role in ensuring safe forklift operation in crowded warehouse environments. Keypoints on forklifts, such as the tips of the forks and the base of the mast, are detected to estimate their pose. These keypoint detection tasks are often enhanced with object detection models like YOLO or Faster R-CNN [44]. Custom neural networks are trained on annotated datasets of forklift images to improve pose estimation accuracy. YOLO framework is suitable for detecting forklifts in video streams, offering both speed and accuracy.

Workers may be partially occluded by machinery, making pose estimation more difficult. Multi-view pose estimation techniques are needed to address this issue. Workers perform a wide range of movements, such as walking, bending, and lifting. Robust pose estimation models are necessary to handle such diverse activities. Furthermore, real-time pose estimation systems must operate efficiently to provide timely alerts. This requires fast algorithms and hardware acceleration.

Forklift pose estimation can be used to predict collisions with humans or obstacles. For instance, a system could alert the forklift operator if they are approaching a worker or obstacle. It can also detect unsafe forklift operations, such as excessive speed, improper load handling, or operating in restricted areas. Real-time monitoring can help prevent accidents and ensure compliance with safety protocols. Additionally, forklift pose estimation plays a critical role in

the autonomous navigation of forklifts, enabling them to navigate safely and avoid obstacles in the warehouse.

3.7.3 Synthetic Data Generation

Synthetic data generation plays a crucial role in modern machine learning by facilitating the creation of extensive, varied, and accurately labeled datasets for training models. Synthetic datasets generated through Blender often include realistic warehouse settings, with annotated human and forklift poses. Blender's adaptability and comprehensive feature set make it a highly effective tool for generating synthetic data.

4 Data acquisition and data annotation

Just as humans learn through observation, AI models require datasets (ground truth) to learn from, enabling them to apply the knowledge gained to new data. AI models are essentially software programs designed to perform specific tasks that involve decision-making using a dataset. In simple terms, these models are intended to emulate the cognitive process and informed judgment demonstrated by human experts. Similar to humans, AI algorithms rely on datasets (ground truth) for learning, allowing them to generalize the knowledge acquired to new data.

The method of data collection plays an important role in designing an effective machine learning (ML) model. The AI model's decision-making ability is directly influenced by both the quantity and quality of the data collected. The performance and reliability of the model are significantly impacted by the data. As a result, the process of organizing and gathering data often consumes more time than the actual training of the model.

Data collection is followed by the process of image annotation, which involves manually labeling the ground truth information. In simple terms, image annotation refers to visually marking the type and location of objects that need to be recognized by the AI model. For example, to train a deep learning model to detect dogs, one would draw bounding boxes around each dog in every image or video frame and label them as "dog." Once trained, the model will be capable of identifying dogs in new images.

Data collection involves gathering and organizing relevant information to create datasets for machine learning. The type of data collected varies depending on the specific problem the AI model is designed to solve, such as video sequences, frames, images, or patterns. In areas like computer vision, robotics, and video analytics, AI models are typically trained using image datasets to perform tasks like image classification, object detection, and image segmentation. To effectively train these models, image or video datasets must contain pertinent information that allows the model to identify various patterns and make informed decisions. This requires the collection of common events that provide ground truth data, which the model can learn from to improve its predictions.

For example, in industrial safety applications involving optical devices that detect humans and forklifts, it is essential to collect data that includes images or videos of these objects. Cameras in production or logistics settings are employed to capture the required footage to build a dataset. Recent advancements in hardware, analytical methods, network architectures, and image acquisition systems have made it possible to process large volumes of image data. The primary goal of data collection is to gather substantial amounts of diverse data at high speed. However, constructing a well-structured machine learning dataset remains a complex and time-consuming task that requires a systematic approach to maintain quality. The first step is to identify the data sources that will be used for model training. Various techniques are available for collecting image or video data in computer vision applications.

One straightforward option is to utilize publicly available machine learning datasets, which are often free, open-source, and allow modification and redistribution. Public datasets, which may contain millions of data points and annotations, are excellent for training or fine-tuning AI models. These datasets are typically quicker and more cost-efficient than creating a custom dataset from scratch, especially for tasks involving common objects (such as people or faces) or general scenarios. Some datasets are specifically designed and tailored to perform tasks like object detection, facial recognition, or location estimation. However, these may not be suitable for models addressing different problems, in which case a custom dataset would be

necessary.

In most cases, computer vision models are trained using datasets with hundreds or even thousands of images. To enable the AI model to make accurate predictions, effective data gathering is critical. However, with the development of advanced techniques, comparable levels of accuracy can be achieved with smaller datasets. When choosing an image dataset to enhance the performance of a computer vision system, the following factors are particularly important:

- **Image quality**: The images should contain sufficient detail for the AI system to identify and detect the relevant objects. If a human observer cannot easily distinguish the object in the image, the machine learning model will likely struggle to make correct predictions.
- **Data diversity**: A broad dataset helps ensure that the AI model performs reliably across a variety of situations. A lack of variety in the objects, contexts, or scenarios could lead to inconsistencies in the model's predictions.
- **Data quantity**: Having a larger number of images improves the likelihood of obtaining accurate predictions. More data enhances model performance, particularly when the dataset includes many instances of the target objects. In AI training, having more data is always advantageous, as there is no such thing as excessive data.

4.1 Hardware setup for data gathering

Image data can be captured with any type of camera, but utilizing multiple image sources within a single dataset may require additional post-processing for each individual image. Furthermore, if a model is trained using images from various sources and later deployed in a real-world setting, its detection performance could decline due to discrepancies between the training data and the deployment data.

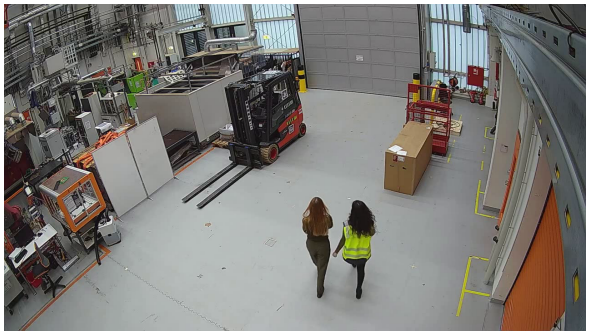
To mitigate this, the hardware setup for data collection replicates the configuration of the entire RTLS system implemented at OHLF. A multi-camera arrangement is employed in the testing environment, consisting of 66 cameras strategically positioned around the shop floor to cover the entire area. This multi-camera setup enables data collection from multiple viewpoints, considering different lighting conditions and potential occlusions. All the cameras transmit data to a central server unit, which oversees the RTLS pipeline. This server also records the data, making it available for generating new datasets that can be used to train the AI model.

Fisheye-lens equipped IP cameras were deployed on the OHLF shop floor to capture images. A dataset for human pose estimation was compiled from 5,893 images, while 17,465 images were used for forklift data collection. These images were annotated using the COCO annotator tool, with 17 keypoints labeled for humans and 8 for forklifts. The data were collected under a variety of conditions, including different poses, occlusions, and lighting situations. Sample images can be seen in Figure 3.

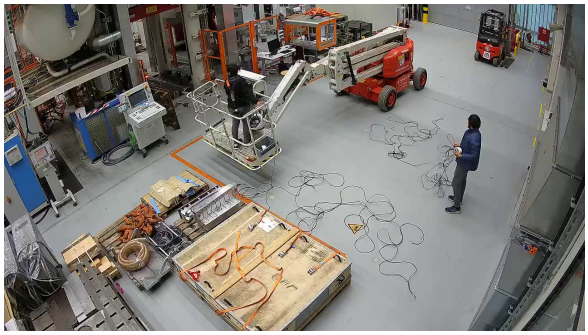
4.2 Data pre-processing and annotation

YOLO is a highly efficient object detection algorithm that delivers accuracy comparable to models like RCNN. However, to achieve optimal performance, it needs to be trained on images that closely resemble the characteristics of the images it will process later. Objects are detected by YOLO using a convolutional neural network (CNN), applied within predefined regions called anchors, centered on each cell of a 13×13 grid overlaying the image. Regardless of the original resolution, all input images are resized to 416×416 pixels for both training and detection. If there is a significant difference in aspect ratios between the training and recognition images, resizing can cause considerable distortion, which may negatively affect detection accuracy.

Figure 2. Digital Twin test environmnt at OHLF



(a)



(b)

Figure 3. Two example images of OHLF collected from two different camera angles. (a) Camera Angle 1, (b) Camera Angle 2.

YOLO's performance declines when there is a considerable mismatch in aspect ratios between training and inference images, as this leads to distortion of object shapes during resizing. To ensure accurate detection, it is crucial that both the training and inference images share similar aspect ratios. A thorough understanding of anchor box characteristics is also vital for improving detection accuracy. Anchors are predefined aspect ratios used throughout the detection process, and their selection should be based on the dataset. A popular approach is to use K-means clustering on object dimensions from the training set, generating anchor boxes that best fit the data. Ultimately, ensuring a consistent object coverage ratio in both training and detection phases is essential for optimizing YOLO's performance. To summarize, achieving the best performance with YOLO requires careful preparation of training images:

1. All training and validation images must be resized to the same dimensions.

2. The object's relative size within the image should remain consistent across both training and detection phases.

Selecting appropriate object classes and gathering relevant data are critical factors in boosting the accuracy of object detection during AI model training. For any image-based AI system, it is essential to organize the dataset efficiently, which includes both the image files (e.g., `.jpg`, `.png`) and their corresponding annotations or labels.

One of the most widely used formats for labeled datasets is the COCO format, which is highly compatible and can be easily converted into other formats if needed. The COCO dataset is structured using JSON and typically contains several key components: `"info"`, `"licenses"`, `"images"`, `"annotations"`, and `"categories"`. These elements help standardize the dataset, making it more efficient and effective for training AI models.

1. **INFO**: This section provides general metadata about the dataset, such as its description, version, year, and contributor information.
2. **LICENSES**: Contains a list of applicable licenses for the images included in the dataset, specifying usage rights and attribution details.
3. **IMAGES**: This part lists all the images in the dataset. It includes metadata for each image (e.g., file name, height, width, and ID) but does not contain any labels, bounding boxes, or segmentation data.
4. **CATEGORIES**: Defines the object classes used in the dataset. Each category (e.g., *dog*, *car*) is assigned a unique ID and belongs to a broader supercategory (e.g., *animal*, *vehicle*). The original COCO dataset includes 90 predefined categories.
5. **ANNOTATIONS**: This section includes all object-level annotations for the dataset. Each entry corresponds to a specific object instance in an image and contains information such as bounding box coordinates, category ID, and image ID. For example, if 64 bicycles appear across 100 images, there will be 64 individual annotation entries for bicycles alone. Multiple instances of the same object in a single image are annotated separately.

Each annotation entry includes several key components. The *image ID* links the annotation to a specific image within the dataset. The bounding box is defined using the COCO format, represented as `[x, y, width, height]`, where `x` and `y` specify the top-left coordinates of the bounding box. The *category ID* corresponds to a specific class listed in the `"categories"` section. Additionally, every annotation is assigned a unique *annotation ID* that distinguishes it from all other annotations in the dataset.

4.2.1 Format of Keypoint

Keypoints offer additional detail about objects by specifying critical points, the connections between them, their locations within the segmentation area, and their visibility status.

In the COCO dataset (as of the 2017 version), keypoints are only defined for the `person` category. However, this concept can be extended to any object class with meaningful anatomical or structural reference points. For instance, keypoints could represent the tail, fins, eyes, and gills of a shark, or components of a robotic arm such as the grabber, joints, and base.

For human figures, keypoints correspond to various body parts, while the *skeleton* defines how these points are connected. For example, a connection such as `[16, 14]` indicates that the `left_ankle` is linked to the `left_knee`.

Keypoint annotations follow a format similar to that used in object detection and segmentation, with one key difference: each keypoint is represented as a group of three values (x, y, v).

- **x** and **y** denote the pixel coordinates of the keypoint within the image.
- **v** represents the visibility status of the keypoint, defined as:
  - $\mathbf{v = 0}$: The keypoint is not labeled; in this case, both `x` and `y` are set to 0.
  - $\mathbf{v = 1}$: The keypoint is labeled but not visible (e.g., occluded or outside the frame).
  - $\mathbf{v = 2}$: The keypoint is both labeled and visible in the image.

For example, the annotation [131, 250, 2] indicates that there is a keypoint at pixel coordinates x=131, y=250, and the value of 2 means that the keypoint is visible.

4.2.2 Choice of annotation tool

There is a wide range of annotation tools available, and choosing the most appropriate one is crucial. Various annotation tools were tested for dataset labeling, as each tool offers distinct functionalities. For example, some tools can only annotate bounding boxes, while others can annotate not only bounding boxes but also keypoints and polygonal segmentation. Given the algorithms employed in this study for training and evaluating the recognition model, it is necessary to annotate both keypoints and polygonal segmentation for the transport vehicle.

Although LabelMe offers functionality that meets the necessary requirements, its usability is suboptimal, and the annotation file format it generates often requires conversion before use. Consequently, COCO Annotator was selected as the tool for dataset creation. COCO Annotator is a versatile, web-based image annotation tool that efficiently labels images for tasks like image localization and object recognition. It provides a wide range of features, including the ability to annotate image segments (or parts of segments), track object instances, annotate distinct visible parts of objects, and export annotations in the standard COCO format. The tool is user-friendly, meets the necessary functionality, and generates annotation files that do not require further conversion.

4.2.3 Data annotation for persons

For the detection of individuals, 17 keypoints are selected for annotation. These keypoints comprises of the `nose`, left and right `eye`, `ear`, `shoulder`, `elbow`, `wrist`, `hip`, `knee`, and `ankle` [83].

4.2.4 Data annotation for forklifts

For forklift keypoints, 8 feature points are selected for annotation. These points include the four wheels of the vehicle and the four corner points of the vehicle's roof. These 8 points are chosen as feature points to assist with vehicle positioning after recognition. In COCO Annotator, these 8 feature points are labeled as `top_front_left`, `top_front_right`, `top_back_left`, `top_back_right`, `bot_front_left`, `bot_front_right`, `bot_back_left`, and `bot_back_right`. Initially, these 8 keypoints are annotated, followed by polygonal segmentation.

Once all image samples have been annotated, a rigorous inspection of the annotation quality is necessary. In addition to the previously mentioned issues with keypoint visibility, there may be other problems such as misaligned keypoints or inaccurate polygonal segmentations. Therefore, it is essential to carefully inspect the final annotated image samples, as incorrect annotations can significantly impact the performance of the training model, leading to false recognition.

Only image samples that have been reviewed and corrected are suitable for use in dataset creation.

While the COCO format is commonly utilized, the YOLO algorithm selected for pose estimation uses the default YOLO format, which stores labels in `.txt` files.

The format for the `.txt` files should follow the specifications outlined in [51]:

1. One text file per image: For every image, a corresponding `.txt` file sharing the same filename should be provided.
2. One row per object: Each row in the text file represents one object instance in the image.
3. Object information per row: Each row includes the following details about the object instance:
    - Object class index: An integer indicating the object's class (e.g., 0 for person, 1 for car, etc.).
    - Object center coordinates: The x and y coordinates of the object's center, normalized between 0 and 1.
    - Object width and height: The object's width and height, normalized between 0 and 1.
    - Object keypoint coordinates: The object's keypoints, normalized between 0 and 1.

Here is an example of the label format for the pose estimation task:

Format with Dim = 2:

$$< \text{class-index} > \ < x > < y > < \text{width} > < \text{height} > \\ < px1 > \ < py1 > < px2 > < py2 > \cdots < pxn > < pyn >$$

Format with Dim = 3:

$$< \text{class-index} > \ < x > < y > < \text{width} > < \text{height} > \\ < px1 > \ < py1 > < p1 - \text{visibility} > \cdots < pxn > < pyn > < p2 - \text{visibility} >$$

In this format, `<class-index>` refers to the index of the object's class, while `<x>`, `<y>`, `<width>`, and `<height>` represent the coordinates of the bounding box. Additionally, `<px1>`, `<py1>`, `<px2>`, `<py2>`, ..., `<pxn>`, `<pyn>` correspond to the pixel coordinates of the keypoints, with each set of coordinates separated by spaces.

The Ultralytics framework utilizes a YAML file format to define the dataset and model setup for training detection models. Below is an example of how the YAML format is used to define a detection dataset:

```
# Dataset configuration for YOLOv8

# Path to the dataset
train: /path/to/train/images
val: /path/to/val/images

# Number of classes
nc: 80
```

```
# Class names
names: ['person', 'car', 'bus', 'truck', 'bike', 'cat', 'dog', ... ]
```

The `"train"` and `"val"` entries define directory paths for training and validation images, respectively. The `"names"` field contains a list of class names, which should correspond to the object class indices in the YOLO dataset files.

For symmetric points, such as those on the left and right sides of a person or face, the `"flip_idx"` field is needed. This field specifies how the indices of symmetric keypoints should be flipped. For instance, consider five facial landmarks: [left eye, right eye, nose, left mouth, right mouth], with the original indices as [0, 1, 2, 3, 4]. The `"flip_idx"` would be [1, 0, 2, 4, 3], meaning the left and right points (indices 0-1 and 3-4) are swapped, while the nose (index 2) remains unchanged.

This is particularly useful for augmentation techniques like flipping the image horizontally during training, ensuring that the symmetry of the object is maintained in the dataset.

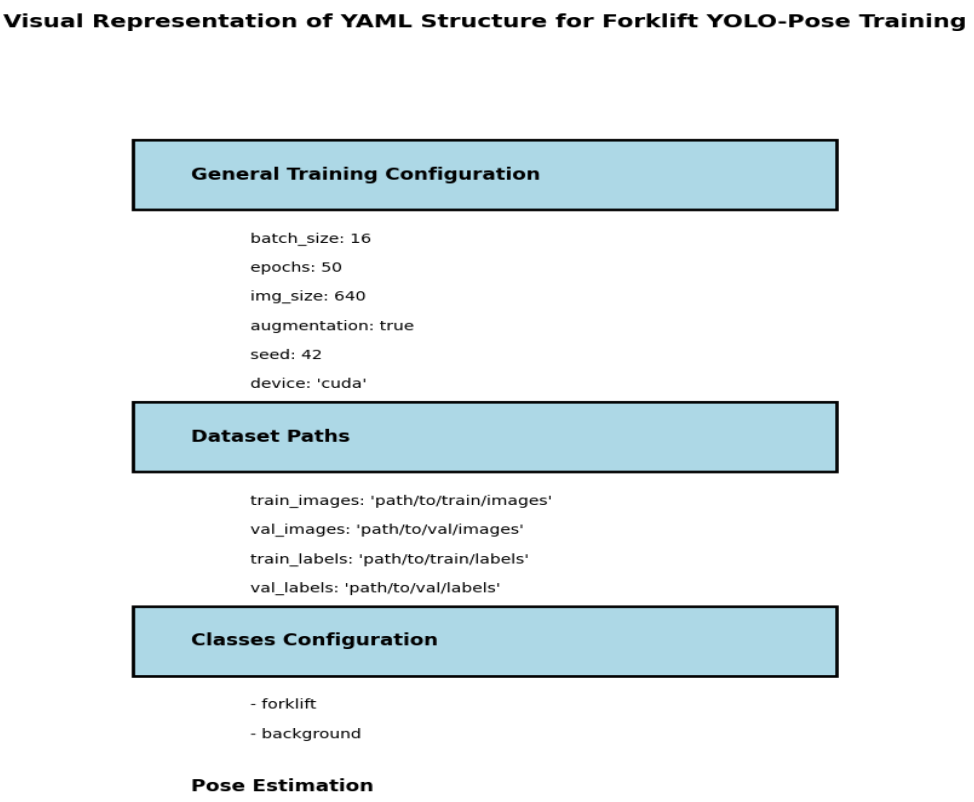


Figure 4. Structure of the YAML file for Forklift YOLO-Pose training

The annotations for forklifts have been converted to the YOLO format, as shown in Figure 4.

4.3 Synthetic Data Generation

Besides relying solely on real-world image collection, this study also leverages synthetic data generation. A digital dataset was developed using the Blender software [26], which enabled the rendering of forklift scenes. Unlike manual data acquisition and annotation, both of which demand significant time and cost, synthetic generation provides an efficient alternative, as the associated annotations can be inherently included. When estimating human or machine poses, the labeling task becomes more intricate, involving not only numerous keypoints (17 for people, 8 for forklifts) but also a visibility indicator to support more robust detection. The present work focuses on embedding these visibility flags directly within the annotation pipeline to enhance data generation procedure.

4.3.1 Motivation Behind Synthetic Data

Data Imbalance: Real-world datasets may lack sufficient examples of critical scenarios such as near-misses or collisions, which are often rare but vital for training. The challenges of data collection in industrial environments include:

- **Privacy Concerns**: Safeguarding the anonymity of workers while ensuring compliance with ethical standards during data capture.
- **Hazardous Conditions**: Capturing real-world data in active industrial environments can be difficult and unsafe.

Synthetic data provides a viable solution by addressing these challenges through the following advantages:

- Scenario Diversity: Generating diverse scenarios covering a wide range of forklift-human interactions and environmental variations.
- Precision Annotations: Automated and consistent labeling of keypoints, poses, and trajectories for forklifts and humans in 2D and 3D.
- Scalability: Ability to generate a virtually unlimited amount of data for training, including rare events.

4.3.2 Simulation Environment Design

The synthetic data generation process has been developed in Blender. Blender is an open-source 3D rendering software with a vast public community. It can be integrated with Python programming to automate the process of scene preparation and rendering. Images captured using IP cameras installed in the warehouses are used as backgrounds for preparing the blender scenes. This helps in recreating original image scenarios. Resources like 3D models of forklifts, humans, and random warehouse objects can be either modeled or purchased online. For human models, an open source software called Animations are added to humans and forklifts to simulate various interactions, which are not always possible to recreate in reality. Along with animation, commonly used accessories like vests and helmets of various colors have also been added. In the following subsections, each aspect of the synthetic data generation process is discussed elaborately.

4.3.2.1 Human and Forklift Model design and Annotation Forklift models are an integral part of the synthetic data generation process. **3D forklift models** were acquired from public repositories or offering industrial vehicle models in formats like **OBJ, FBX, and STL**, or purchased online. The quality, detail, and polygon complexity of these models differ,

necessitating preprocessing to make them appropriate for keypoint-based analysis and simulation in **Blender**. MakeHuman has been used [52]. MakeHuman is platform-independent human modeling software where realistic-looking human models can be generated with variations in ethnicities, skin color, hair, age and clothing textures. This helps in adding variation to the human models to used in synthetic data generation process. These models are processed further in Blender to be made suitable for our use cases. The human models are then rigged using the Maximo web portal, which is very useful to generate animations to diversify the synthetic image data.

The first step involved importing the selected forklift model into Blender using the built-in import functions. The imported model was visually inspected in **Object Mode** to assess its structure, complexity, and scaling relative to real-world dimensions. Some common issues with downloaded models include high polygon counts, misaligned origins, unnecessary components, and incorrect scaling, which must be addressed before further processing.

To enhance efficiency and ensure smooth performance, **mesh simplification** techniques were applied to reduce the number of unnecessary polygons while preserving the forklift's essential shape. This was achieved using:

- **Blender's Decimate Modifier**, which systematically reduces polygon density
- **Manual cleanup in Edit Mode**, where redundant faces, edges, or vertices were removed.

By reducing the polygon density and cleaning up unnecessary parts of the 3D forklift model, the computational load was minimized, making real-time simulations and keypoint tracking more efficient.

After refining the model, adjustments were made to ensure the forklift matched real-world dimensions. Blender's **transform tools (Scale, Rotate, Translate)** were used to correctly align the model within the 3D space. The forklift's pivot point was set to the center of mass to ensure accurate movement tracking during simulation. Once preprocessing was completed, the model was saved in Blender's native **.blend format** to preserve modifications. Additionally, an exported version in OBJ or FBX format was created for compatibility with external applications if required. Figure 5 and Figure 6 shows examples of 3D Forklift and Human Models used in our synthetic data generation process.

4.3.2.2 Adding Keypoints Using Vertices and Edges

After refining the forklift model, key structural points were identified to aid pose estimation and movement tracking. To define these keypoints:

1. The model was switched to **Edit Mode** within Blender.
2. The **Vertex Tool** was used to add new vertices at each of the eight designated keypoint positions.
3. The **Snap Tool** (`Shift + S`) was utilized to accurately align the vertices with the forklift's structure.
4. Edges were drawn between selected keypoints to form a **skeletal representation** of the forklift.

These defined keypoints served as reference markers for tracking **motion**, conducting **pose estimation**, and performing **collision detection** in subsequent simulations. The **eight keypoints** were strategically chosen to capture essential forklift components and track movement accurately. The keypoints were placed at:

Figure 5. 3D Model of Forklift

- **Front-left fork tip** – Represents the leftmost lower extension of the forklift's forks.
- **Front-right fork tip** – Represents the rightmost lower extension of the forklift's forks.
- **Top of the mast** – Indicates the highest point of the forklift's lifting mechanism.
- **Rear-left wheel** – Denotes the left rear wheel, important for tracking the vehicle's motion.
- **Rear-right wheel** – Denotes the right rear wheel, important for tracking the vehicle's motion.
- **Chassis front-left corner** – Marks the leftmost front corner of the forklift's body.
- **Chassis front-right corner** – Marks the rightmost front corner of the forklift's body.
- **Chassis rear-center** – Serves as the central reference point for the forklift's orientation.

For Human Keypoints 17 vertices and connected edges representing limbs were added to the human model following COCO human keypoint definition. Each of these points was strategically placed to ensure they provided **valuable data** during the human and forklift simulation.

4.3.2.3 Storing Keypoints for Future Use Once the keypoints were established, they were saved as a distinct mesh object in Blender. This setup allowed the keypoints to be manipulated separately from the forklift model. Furthermore, each keypoint was assigned **unique identifiers** to simplify access and tracking through **Python scripts**. Figure 7 and Figure 8 shows examples of 3D Forklift and Human Models with keypoints for pose used in our synthetic data generation process.

By carefully selecting, adding, and organizing these keypoints, the model was enhanced for **further simulations, tracking, and AI-driven pose estimation applications**. To ensure that all possible scenarios were covered, the following categories of poses were simulated and

Figure 6. 3D Model of Forklift

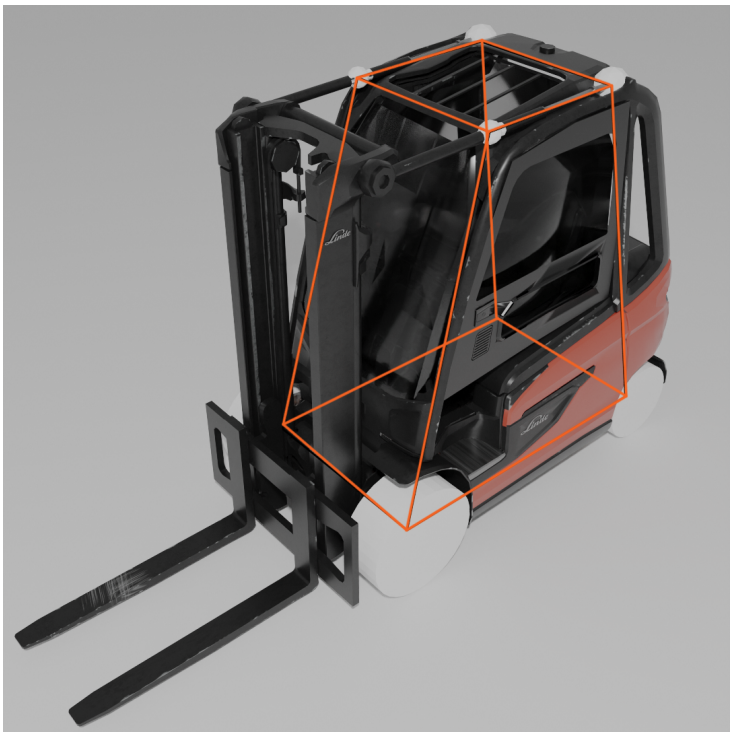


Figure 7. 3D Model of Forklift with keypoints and trackers.

annotated. Sphere meshes were added around the keypoints and logic was developed in the Blender coding environment to generate images simultaneosly with and without them, the two images were then compared to get visibility indices for each key point, color level comparision was done to check if the material colour of the center of the sphere meshes match. If they

Figure 8. 3D Model of Human with keypoints.

matched, visibility indes was assigned as 2 (visible).

4.3.3 Forklift Poses:

- Operational states, such as idle, lifting, moving, reversing, and turning.

Detailed annotations included:

- **Keypoints**: Specific locations like mast corners, fork tips, and wheel hubs.
- **Bounding Boxes**: Enclosing boxes that represent the forklift structure.
- **Trajectories**: Path and motion data to analyze speed, direction, and movement patterns.

Figure 9 shows a 3D model of Forklift with Trackers.

4.3.4 Human Poses:

- Common actions such as walking, lifting, crouching, and standing.
- Safety-critical behaviors such as slipping, tripping, and reacting to approaching forklifts.

Annotations included 2D keypoints for body joints (e.g., shoulders, elbows, knees, and ankles).

Figure 10 and Figure 11 shows a 3D model of Humanwith Trackers in T-Pose and while walking. The trackers bend with movemento of the joints.

4.3.5 Interaction Scenarios:

- Collaborative tasks (e.g., directing forklifts, loading/unloading materials).

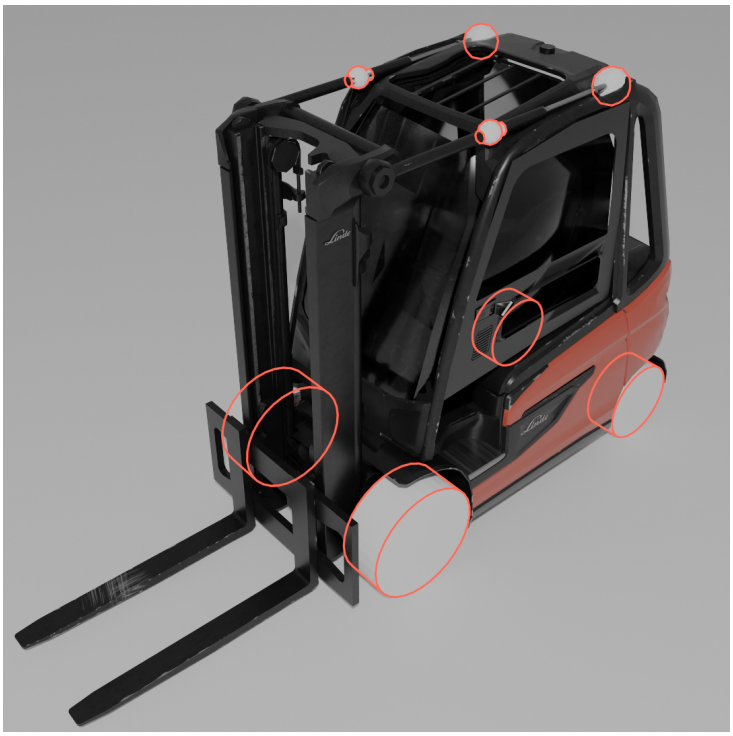


Figure 9. 3D Model of Forklift trackers.

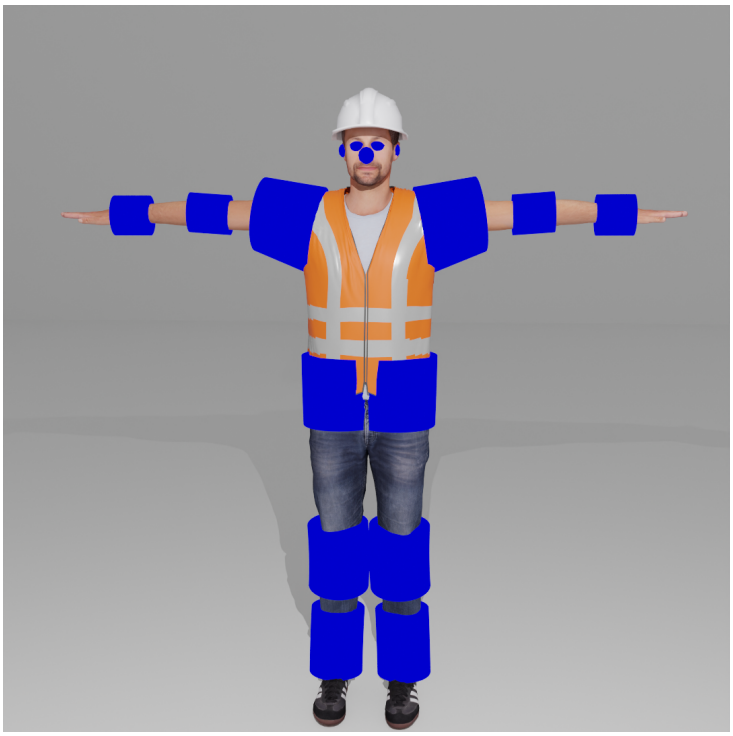


Figure 10. 3D Model of Human with Trackers

- High-risk situations (e.g., proximity violations, obstructed visibility).

Annotations were exported in standardized formats (e.g., COCO, OpenPose) to ensure compatibility with existing machine learning frameworks.

Figure 11. 3D Model of Human with Trackers in Walking pose

4.4 Data Augmentation

To further improve the dataset's applicability, a range of occlusions was introduced. These occlusions were added to simulate real-world scenarios where humans and forklifts are often partially obstructed. This type of obstruction results in the non-visibility of some keypoints, making the visibility index crucial. During the synthetic data generation process, the labels account for the visibility of each keypoint, assigning a value of either 0 (for occluded keypoints) or 1 (for visible keypoints). Mainly, three approaches were used to create such occlusions in the simulation environment.

4.4.1 By creating masks using planes in Blender:

To give an appearance such that an object in a scene is occluding a certain part of a human or forklift, masking was done by importing a plane in the simulation environment. The shape of the plane is then modified in edit mode so that it can cover the certain object that we want to use as an occlusion. After adding the plane, a modifier is applied on the plane to mimic the background, this allows the plane to abstract light to pass through it, thus working as an occlusion. This process is quite effective in creating occlusion but takes very long to edit them to the required shape. In the case of cluttered scenarios, this process becomes cumbersome.

4.4.2 By creating masks using Decals in Blender:

This process needs an image editing software that can cut out an object from an image. Image editing software like Photoshop is very effective in this kind of work but is an expensive tool. Linux has its own free version of photo editing tool called GIMP. Images which are used as backgrounds in simulation environment creation are imported in GIMP. The objects of interest, to be used as occlusion is cut from the images using cutting tool in GIMP. These cut parts are saved with blank ground. In Blender, a third-party Addon called X-Decal can be installed,

which lets these cut parts be imported as a vector that works like stickers on the background. These decals are scaled and translated in appropriate locations to overlay similar parts. They can also be placed randomly in the scenes to allow the creation of more occlusion. This is a very effective method for occlusion simulation. Only downside is that it needs additional software, like Photoshop or GIMP. For our work GIMP has been the choice of tool.

4.4.3 By 3D scanning:

Creating 3D objects using mobile devices has become increasingly accessible thanks to various apps and tools designed for that specific purpose. The applications have intuitive touch control for easy scanning of objects. 3D scanning is done using mainly two technologies, Photogrammetry based and LIDAR based. The models were processed further in softwares like Meshlab developed by Cignoni et al. [14] and Blender [26].

Photogrammetry-Based Scanning:

- **Method**: Captures multiple photographs from different angles to create a 3D model through software analysis.
- **Accuracy**: Dependent on image quality and texture; can be less accurate in low-light or featureless environments.
- **Equipment**: Requires a camera and specialized software.
- **Best Use**: Ideal for capturing detailed textures and colors of static objects.

LiDAR-Based Scanning:

- **Method**: Uses laser pulses to measure distances and create a precise 3D point cloud of the environment or object.
- **Accuracy**: Highly accurate, especially in varying lighting conditions.
- **Equipment**: Requires a device with a LiDAR sensor (e.g., iPhone 12 Pro or later).
- **Best Use**: Excellent for quick, accurate 3D mapping of objects and environments.

Photogrammetry relies on photos and is best for detailed textures, while LiDAR provides high accuracy and is ideal for precise measurements in real-time. For our work we tried both techniques and they give comparably good results. Since our 3D object models are usually far from the camera they do not need very detailed texture.

This method of occlusion creation is very effective, especially when occlusions are to be added externally, they fail to produce good results for the occlusions present in the scene.

For our work, we have used all types of occlusion creation techniques discussed above as and when required. Figure 12 shows scan of a chair used as an object during synthetic data generation process.

4.5 Overview of Synthetic Data Generation Process

The Flowchart in Figure 13 shows the procedure followed to generate synthetic data using Blender. The Blender file creation has also been automated which is a very significant development in making the process of Simulation Environment Design more efficient. The Python script developed to render the data also takes into account parameters like different lighting conditions.

Figure 12. 3D Model of a Chair generated using Photogrammetry and cleaned in Blender to be used as a 3D object during synthetic data generation process

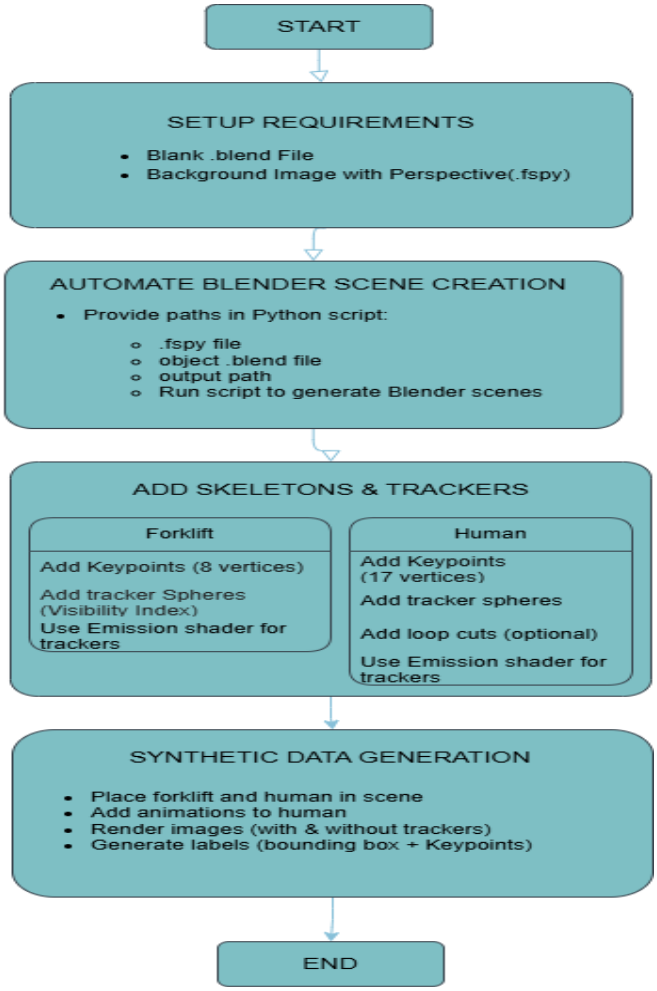
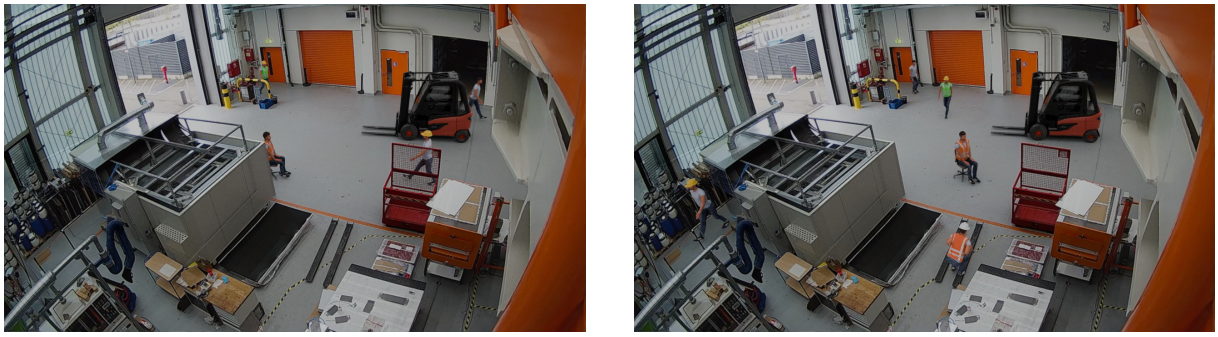


Figure 13. Overview of Synthetic Data Generation Process in Blender

Figure 14 shows an example of two synthetically generated images used in our training set. Figure 15 shows the distribution of data in the forklift and human dataset



(a) **(b)**

Figure 14. Two example images Synthetically generated data with a camera angle of OHLF collected in the background. (a) Forklift and Human in various poses, (b) Synthetic Data with occlusions

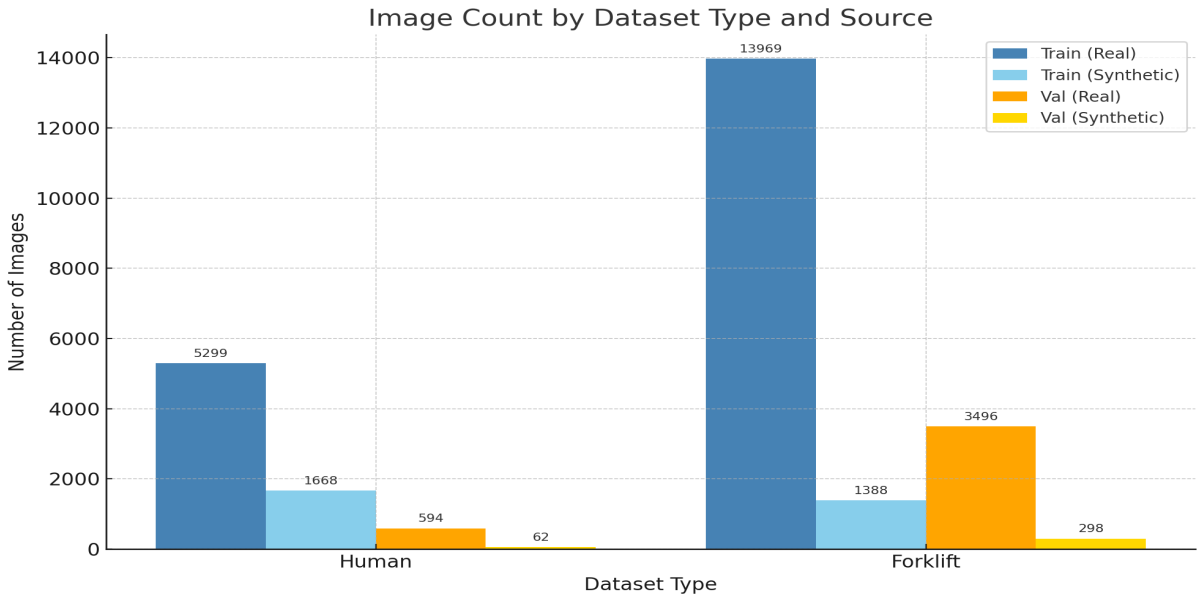


Figure 15. Distribution of labeled data in Training and Validation set for Forklift and Humans

5 Methodology

Analyzing the existing real-time localization framework reveals several potential challenges. One fundamental question arises: can an object identified on a flat image surface be mapped accurately to a global coordinate system? If so, what is the expected precision, and what factors contribute to inaccuracies? Even when operating at peak performance, the current setup might face problems in identifying various poses, especially when occluded. Are there alternative approaches that could improve spatial precision?

5.1 Estimating Position Using 2D Visual Data

Two types of visual data, 2D and 3D, differ significantly. A 2D image is composed of pixels, while 3D images utilize voxels. Researchers have proposed many approaches to extract meaningful information from 2D pixel data. When a real-world scene is captured, it is inherently three-dimensional, but the resulting image is limited to two axes, horizontal and vertical, producing a flat projection. The imaging devices used in this study are cameras, which capture 2D arrays of pixels. Each pixel is defined by its position (x, y) and intensity value. The resolution, depth, and matrix dimensions determine the quality of the digital image. Higher matrix dimensions generally yield better resolution [33].

Figure 16. Representation of a 2D pixel array as the coordinate system for localizing objects in the environment

Image features such as corners, edges, and textures help distinguish objects. For example, a square is recognized by its four corners and edges. As image clarity increases, these features become more prominent. These characteristics are used by machine learning or deep learning models to identify, classify, and localize objects within the frame. However, this localization is limited to the image plane. To project it into world coordinates, camera-specific parameters must be known.

5.1.1 Camera Parameters and Coordinate Transformations

Mapping image points to real-world coordinates involves two parameter sets:

- **Extrinsic parameters:** Describe the camera's physical position and orientation in space.

- **Intrinsic parameters:** Describe how the camera captures the image, including focal length, field of view, and sensor geometry.

These parameters form transformation matrices that convert coordinates between systems. The intrinsic matrix transforms from pixel space to the camera coordinate frame, while the extrinsic matrix maps from camera space to world coordinates.

If we assume a fixed depth (e.g., $z = 0$ for ground plane), then we can estimate a subject's position in the world using image observations. For instance, a ceiling-mounted security camera can determine a person's position on the floor.

This is done using a homography matrix, which defines the transformation between two planes:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Here, H is the 3×3 homography matrix. At least four known point pairs between image and world plane are required to compute this matrix. Once determined, it enables accurate translation from 2D detections to real-world locations.

5.2 Factors Influencing Localization Accuracy

The current system, designed to detect and localize humans and forklifts in industrial environments, employs bounding boxes based on 2D imagery. While the detection models are fast and lightweight, several challenges affect positional accuracy.

Different camera perspectives can result in varying bounding box bases for the same object. The footpoint used for position estimation may appear at different locations depending on the orientation of the object in each frame. In addition, occlusions can hide parts of the object, particularly the lower half, causing the bounding boxes to capture only the upper portion. This leads to inaccurate position estimation when mapped to world coordinates.

5.3 Minimizing Euclidean Distance Between Sensor Estimates

Multiple cameras capture the same object from various viewpoints. Their individual outputs are later fused to produce a single estimate. If four angles detect the same object, pairwise Euclidean distances are calculated. If all lie within an acceptable margin, an Extended Kalman Filter fuses the results [2].

The Euclidean distance between two points is calculated as:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

If one camera reports an inconsistent position compared to others, the object may be falsely treated as two separate instances.

This discrepancy can arise due to:

- Inaccurate camera calibration
- Incomplete bounding boxes
- Orientation mismatches caused by differing camera perspectives

Such issues lead to false representations in the digital twin and incorrect system responses. While the detections themselves may be valid, improper data fusion due to these factors compromises the final result. To mitigate this, precise initial calibration is essential. Relative calibration should be conducted using full image frames to ensure accurate pixel-to-world transformations across all devices.

To reduce the error from bounding boxes, it is essential to pair the bounding box detection algorithm with another method that addresses the limitations of bounding box detection in the current scenario. A suitable choice in this case is a pose estimation algorithm based on keypoints. Even when an object is partially obscured, typically by the foot-point, the data from other detected keypoints can help estimate the foot-points position. This estimate, when combined with bounding box data, can enhance the accuracy and reliability of the final foot-point estimation.

5.4 Open-Source Datasets for Human Pose Estimation

In human pose estimation (HPE) and detection, machine learning and computer vision algorithms face significant challenges. The dynamic nature of human and animal movements adds complexity to the task. The presence of various objects and environmental factors in images and videos further complicates the process. Elements such as clothing, lighting conditions, occlusions, viewing angles, backgrounds, and the tracking of multiple people or animals all contribute to the difficulty of pose estimation. HPE datasets are crucial across numerous fields, including healthcare, sports, retail, security, intelligence, and military applications. To ensure that algorithmically generated models can handle a wide range of data variations and edge cases effectively, training them on large and diverse datasets is critical. This approach leads to the successful achievement of project goals.

In situations where resources such as time or funding are constrained, utilizing freely available and open-source image or video datasets presents a practical alternative. These datasets are typically well-structured for training and deployment, often including pre-applied annotations and labels. By combining multiple open-source datasets or integrating them with proprietary data, the overall dataset size can be increased. However, it is crucial to emphasize that the quality, accuracy, and diversity of the images and videos within a dataset significantly influence the performance of computer vision models. This work leverages this aspect by utilizing pre-trained YOLO pose models, specifically the YOLOv8n, YOLOv8s, and YOLOv8m versions. These models have been trained on well-known datasets such as COCO and MPII, among others. To enhance the generalization capacity, we fine-tune these models on a custom dataset containing images of forklifts and humans. Additionally, we have explored the use of synthetically generated data for both forklifts and humans to further train and validate the models.

5.5 Evaluation Metrics for Object Detection and Keypoint Detection Algorithms

To understand the distribution of data during training and perform well during inference, deep learning algorithms require effective evaluation metrics. The performance of a Pose Estimation model can be evaluated using various methods, but careful selection of the appropriate metric is crucial. One key distinction to make is between pixel-wise and object-wise evaluations.

Pixel-wise evaluation checks the accuracy of pixel labels, whereas object-wise evaluation focuses on matching specific objects in the ground truth with identifiable objects in the output. Pixel-wise evaluation may cause an issue when objects overlap, such as in layered images, where a

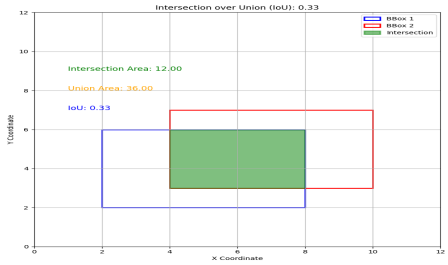


Figure 17. The computation of Intersection over Union (IoU) involves dividing the area of overlap between the bounding boxes by the area of their union.

single pixel could be assigned to multiple classes. On the other hand, object-wise evaluation must account for scenarios involving many-to-one or one-to-many detections.

It is crucial to acknowledge that the selection of evaluation metrics is contingent upon the specific application. For example, when performing keypoint detection in pose estimation, identifying a single pixel for a joint might suffice. In this section, some of the assessment metrics needed for pose estimation are briefly discussed.

- **Intersection over Union (IoU) / Jaccard Index (JI)** : A common way to evaluate the correctness of an object proposal in object detection is Intersection over Union (IoU). It serves as an evaluation metric to assess the accuracy of an object detector on a given dataset. The application of IoU is based on the available ground truth data and output predictions, which, in our case, are bounding boxes. Computing IoU (as shown in Figure 17) can be determined by:

$$IoU(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cap \boldsymbol{B}}{\boldsymbol{A} \cup \boldsymbol{B}}; \quad IoU(\boldsymbol{A}, \boldsymbol{B}) \in [0, 1]$$

  Suppose set A is the set of predicted bounding box pixels and set B is the set of true bounding box pixels, then the computation will be done as:

$$AreaofOverlap = A \cap BAreaofUnion = A \cup B$$

- **Accuracy**: Accuracy is defined as the total number of correct predictions over the total number of predictions. It is calculated as:

$$Accuracy = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{TN} + \boldsymbol{FN}}$$

  where TP = true positive, TN = true negative, FP = false positive, FN = false negatives.

- **Precision and Recall**: Precision, often referred to as positive predictive value, can be expressed as:

$$\boldsymbol{precision} = \boldsymbol{PPV} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

  Precision can be defined as the fraction of positive predictions that truly belong to the positive class.

  Recall (also known as sensitivity) can be represented as:

$$\boldsymbol{recall} = \boldsymbol{sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

The precision-recall (PR) curve illustrates how recall in a computer vision model changes for a given precision and vice versa. A model with a high area under the curve signifies strong recall and precision, while a smaller area under the curve indicates lower recall or precision. The process of plotting the model's precision and recall as a function of the model's confidence threshold is the precision-recall curve.

- **Mean Average Precision (mAP)** : Average Precision is determined by calculating the weighted mean of precisions at each threshold, with the weight being the increase in recall compared to the previous threshold. The mAP is calculated by finding Average Precision (AP) for each class and then averaging over a number of classes.

$$\boldsymbol{mAP} = \frac{1}{|\boldsymbol{O}|} \sum_{c \in \boldsymbol{O}} \boldsymbol{AP}(c)$$

The mAP captures the balance between precision and recall while accounting for both false positives (FP) and false negatives (FN), making it an ideal metric for a wide range of detection tasks.

- **F1 Score** : The F1 score measures the equilibrium between precision and recall [23]. It identifies the optimal confidence score threshold where both precision and recall achieve their highest values. A high F1 score indicates high precision and recall, and vice versa.
- **Percentage of Correct Key-points (PCK)** : The Percentage of Correct Keypoints (PCK) is employed as an accuracy metric, evaluating whether the distance between the true joint and the predicted keypoint falls within a predefined threshold [58]. PCK is generally set with respect to the scale of the object, which is enclosed within the bounding box.

  Thresholds can be either 0.5 or 0.2.

  - PCK@0.5 denotes PCK when the threshold = 50% of the head bone link.
  - PCK@0.2 denotes PCK when distance between predicted and true joint ¡ 0.2 * torso diameter.
  - Sometimes, 150 mm is taken as the threshold.

  It mitigates the issue of shorter limb lengths, as shorter limbs are associated with smaller torso and head bone segments [81].

- **Object Keypoint Similarity (OKS)** : OKS is calculated using the distance between ground truth points and predicted points and is normalized by the scale of the person. Keypoint and Scale constants are needed to equalize the importance of each keypoint. For example, neck location should be considered more precise than hip location.

$$OKS = \exp\left(-\frac{d_i}{2s^2{k_i}^2}\right)$$

  - d_i is the Euclidean distance between ground truth keypoint and predicted keypoint.
  - s is scale: the square root of the object segment area.
  - k is the per-keypoint constant that controls fall-off.

  OKS only indicates how close the predicted keypoint is to the true keypoint (value from 0 to 1). One important part of OKS is Average Precision with a threshold. Commonly used threshold values are 0.5 or 0.75. OKS needs to be compared with the threshold, and if it is

greater, the keypoint is considered as detected. The OKS is more difficult to calculate than PDJ. Ideal predictions will yield an OKS value of 1, whereas predictions with keypoints deviating significantly (by more than a few standard deviations, $s \times k_i$) will result in an OKS value of 0 [34].

5.6 Choosing the keypoint detection algorithm

Present two-stage heatmap-driven frameworks lack efficiency, as they utilize a proxy L1 objective function for learning, which does not align well with the Object Keypoint Similarity (OKS) metric. Additionally, these systems do not facilitate end-to-end optimization. Extracting human pose information from images containing several individuals is a nontrivial task due to the unpredictable count of people, scale discrepancies, partial occlusions, deformable structures, and various other complexities.

As the count of individuals within a frame increases, top-down pipelines scale proportionally in complexity. This linear growth, combined with unpredictable execution durations, renders them impractical for scenarios requiring real-time inference. In contrast, bottom-up frameworks [8], [55], [59], [61], [41] maintain consistent processing times. These systems operate by leveraging spatial probability maps (heatmaps) to detect all joint locations in one pass, followed by assigning detected landmarks to specific identities using advanced downstream operations. These post-processing stages may involve steps like neighborhood-level maxima filtering, vector field integration, position refinement, or clustering. While Non-Maximum Suppression helps identify peak activations, additional steps are applied to adjust coordinates and reduce discretization errors introduced by resolution downscaling. Despite such enhancements, overlapping or adjacent joints can remain difficult to distinguish due to blurred confidence maps.

Furthermore, these bottom-up procedures cannot be jointly optimized through a single computational graph, as their post-inference stages fall outside the network's learning scope. Various techniques differ significantly in how this grouping is handled, ranging from mathematical programming solutions [29] to custom-engineered heuristics [8]. These steps often require substantial computation and do not benefit from parallelism accelerators designed for CNNs. While unified pose predictors [57] attempt to overcome these segmentation issues, they still trail traditional bottom-up methods in terms of prediction precision, often relying on auxiliary refinements to enhance accuracy.

The YOLO-Pose paradigm incorporates anchor-based regression, where each anchor is mapped to its respective box and complete pose coordinates. It solves the ambiguity of nearby keypoints from different individuals by associating them with distinct anchors. Heatmaps can struggle when spatial proximity between similar joints arises, but anchor separation avoids this confusion by pre-grouping each keypoint to its associated anchor. Hence, no further association is necessary. As seen in Figure 18, visual clutter can cause cross-identity confusion in traditional bottom-up methods, while our method inherently avoids this flaw.

Unlike traditional top-down approaches, YOLO-Pose inference remains unaffected by the number of detected persons. Its consistent performance, coupled with a streamlined post-processing architecture, effectively integrates the advantages of both top-down and bottom-up approaches, striking a balanced trade-off between speed and accuracy.

YOLO-Pose adopts a bottom-up approach for human pose estimation using a single input image. Unlike traditional methods that rely on heatmaps, YOLO-Pose directly associates all keypoints of an individual with predefined anchor points. This anchor-based formulation eliminates the need for complex grouping algorithms during post-processing. Built upon the YOLOv8 object detection architecture [20], YOLO-Pose is highly extensible and can be integrated into various

Figure 18. HigherHRNetW32 output demonstrating how the grouping method may easily go wrong even when the keypoint locations are largely accurate. In cluttered situations, bottom-up techniques are susceptible to these grouping mistakes. Adapted from [51]

computer vision frameworks. The complete architecture, including the keypoint prediction heads used for pose estimation, is illustrated in Figure 19.

Human pose estimation, in this context, is treated as a single-class detection problem, where each detected person is associated with 17 anatomical keypoints. Each keypoint is represented by a tuple $(x, y, \text{confidence})$, resulting in a total of 51 values per individual. Consequently, for every anchor, the keypoint prediction head outputs 51 components, while the bounding box head contributes an additional 6 elements. The complete prediction vector $\boldsymbol{P_v}$ for an anchor with n keypoints is defined as:

$$\boldsymbol{P_v} = \{\boldsymbol{C_x}, \boldsymbol{C_y}, \boldsymbol{W}, \boldsymbol{H}, \boldsymbol{box_{conf}}, \boldsymbol{class_{conf}}, \boldsymbol{K_x}^1, \boldsymbol{K_y}^1, \boldsymbol{K_{conf}}^1, \ldots, \boldsymbol{K_x}^n, \boldsymbol{K_y}^n, \boldsymbol{K_{conf}}^n\}$$

Keypoint confidence scores are supervised using a visibility flag. Ground truth confidence for a keypoint is set to 1 if it is visible or occluded in the image, and 0 if it is located outside the frame [51].

Each anchor matched to a person in the input image encodes both the bounding box and the complete 2D pose of that individual. The coordinates of the bounding box are normalized relative to the anchor center, and its dimensions are scaled against the anchor's height and width. A similar transformation is applied to the keypoints, with positions normalized relative to the anchor center; however, keypoint coordinates are not scaled by anchor dimensions. This formulation enables both box and keypoint predictions to be centered around the anchor point.

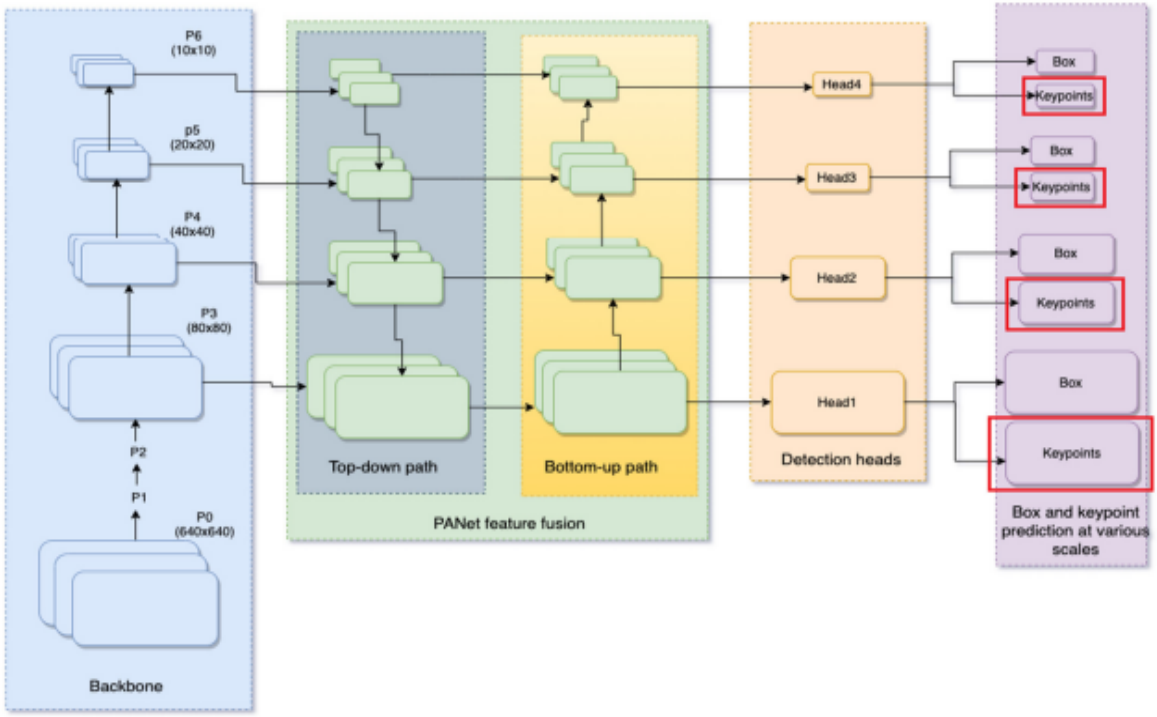


Figure 19. YOLOv8-based YOLO-pose architecture. Darknetcsp backbone processes input picture and produces feature maps at different sizes (P3, P4, P5, P6). These feature maps are fused across several scales using PAnet. The PANet output is sent to the detecting heads. The last branches of each detection head are box head and keypoint head. Adapted from [51]

Importantly, this method is agnostic to the anchor's dimensions for keypoint regression, which allows it to be easily extended to anchor-free object detection architectures such as YOLOX [29] and FCOS [75].

The YOLOv8n, YOLOv8s and YOLOv8m pose models were selected for investigation in this study due to its compact architecture and the fastest inference time among the YOLOv8 variants. Its minimal latency makes it particularly suitable for real-time pose estimation tasks, where rapid processing is essential.

The YOLOv8 pose models were configured using the following set of hyperparameters:

Parameter	Value
Epochs	150
Image Size	640,960
Optimizer	SGD
Initial learning rate	0.01
Learning rate factor	0.01
Weight decay	0.0005
Number of warm-up epochs	3
Warm-up momentum	0.93
Warm-up bias learning rate	0.1

Table 1. Chosen Hyperparameter Values

6 Training and evaluation of the keypoint detection models for person and forklift detection

Several YOLO models were trained to determine which model performs best, namely YOLOv8n, YOLOv8s and YOLO v8m. Two different datasets were used to train these models. One with data collected from Sentics OHLF and several other clients of Sentics GmbH, and another with synthetically generated data added to the original. To train these models two image sizes were used. 640 pixels and 960 pixels. The YOLO models used were pretrained models.

The training process was executed on an NVIDIA RTX 3080 GPU, using a batch size of 16. Training began with 150 initial epochs. The Stochastic Gradient Descent (SGD) optimizer, renowned for its efficiency [85], was applied, with the learning rate gradually declining from 1×10^{-4} to 0 across 30,000 iterations. The model was optimized using the L2 loss function (Least Squares Error).

To assess the learning progression, six evaluation metrics were visualized: **Training Pose Loss**, **Pose Precision**, **Pose Recall**, **Pose mAP@[.50:.95]**, **Pose F1 Score**, and the **Precision-Recall Curve**. These indicators were utilized to analyze performance for both forklift and human datasets, in scenarios with and without the inclusion of synthetic data. The corresponding experimental findings are detailed below.

6.1 Results of Human Keypoint detection model using custom dataset

The plots in the Figure 20 suggest that all the models except YOLO v8m with 640 image size show significant decrease in Pose Loss and the stabilize well. Which means all models except YOLO v8m with 640 image size have trained well with no sign of underfitting. From the mAP@[.50:.95] plots it can be observed the value across models lie in the range of 0.68 to 0.78. Higher the mAP value, better it is in terms of the model to generalize for pose estimation across strict thresholds. YOLO v8s with 960 image size perform best in this regard. It is the heaviest model and takes the highest time to train. YOLO v8n with 640 image size has the lowest mAP but also train the fastest. Pose Presicison is quite consistent among all the models and value lies around 0.92. This shows that there are very few false positives, models are very confident and accurate. YOLO v8m (960) performs siligtly better than the others. Pose Recall is also good for all the models, YOLOv8m(960) being the best. It means that the models are able to detect high number of true keypoints. F1 score for all the models are also high, indicating robust performance and a strong balance between precision and recall, with YOLO v8m being the best. Precision-Recall curve also shows that the models perform well in terms of their ability to predict higher number of pose keypoints with accuracy.

The YOLOv8m_640 model stopped earlier than expected. This could be due to early stopping based on validation loss or performance criteria, overfitting, or a learning rate issue. The model might have converged faster due to its complexity, or insufficient data/augmentation may have caused early stabilization.

6.2 Results of human keypoint detection model using custom dataset and synthetic data

The plots in the Figure 21 suggest that all the models trained on dataset including synthetically generated data also show a significant decrease in Pose Loss and stabilize well. Which means all models have trained well with no sign of underfitting. From the mAP@[.50:.95] plots it can be observed the value across models lie in the range of 0.69 to 0.8. Higher the mAP value, the better it is in terms of the model to generalize for pose estimation across strict thresholds.

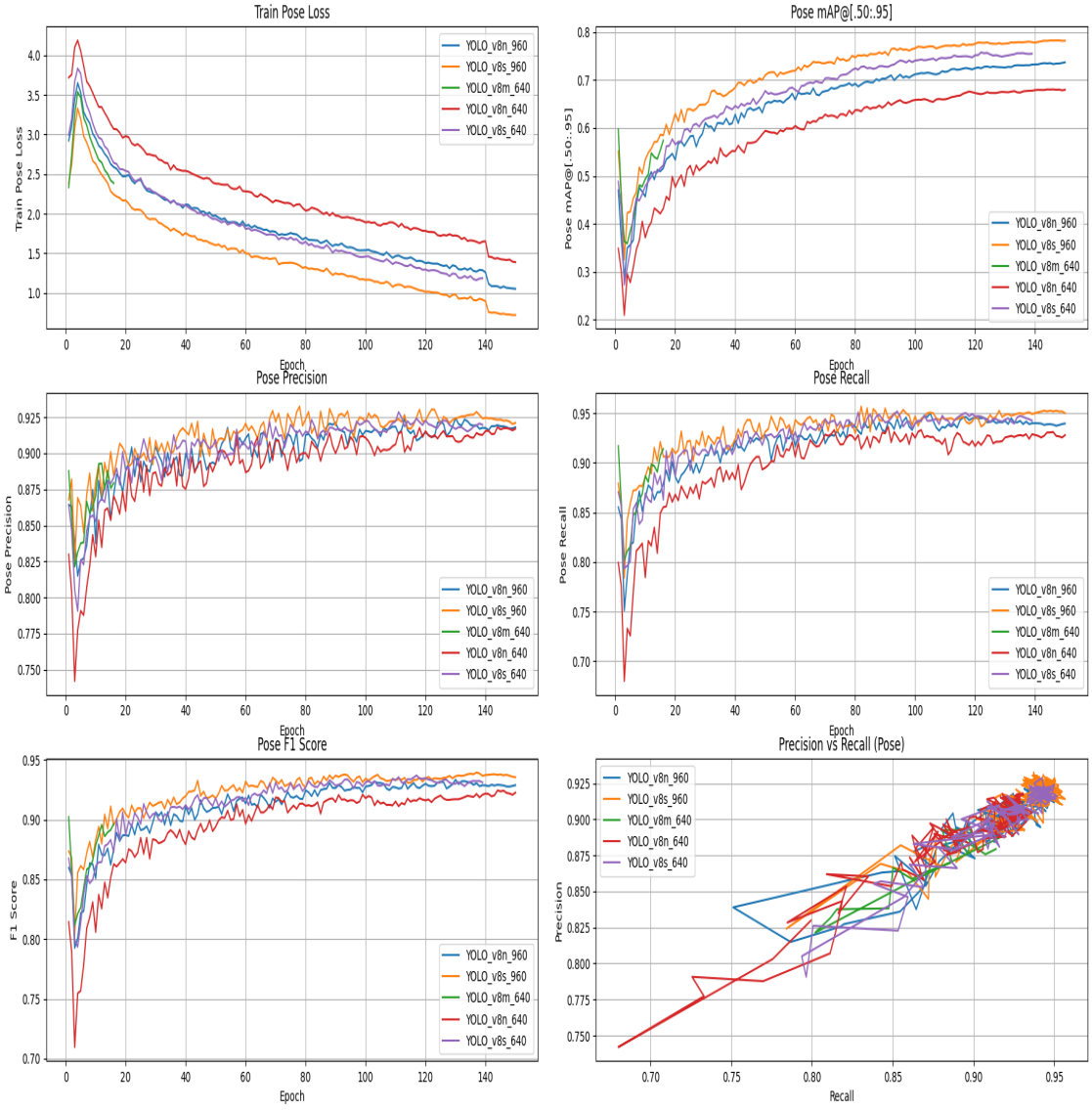


Figure 20. Human Pose Benchmark on Custom Dataset

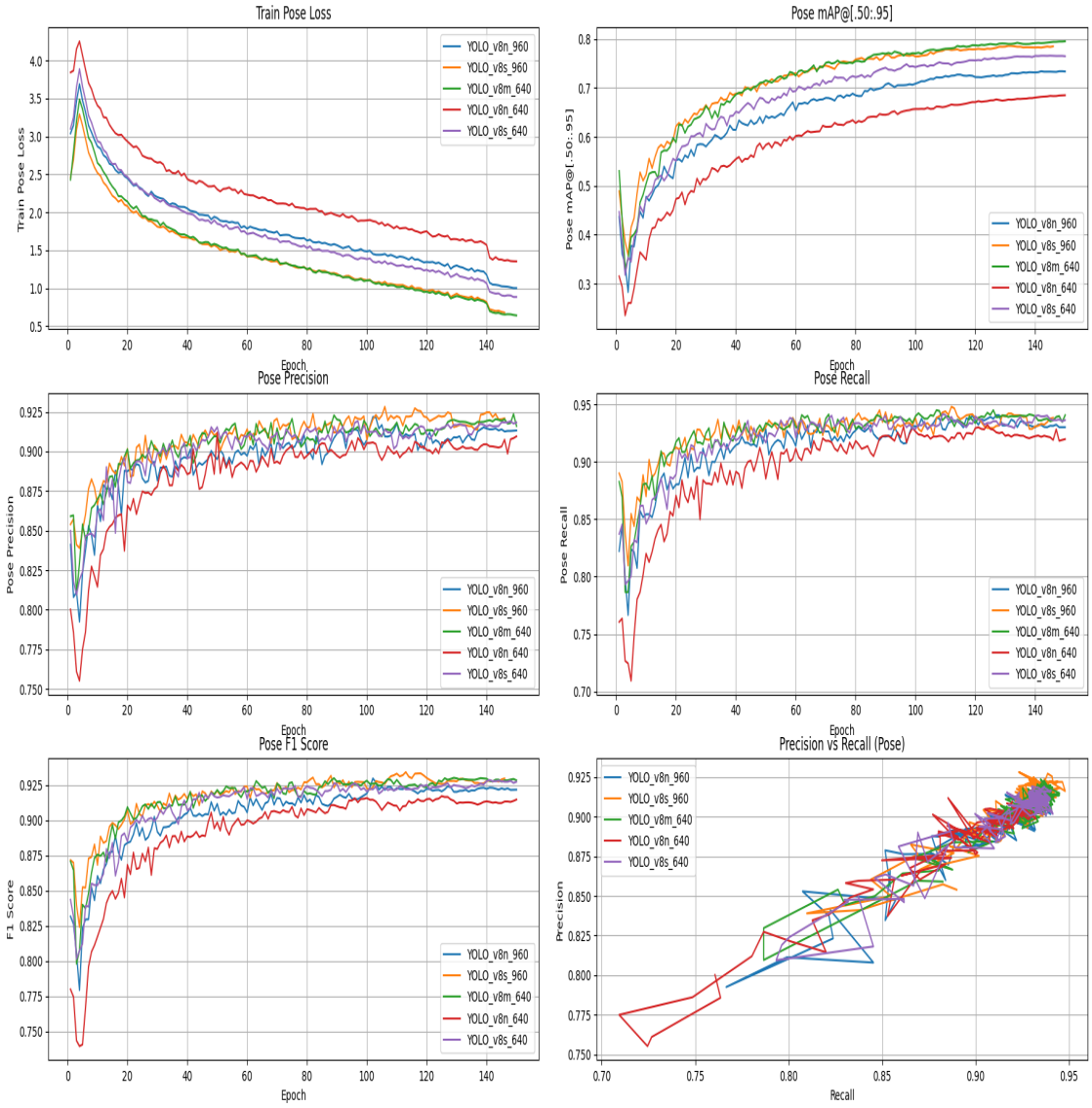


Figure 21. Human Pose Benchmark with Synthetic Data

YOLOv8m_640 perform best in this regard. It is one of the heavier models. YOLO v8n with 640 image size has the lowest mAP but also train the fastest. Pose Precision is quite consistent among all the models, and the value lies around 0.9 - 0.92. This shows that there are very few false positives; models are very confident and accurate. YOLO v8m (960) performs slightly better than the others. Pose Recall is also good for all the models, YOLOv8m(960) being the best. It means that the models are able to detect high number of true keypoints. The F1 scores for all the models are also high, indicating robust performance and a strong balance between precision and recall, with YOLO v8m being the best. Precision-Recall curve also shows that the models perform well in terms of their ability to predict higher number of pose keypoints with accuracy.

There is no early stopping of YOLO v8m_640 model, which could be contributed to the fact that adding synthetic data helped in stabilizing the model training.

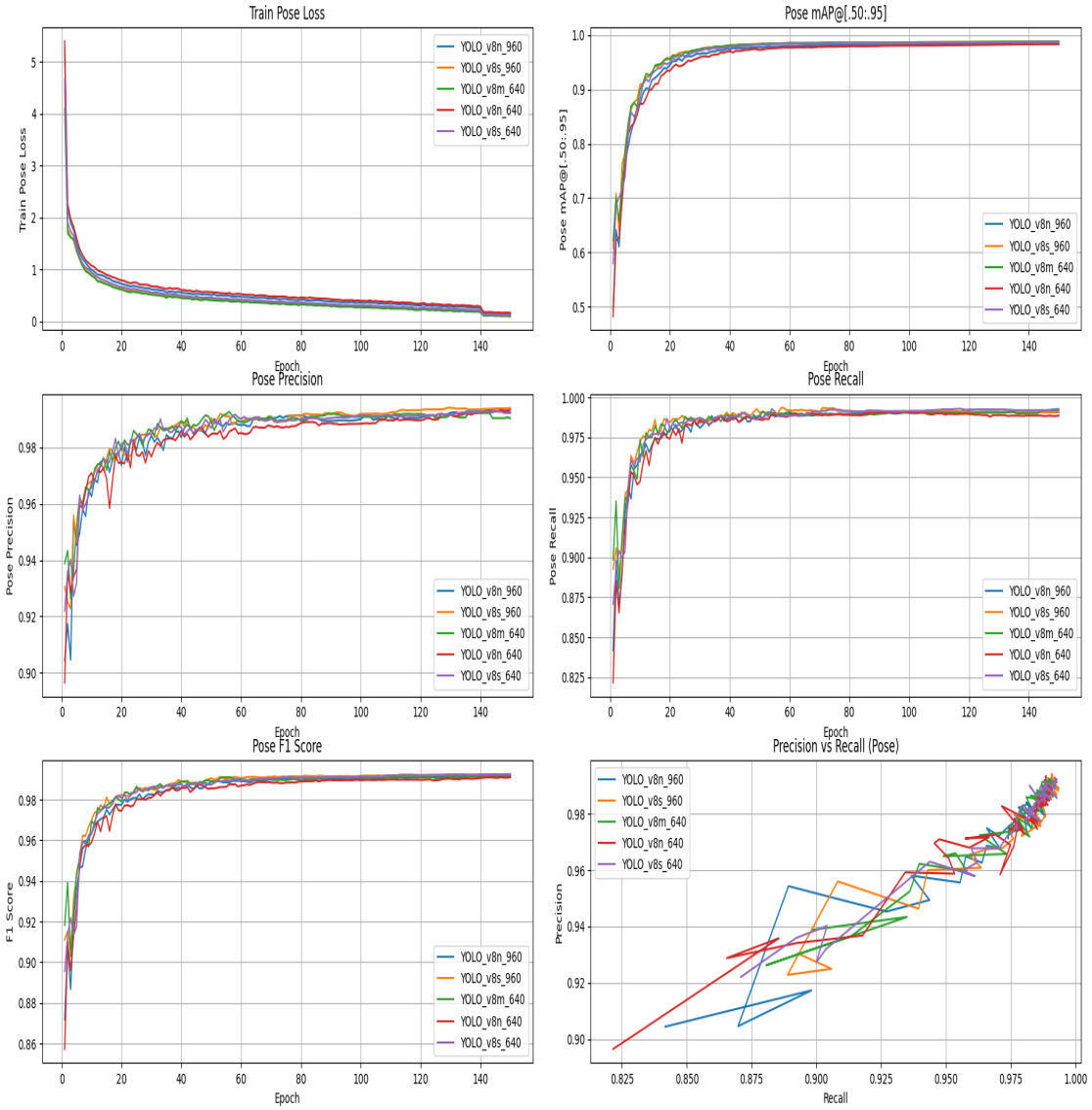


Figure 22. Forklift Pose Benchmark on Custom Dataset

6.3 Results of forklift keypoint detection model using custom dataset

The plots in the Figure 22 suggest that all the models trained on custom dataset, also show significant decrease in Pose Loss and the stabilize well. Which means all models have trained well with no sign of underfitting. From the mAP@[.50:.95] plots it can be observed the value across models lie in near 0.98. Higher the mAP value, better it is in terms of the model to generalize for pose estimation across strict thresholds. Heavier YOLO v8 models perform best in this regard. Pose Precision is quite consistent among all the models and value lies around 0.99. This shows that there are very few false positives, models are very confident and accurate. Heavier YOLO v8 models performs siligtly better than the lighter ones. Pose Recall is also good for all the models, heavier YOLOv8m models being the best. It means that the models are able to detect high number of true keypoints. F1 score for all the models are also high, indicating robust performance and a strong balance between precision and recall, with heavier YOLO v8 models out performing the lighter model slightly. Precision-Recall curve also shows that the models perform well in terms of their ability to predict a higher number of pose keypoints with accuracy.

6.4 Results of forklift keypoint detection model using custom dataset and synthetic data

The plots in the Figure 22 suggest that all the models trained on dataset including synthetically generated data also show significant decrease in Pose Loss and the stabilize well. Which means all models have trained well with no sign of underfitting. From the mAP@[.50:.95] plots it can be observed the value across models lie in the range of around 0.98. Higher the mAP value, better it is in terms of the model to generalize for pose estimation across strict thresholds. Heavier YOLO v8 models perform best in this regard. Pose Presicison is quite consistent among all the models and value lies around 0.99. This shows that there are very few false positives, models are very confident and accurate. Heavier YOLO v8 models performs siligtly better than the lighter ones. Pose Recall is also good for all the models, heavier YOLOv8m models being the best. It means that the models are able to detect high number of true keypoints. F1 score for all the models are also high, indicating robust performance and a strong balance between precision and recall, with heavier YOLO v8 models out performing the lighter model slightly. Precision-Recall curve also shows that the models perform well in terms their ability to predict higher number of pose keypoints with accuracy.

YOLO v8_S has almost identical performance to medium model, but slightly less consistent in mAP. Could be used in deployment as it has a good balance of speed and performance.

Overall performance indicate that heavier YOLO perform the better among the YOLO models. For Human pose estimation, among the models trained on dataset with synthetic data in it, YOLOv8m_640 performs best in terms of the parameter mAP@[.50:.95], which points out that it has the best trade-off between precision and recall on a range of tight bounds. For dataset without synthetic data YOLO v8s_960 perform best. Among the models trained on Forklift data, the general trend is that the heavier mmodel perform slightly better that the lighter onea, although very slightly. Among the lighter models, YOLO v8n(640), the lightest obe, performed the worst in all cases. But its performance is also comparable to the heavier models. It has a advantage that it trains the fastest and in terms of its deployment ability in NVDIA Jetson devices. It is the most suitable candidate. It cases like RTL system where speed of detection is of essence, it can be most suitable.

6.5 Comparing models with and without Synthetic Data

The mAP reflects the trade-off between precision and recall, considering both false positives (FP) and false negatives (FN), which makes it an optimal metric for various detection applications. We have used mAP@[.50:.95] to determine the effect of adding synthetic data to the process of pose estimation. In Figure 24, we have plotted the mAP scores of the best performing models on the Human datasets. it can be observed that the model trained on Human datasets containing synthetic data performs slightly better than the one trained on data without it. A similar trend can be observed in the Figure 25, comparing the small nano models. Though the difference is lesser. This signifies the advantage of adding synthetic data to the training procedure is beneficial to boost the model performances.

For models trained on Forklift data the effect is not very prominent. It can be observed in Figure 26 that the performance of heavier models is very similar for both the datasets. Figure **??** shows the mAPs of worst worst-performing models. It also shows similar performance of the lightest models on both the dataset. It cannot be stated conclusively from these plots if the addition of synthetic data helped in case of Forklift pose estimation.

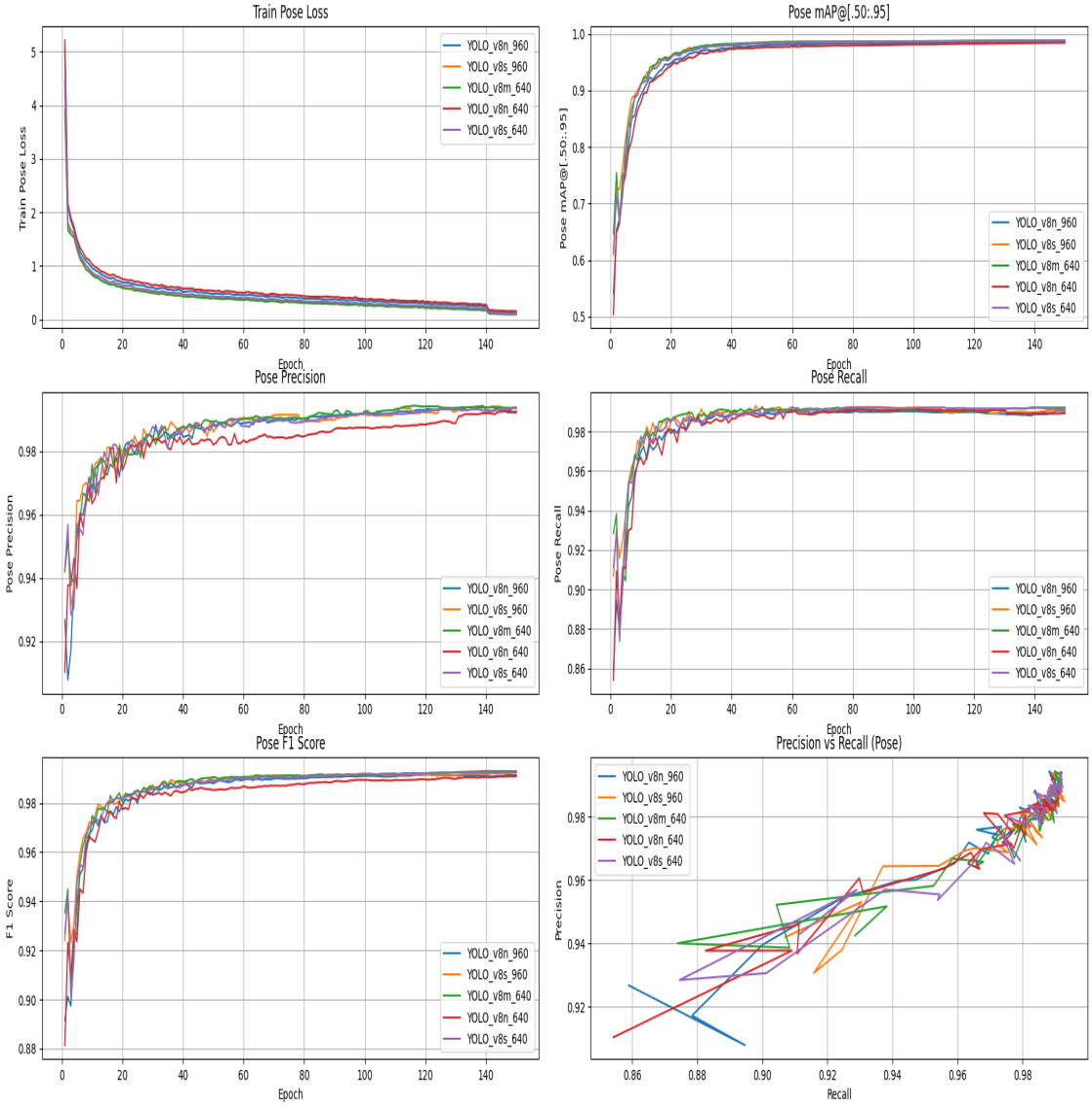


Figure 23. Forklift Pose Benchmark with Synthetic Data

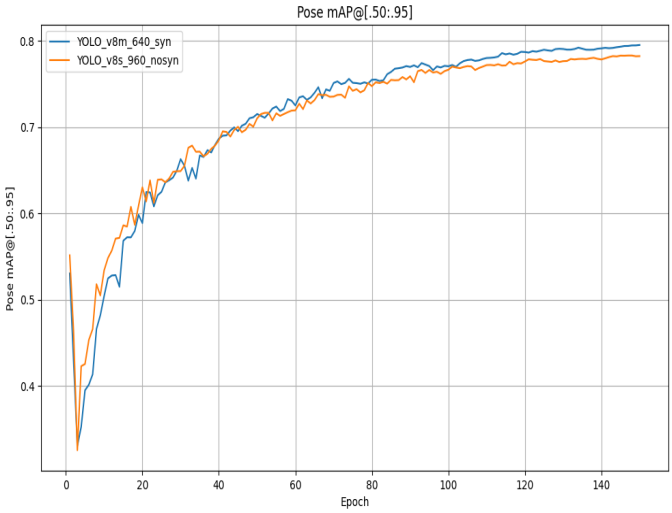


Figure 24. Comparision of mAP@[.50:.95] of Best Performing Human Pose Models on Synthetic Data and Without Synthetic Data

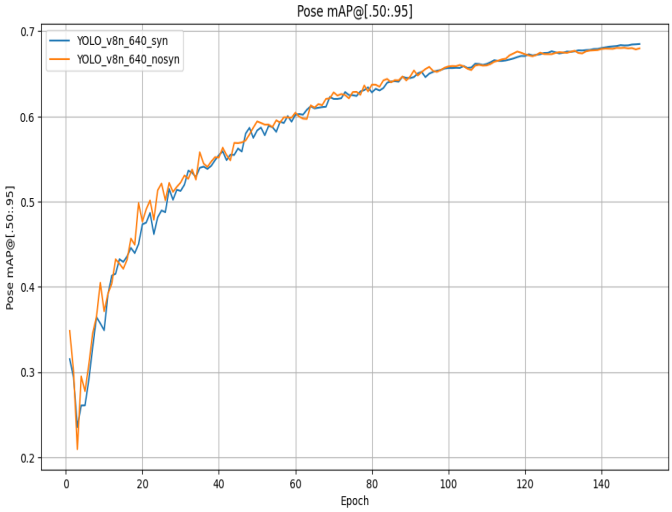


Figure 25. Comparision of mAP@[.50:.95] of Worst Performing Human Pose Models on Synthetic Data and Without Synthetic Data

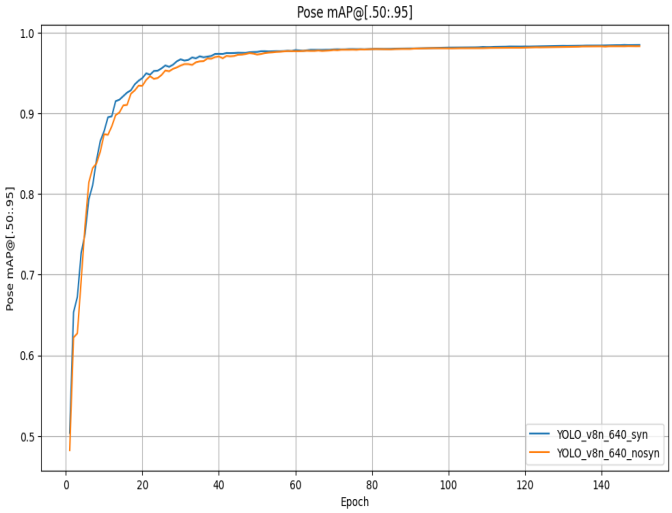


Figure 26. Comparision of mAP@[.50:.95] of Best Performing Forklift Pose Models on Synthetic Data and Without Synthetic Data

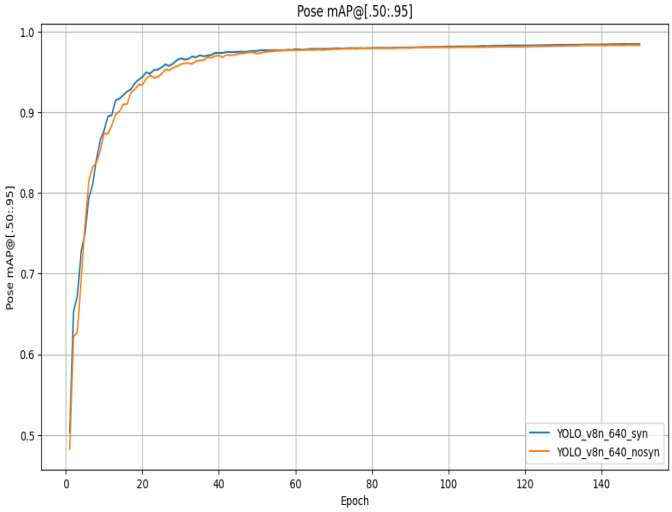


Figure 27. Comparision of mAP@[.50:.95] of Worst Performing Fuman Pose Models on Synthetic Data and Without Synthetic Data

6.6 Results of Human and Forklift Keypoint Detection Model on Test Set

Two test sets were used to test the performances of the models. One was for Human Pose Test, which has 390 labeled images. The dataset for the Forklift Pose Test has 495 labeled images. Comparisons were done between the heaviest and lightest YOLO models, ie. YOLO v8s_960 and YOLO v8n_640. Both the models trained on these configurations with and without synthetic data were done.

It was observed that the heavier model was able to identify more number of human instances compared to the lighter models. Which is expected. Between the models trained with and without synthetic data, the models trained on synthetic data slightly outperform the models trained without synthetic data in terms on number of detection.

Model Name	Number of Correct Predictions
YOLO v8s_960_nosyn	369
YOLO v8s_960_syn	376
YOLO v8n_640_nosyn	358
YOLO v8s_640_syn	358

Table 2. Number of Correct Predictions for Human Models

One notable observation was that the model trained with synthetic data not only predicted a greater number of human instances, but also successfully detected individuals in bending poses. In contrast, the model trained without synthetic data failed to identify such poses. This clearly demonstrates the advantage of incorporating synthetic data during the training process. Figure 28 shows an example

(a) Bending pose detected by model trained on Synthetic data.

(b) Bending pose not detected by model not trained on Synthetic data.

Figure 28. Bending Human Pose detection was possible with model trained on Synthetic Data. (a) Synthetic Data, (b) Non-Synthetic Data.

Model Name	Number of Correct Predictions
YOLO v8s_960_nosyn	462
YOLO v8s_960_syn	466
YOLO v8n_640_nosyn	447
YOLO v8s_640_syn	457

Table 3. Number of Correct Predictions for Forklift Models

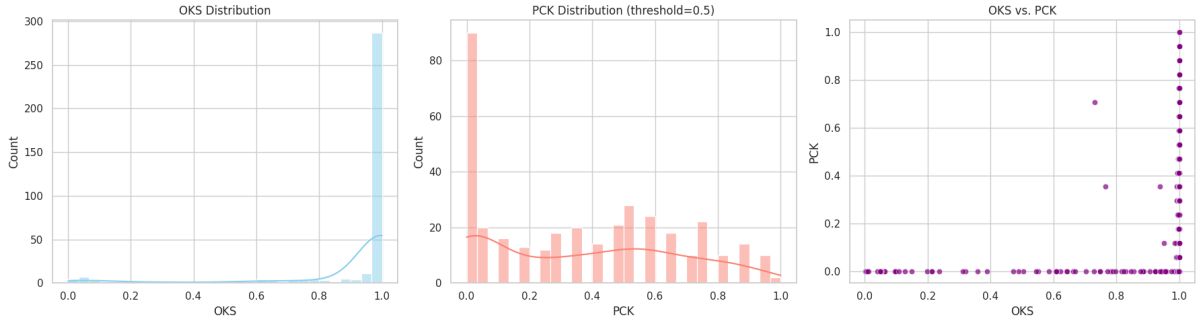
In the case of Forklift models, a similar pattern was observed, the model trained with synthetic data not only predicted a greater number of forklift instances, but also successfully detected forklits placed far away from the camera at a weird angle. In contrast, the model trained without synthetic data failed to identify such forklifts. This again clearly demonstrates the advantage of incorporating synthetic data during the training process. Figure 29 shows an example.

(a) Forklift far away from camera detected by model trained on Synthetic data.

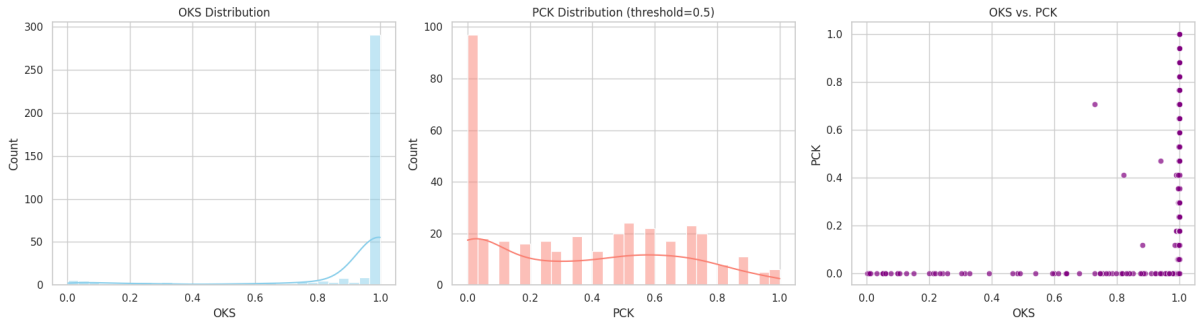
(b) Forklift far away from camera detected by model trained on Non-Synthetic data.

Figure 29. Detection of Forklift far away from camera: comparison between models trained on Synthetic and Non-Synthetic Data. (a) Synthetic Data, (b) Non-Synthetic Data.

Experiments were done with PCK and OKC to demonstrate the quality of the predictions. Figure 30 and Figure 31 show the distribution. It can be observed that both the heavier models trained on synthetic data and custom data show better performance in terms of the accuracy of pose prediction.



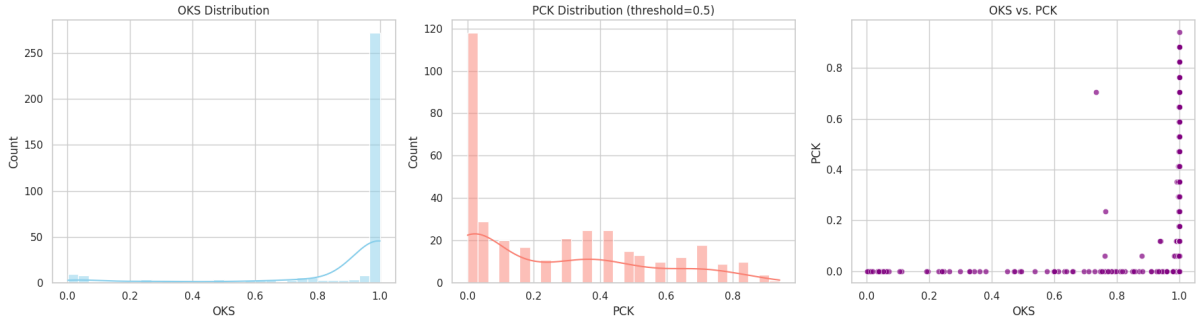
(a) PCK and OKC distribution plots for detected keypoints of Human with model trained on Synthetic data



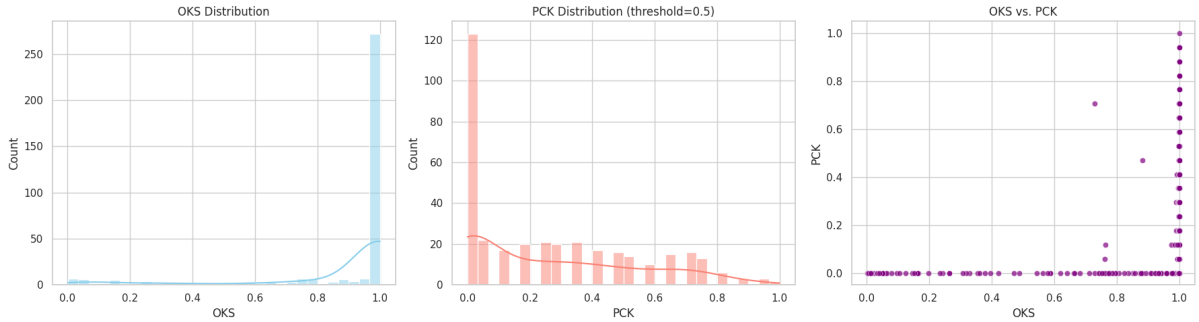
(b) PCK and OKC distribution plots for detected keypoints of Human with model trained on Custom data

Figure 30. PCK OKC distribution to demonstrate the quility of predictions with YOLO v8s_960 model (a) Synthetic Data, (b) Non-Synthetic Data.

From Figure32 and Figure33, It can be observed that the predictions are marginally better with the heavier models, suggesting that they can give more accurate predictions.

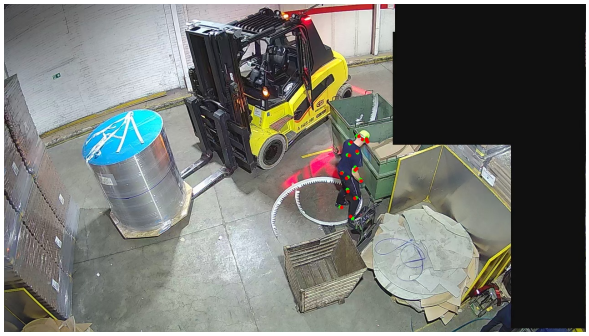


(a) PCK and OKC distribution plots for detected keypoints of Human with model trained on Synthetic data

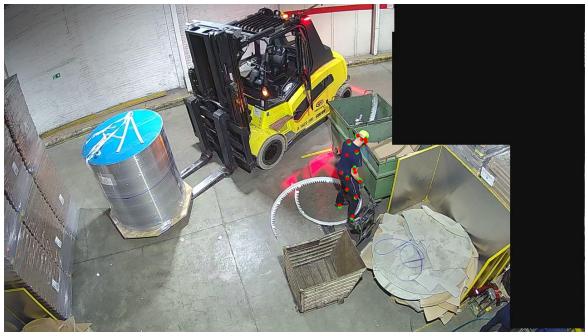


(b) PCK and OKC distribution plots for detected keypoints of Human with model trained on Custom data

Figure 31. PCK OKC distribution to demonstrate the quility of predictions with YOLO v8n_640 model (a) Synthetic Data, (b) Non-Synthetic Data.



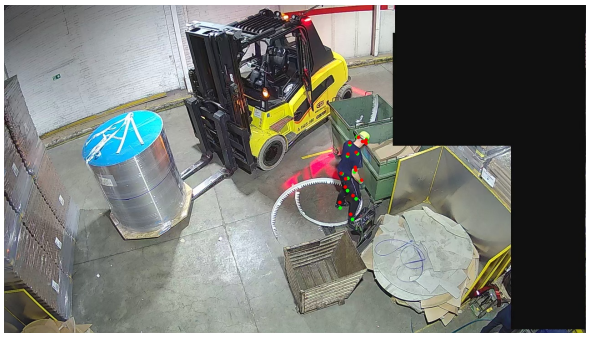
(a) Visual representation of ground truth and prediction keypoints of Human with model trained on Synthetic data



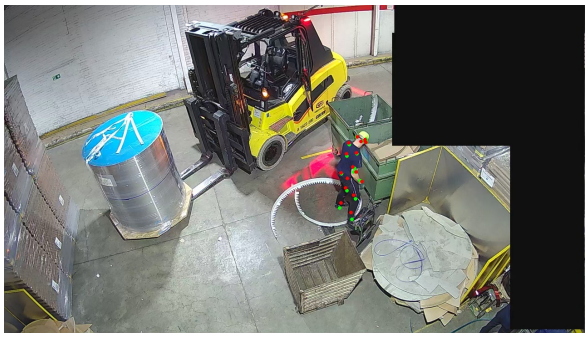
(b) Visual representation of ground truth and prediction keypoints of Human with model trained on Custom data

Figure 32. Visual representation of ground truth (green) and prediction (red) keypoints of Human with model(YOLO v8s_960 trained on Synthetic data and Custom Data, (a) Synthetic Data, (b) Non-Synthetic Data.

Figure 34 and Figure 35 show the distribution. It can be observed that both the heavier models trained with synthetic data and custom data show better performance in terms of the accuracy of pose prediction.

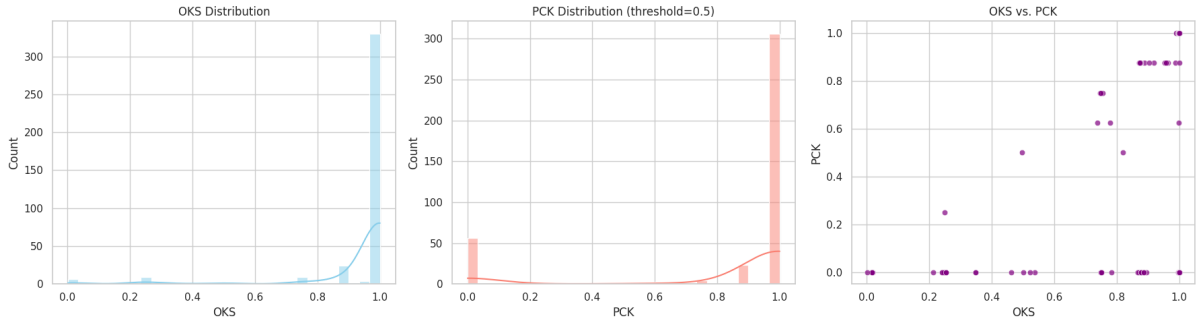


(a) Visual representation of ground truth and prediction keypoints of Human with model trained on Synthetic data

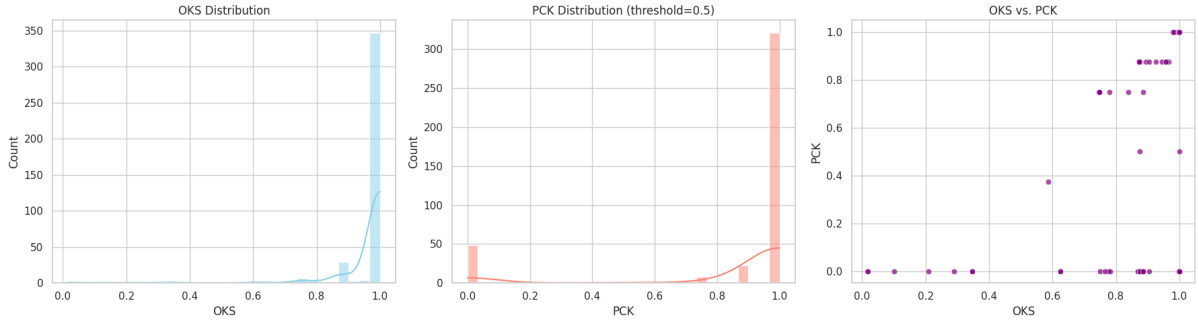


(b) Visual representation of ground truth and prediction keypoints of Human with model trained on Custom data

Figure 33. Visual representation of ground truth (green) and prediction (red) keypoints of Human with model(YOLO v8n_640 trained on Synthetic data and Custom Data, (a) Synthetic Data, (b) Non-Synthetic Data.



(a) PCK and OKC distribution plots for detected keypoints of Forklift with model trained on Synthetic data

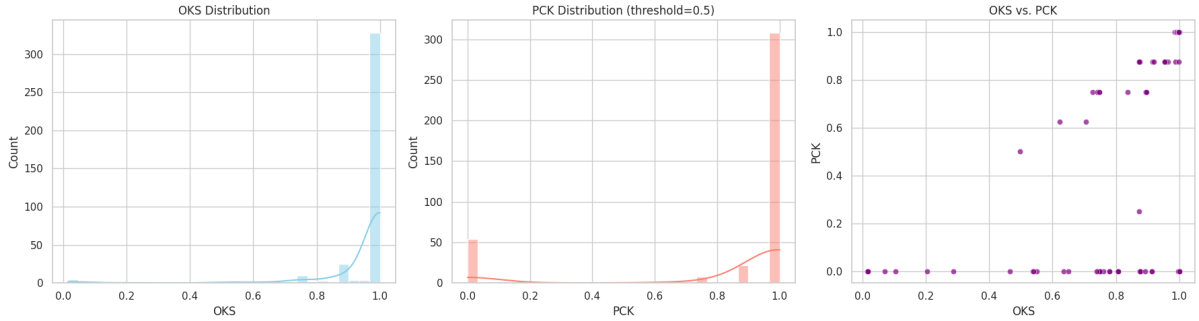


(b) PCK and OKC distribution plots for detected keypoints of Forklift with model trained on Custom data

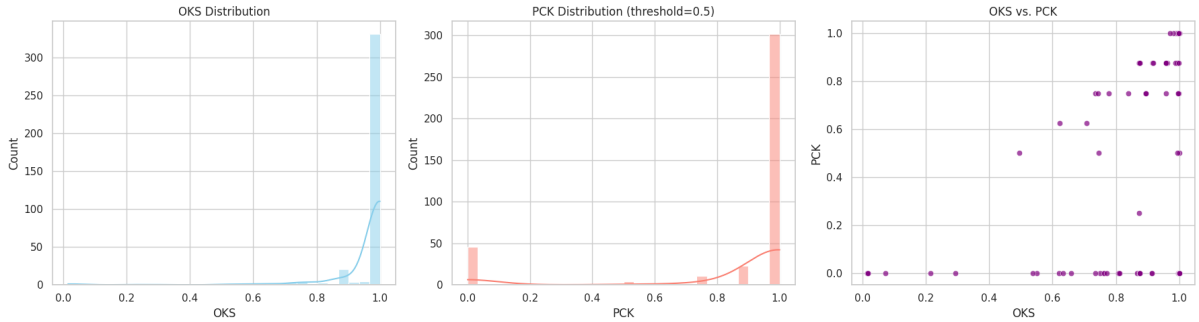
Figure 34. PCK OKC distribution to demonstrate the quility of predictions with YOLO v8s_960 model (a) Synthetic Data, (b) Non-Synthetic Data.

From Figure32 and Figure33, It can be observed that the predictions are comparable across all models.

Large amounts of synthetic data can be produced quickly, and the background of a synthetic image as well as the shape of a synthetic item can both be modified with ease. As a result, it is possible to train a recognition model with greater generalization capability and expand the

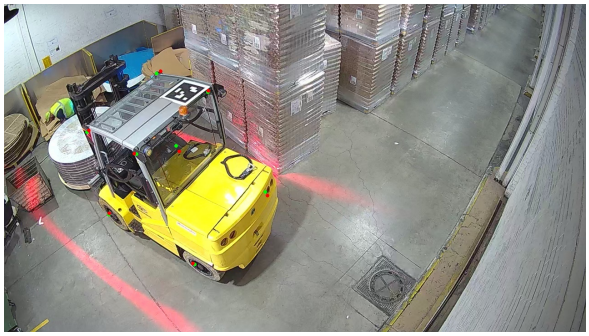


(a) PCK and OKC distribution plots for detected keypoints of Forklift with model trained on Synthetic data

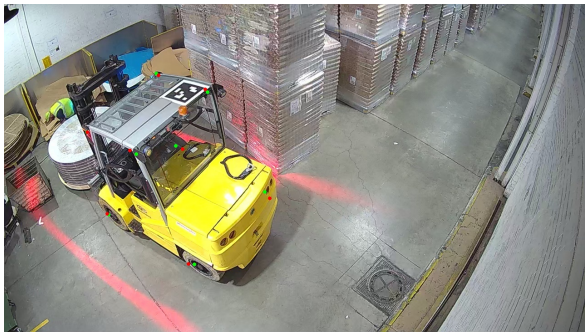


(b) PCK and OKC distribution plots for detected keypoints of Forklift with model trained on Custom data

Figure 35. PCK OKC distribution to demonstrate the quility of predictions with YOLO v8n_640 model (a) Synthetic Data, (b) Non-Synthetic Data.



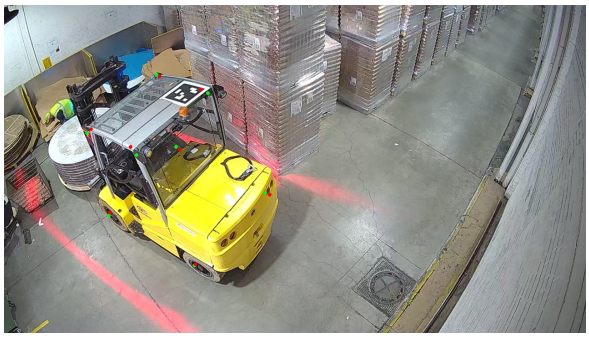
(a) Visual representation of ground truth and prediction keypoints of Forklift with model trained on Synthetic data



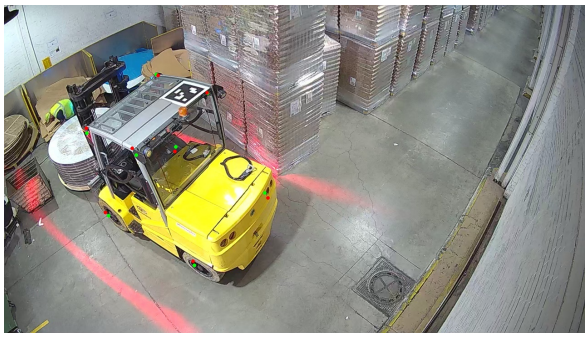
(b) Visual representation of ground truth and prediction keypoints of Forklift with model trained on Custom data

Figure 36. Visual representation of ground truth (green) and prediction (red) keypoints of Forklift with model(YOLO v8s_960 trained on Synthetic data and Custom Data, (a) Synthetic Data, (b) Non-Synthetic Data.

variety of the data.



(a) Visual representation of ground truth and prediction keypoints of Forklift with model trained on Synthetic data



(b) Visual representation of ground truth and prediction keypoints of Forklift with model trained on Custom data

Figure 37. Visual representation of ground truth (green) and prediction (red) keypoints of Forklift with model(YOLO v8n_640 trained on Synthetic data and Custom Data, (a) Synthetic Data, (b) Non-Synthetic Data.

7 Conclusion and Future Work

In the course of this research, pose recognition models for both humans and forklifts were thoroughly investigated by enhancing the quality of the dataset and optimizing the detection algorithms. Initially, the causes of limitations of currently used localization based on bounding box were examined. It was found that foot-point estimation using bounding box coordinates contributed significantly to the errors. This observation led to the adoption of a pose detection approach to achieve more precise, real-time localization of detected objects.

Experiments conducted in this study demonstrated that while existing deep learning models for 2D human pose estimation perform well on widely used open source datasets such as COCO and MPII, their performance degrades significantly in complex industrial environments like the OHLF shop floor. This thesis emphasizes the critical role of high-quality, application-specific annotated datasets in training deep learning models effectively. The availability of diverse annotations, including variations in clothing, use of aprons and masks, and instances of occlusion caused by other individuals, machines, or vehicles proved essential in improving the models' generalization capabilities.

Nonetheless, challenges persist in the domain of 2D human pose estimation. These include high computational demands, potential biases in the training data, and difficulties in accurately detecting poses under occlusions or postures like sitting, crouching, lying down, etc. Due to insufficient annotation quality in the previously collected forklift dataset, it was not feasible to train a reliable recognition model. Consequently, a new dataset was collected and carefully annotated to ensure training efficacy. Experimental results showed that this dataset was sufficient to train a reasonably effective rpose estimation models. Expanding datasets, however, demands substantial annotation effort and time.

To address this limitation, synthetic data generation was explored, as discussed. Synthetic datasets offer the advantage of being generated rapidly and at scale, with automatic annotations. Additionally, synthetic backgrounds and object shapes can be easily manipulated, introducing substantial variety to the data and improving the generalization ability of the trained models. One of the main goals of this work was to develop a synthetic data generation pipeline that can help prepare annotated synthetic data with humans and forklifts. That was successfully developed during the course of this work. The current data generation pipeline can generate

high-quality annotated data in YOLO format containing bounding boxes and keypoints with visibility index for each keypoint. It was achieved using Blender. Other tools were used to generate human and forklift models. All the models were animated in Blender, which made it possible to generate various poses. Provision to add occlusions to the scenes were also developed to make the data more realistic and practical.

This work demonstrated the strong potential of recognition algorithms, particularly when supported by high-quality data. Five YOLO models were tested, and it was obsereved that the heavier models provided better accuracy in terms of key point estimation which was charecterized by their higher mAP scores. One disadvantage of using heavier models are its computational cost, which might make them impractical in implementations where speed of detection is of essence, which is true for our specific application. The lighter nano models were comparable in terms to number of human and forklift detection, but, they lack behind in accurate pose estimation compared to the heavier models. Even then, their speed of detection is a big advantage. For exanple, the lightest Nano model detected Human and Forklift poses 30-43% faster than the heavier models, which make them useful for certain applications. They also have another advantge that it is easier to deply on Jetson devices as they are lighter. While increasing the dataset size is one path to improvement, future research may also explore alternative neural network architectures for further performance enhancement. Moreover this work emphasized that Synthetic data helped in improving detection and pose estimation quality, especially in detecting a variety of poses. Subsequent investigations should focus on extending the dataset, both real and synthetic. Other synthetic data generation processes like NVIDIA Omniverse and Unity can be experimented with.

References

[1] Jungmo Ahn, JaeYeon Park, Sung Sik Lee, Kyu-Hyuk Lee, Heesung Do, and JeongGil Ko. Safefac: Video-based smart safety monitoring for preventing industrial work accidents. *Expert Systems with Applications*, 215:119397, 2023.

[2] Mary B Alatise and Gerhard P Hancke. Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter. *Sensors*, 17(10):2164, 2017.

[3] Yali Amit, Pedro Felzenszwalb, and Ross Girshick. Object detection. In *Computer vision: A reference guide*, pages 875–883. Springer, 2021.

[4] Blaxtair. Blaxtair: Proximity warning system for pedestrians, 2025. Accessed: 2025-04-04.

[5] Hannah M Boland, Morgan I Burgett, Aaron J Etienne, and Robert M Stwalley III. An overview of can-bus development, utilization, and future potential in serial network messaging for off-road mobile equipment. *Technology in Agriculture*, 2021.

[6] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th international conference on computer vision*, pages 1365–1372. IEEE, 2009.

[7] Sing T Bow. *Pattern recognition and image preprocessing*. CRC press, 2002.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[10] Ju Yong Chang and Kyoung Mu Lee. 2d–3d pose consistency-based conditional random fields for 3d human pose estimation. *Computer Vision and Image Understanding*, 169:52–61, 2018.

[11] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 1212–1221, 2017.

[12] Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Computer vision and image understanding*, 192:102897, 2020.

[13] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1831–1840, 2017.

[14] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Guido Ranzuglia, et al. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, volume 2008, pages 129–136. Salerno, 2008.

[15] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[16] Paolo Dabove, Vincenzo Di Pietra, Marco Piras, Ansar Abdul Jabbar, and Syed Ali Kazim. Indoor positioning using ultra-wide band (uwb) technologies: Positioning accuracies and sensors' performances. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 175–184. IEEE, 2018.

[17] Thiago D'Angelo, Marina Mendes, Breno Keller, Rafael Ferreira, Saul Delabrida, Ricardo Rabelo, Hector Azpurua, and Andrea Bianchi. Deep learning-based object detection for digital inspection in the mining industry. In *2019 18th ieee international conference on machine learning and applications (ICMLA)*, pages 633–640. IEEE, 2019.

[18] German Social Accident Insurance (DGUV). Forkift accidents, 2024. `https://www.dguv.de/en/facts-figures/index.jsp`.

[19] Alexandre Dolgui and Jean-Marie Proth. *Supply chain engineering: useful methods and techniques*, volume 539. Springer, 2010.

[20] Chengang Dong and Guodong Du. An enhanced real-time human pose estimation method based on modified yolov8 framework. *Scientific Reports*, 14(1):8012, 2024.

[21] Lixuan Du, Rongyu Zhang, and Xiaotian Wang. Overview of two-stage object detection algorithms. In *Journal of Physics: Conference Series*, volume 1544, page 012033. IOP Publishing, 2020.

[22] Songlin Du, Zhiwen Zhang, and Takeshi Ikenaga. Anatpose: Bidirectionally learning anatomy-aware heatmaps for human pose estimation. *Pattern Recognition*, 155:110654, 2024.

[23] Wael M Elmessery, Joaquín Gutiérrez, Gomaa G Abd El-Wahhab, Ibrahim A Elkhaiat, Ibrahim S El-Soaly, Sadeq K Alhag, Laila A Al-Shuraym, Mohamed A Akela, Farahat S Moghanm, and Mohamed F Abdelshafie. Yolo-based model for automatic detection of broiler pathological phenomena through visual and thermal images in intensive poultry houses. *Agriculture*, 13(8):1527, 2023.

[24] Xin Feng, Youni Jiang, Xuejiao Yang, Ming Du, and Xin Li. Computer vision algorithms and hardware implementations: A survey. *Integration*, 69:309–320, 2019.

[25] Mariano Focaraccio. Accidents with forklifts: how to avoid them with technology, minimizing economic losses., 2025.

[26] Blender Foundation. Blender: 3d creation suite. `https://www.blender.org`, 2021. Accessed: 2025-04-12.

[27] Fabrizio Gabbiani and Jens Midtgaard. Neural information processing. *Encyclopedia of Life Sciences (Nature Publishing Group)*, page 112, 2001.

[28] Gionatan Gallo, Francesco Di Rienzo, Federico Garzelli, Pietro Ducange, and Carlo Vallati. A smart system for personal protective equipment detection in industrial environments based on deep learning at the edge. *IEEE Access*, 10:110862–110878, 2022.

[29] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[30] GeeksforGeeks. Fast r-cnn: Faster, better, and more accurate object detection, 2021. Accessed: 2025-04-11.

[31] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[32] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[33] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson International, 4 edition, 2017.

[34] Tanmay Gupta, Ryan Marten, Aniruddha Kembhavi, and Derek Hoiem. Grit: General robust image task benchmark, 2022.

[35] Bastian Hartmann. *Human worker activity recognition in industrial environments*. KIT Scientific Publishing, 2014.

[36] Poyraz Umut Hatipoglu, Ali Ufuk Yaman, and Okan Ulusoy. Overhead object projector: Overprojnet. *Intelligent Systems with Applications*, 20:200269, 2023.

[37] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2888–2897, 2019.

[38] Sabera Hoque, Md Yasir Arafat, Shuxiang Xu, Ananda Maiti, and Yuchen Wei. A comprehensive review on 3d object detection and 6d pose estimation with deep learning. *IEEE Access*, 9:143746–143770, 2021.

[39] Tim Horberry, Tore J Larsson, Ian Johnston, and John Lambert. Forklift safety, traffic engineering and intelligent transport systems: a case study. *Applied ergonomics*, 35(6):575–581, 2004.

[40] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 34–50. Springer, 2016.

[41] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11977–11986, 2019.

[42] Iveta Kubasakova, Jaroslava Kubanova, Dominik Benco, and Dominika Kadlecová. Implementation of automated guided vehicles for the automation of selected processes and elimination of collisions between handling equipment and humans in the warehouse. *Sensors*, 24(3):1029, 2024.

[43] Gaurav Kumar and Pradeep Kumar Bhatia. A detailed review of feature extraction in image processing systems. In *2014 Fourth international conference on advanced computing & communication technologies*, pages 5–12. IEEE, 2014.

[44] Jieun Lee, Tae-yong Kim, Seunghyo Beak, Yeeun Moon, and Jongpil Jeong. Real-time pose estimation based on resnet-50 for rapid safety prevention and accident detection for field workers. *Electronics*, 12(16):3513, 2023.

[45] Katherine Leon, Domingo Mery, Franco Pedreschi, and Jorge Leon. Color measurement in lâ aâ bâ units from rgb digital images. *Food research international*, 39(10):1084–1091, 2006.

[46] Jianchu Lin, Shuang Li, Hong Qin, Hongchang Wang, Ning Cui, Qian Jiang, Haifang Jian, and Gongming Wang. Overview of 3d human pose estimation. *CMES-Computer Modeling in Engineering & Sciences*, 134(3), 2023.

[47] Linde Material Handling. Linde bluespot, 2025. Accessed: 2025-04-09.

[48] Linde Material Handling. Truckspot – ein innovatives warnsystem fÃ¼r mehr sicherheit, 2025. Accessed: 2025-04-04.

[49] Yu Han Liu. Feature extraction and image recognition with convolutional neural networks. In *Journal of Physics: Conference Series*, volume 1087, page 062032. IOP Publishing, 2018.

[50] Qianmai Luo, Chengshuang Sun, Ying Li, Zhenqiang Qi, and Guozong Zhang. Applications of digital twin technology in construction safety risk management: a literature review. *Engineering, construction and architectural management*, 2024.

[51] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2637–2646, 2022.

[52] MakeHuman Team. Makehuman project source repository. `https://github.com/makehumancommunity/makehuman`, 2025. Accessed April 13, 2025.

[53] Andrea Motroni, Alice Buffi, and Paolo Nepa. Forklift tracking: Industry 4.0 implementation in large-scale warehouses through uwb sensor fusion. *Applied Sciences*, 11(22):10607, 2021.

[54] Christopher Neff, Aneri Sheth, Steven Furgurson, and Hamed Tabkhi. Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation. *arXiv preprint arXiv:2007.08090*, 2020.

[55] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, 30, 2017.

[56] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016.

[57] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6951–6960, 2019.

[58] OECD. Catalogue of tools & metrics for trustworthy ai, 2023. Accessed: 2025-04-15.

[59] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European conference on computer vision (ECCV)*, pages 269–286, 2018.

[60] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In *European conference on computer vision*, pages 241–254. Springer, 2010.

[61] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[62] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Sebastian Thrun. Real-time identification and localization of body parts from depth images. In *2010 IEEE International Conference on Robotics and Automation*, pages 3108–3113. IEEE, 2010.

[63] Niranjan Ravi, Sami Naqvi, and Mohamed El-Sharkawy. Biou: An improved bounding box regression for object detection. *Journal of Low Power Electronics and Applications*, 12(4):51, 2022.

[64] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[65] Stephen M Rutner and C John Langley. Logistics value: definition, process and measurement. *The International Journal of Logistics Management*, 11(2):73–82, 2000.

[66] Jan Schuhmacher, Wjatscheslav Baumung, and Vera Hummel. An intelligent bin system for decentrally controlled intralogistic systems in context of industrie 4.0. *Procedia Manufacturing*, 9:135–142, 2017.

[67] Mubarak Shah. Fundamentals of computer vision. *Orlando: University of Central Florida*, 1997.

[68] Vivek S Sharma, Shubham Mahajan, Anand Nayyar, and Amit Kant Pandit. *Deep Learning in Engineering, Energy and Finance: Principals and Applications*. CRC Press, 2024.

[69] Ming-Hwa Sheu, SM Salahuddin Morsalin, Chung-Chian Hsu, Shin-Chi Lai, Szu-Hong Wang, and Chuan-Yu Chang. Improvement of human pose estimation and processing with the intensive feature consistency network. *IEEE Access*, 11:28045–28059, 2023.

[70] Mupparaju Sohan, Thotakura Sai Ram, and Ch Venkata Rami Reddy. A review on yolov8 and its advancements. In *International Conference on Data Intelligence and Cognitive Informatics*, pages 529–545. Springer, 2024.

[71] Milan Sonka, Vaclav Hlavac, Roger Boyle, Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image pre-processing. *Image processing, analysis and machine vision*, pages 56–111, 1993.

[72] Andreas Specker, Daniel Stadler, Lucas Florin, and Jurgen Beyerer. An occlusion-aware multi-target multi-camera tracking system. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4173–4182, 2021.

[73] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[74] Sebastian Thiede, Brendan Sullivan, Roy Damgrave, and Eric Lutters. Real-time locating systems (rtls) in future factories: technology review, morphology and application potentials. *Procedia CIRP*, 104:671–676, 2021.

[75] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He FCOS. Fully convolutional one-stage object detection. in 2019 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019.

[76] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[77] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[78] Christopher Ull, Hans Ehlers, Emre Yilmaz, Sebastian Lotzien, Thomas A Schildhauer, Charlotte Reinke, and Christiane Kruppa. Injuries after forklift trucks accidents–injury patterns, therapy and outcome in the context of the statutory accident insurance. *Zeitschrift für Orthopädie und Unfallchirurgie*, 160(05):539–548, 2022.

[79] Viatech. Mobile360 d700 ai dash cam, 2025. Accessed: 2025-04-04.

[80] Wei-Chih Wang, Jang-Jeng Liu, and Shih-Chieh Chou. Simulation-based safety evaluation model integrated with network schedule. *Automation in construction*, 15(3):341–354, 2006.

[81] Nivedita Wani and Shailesh Bendale. 3d human motion prediction based deep learning. *International Journal of Advanced Research in Science, Communication and Technology*, pages 543–553, 11 2022.

[82] Zhennan Yan, Yiqiang Zhan, Zhigang Peng, Shu Liao, Yoshihisa Shinagawa, Shaoting Zhang, Dimitris N Metaxas, and Xiang Sean Zhou. Multi-instance deep learning: Discover discriminative local anatomies for bodypart recognition. *IEEE transactions on medical imaging*, 35(5):1332–1343, 2016.

[83] Ning Yang, De-Feng Liu, Tao Liu, Tianyuan Han, Pingyue Zhang, Xuenan Xu, Siyu Lou, Huan-Guang Liu, An-Chao Yang, Cheng Dong, et al. Automatic detection pipeline for accessing the motor severity of parkinsonâs disease in finger tapping and postural stability. *IEEE Access*, 10:66961–66973, 2022.

[84] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *proceedings of the IEEE international conference on computer vision*, pages 1281–1290, 2017.

[85] Jincao Yao, Zhikai Lei, Wenwen Yue, Bojian Feng, Wei Li, Di Ou, Na Feng, Yidan Lu, Jing Xu, Wencong Chen, et al. Deepthy-net: a multimodal deep learning method for predicting cervical lymph node metastasis in papillary thyroid cancer. *Advanced Intelligent Systems*, 4(10):2200100, 2022.

[86] Zitong Yu, Yunxiao Qin, Xiaobai Li, Chenxu Zhao, Zhen Lei, and Guoying Zhao. Deep learning for face anti-spoofing: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5609–5631, 2022.

[87] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022.

[88] Zheng Zhang, Juan Chen, and Qing Guo. Application of automated guided vehicles in smart automated warehouse systems: A survey. *CMES-Computer Modeling in Engineering & Sciences*, 134(3), 2023.

[89] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE transactions on pattern analysis and machine intelligence*, 26(7):892–899, 2004.

[90] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[91] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023.

[92] ZoneSafe. Proximity warning systems: Pedestrian safety, 2025. Accessed: 2025-04-04.