NTNU
Kunnskap for en bedre verden

Department of Mechanical and
Industrial Engineering

Master's Thesis - robotics & automation

Enhancing Real-World Applicability of On-Device Object Detection

Drafts:
The Impact of Dataset Characteristics on the Real-World Accuracy of Privacy-Preserving Human Detection Systems
Enhancing Real-World Applicability of On-Device Object Detection
On-Device Object Detection and Dataset Authenticity: Enhancing Privacy and Real-World Applicability
Exploring the Sensitivity of Object Detection Models to Real-World Dataset Variations
Old: Efficient, accurate, and privacy-preservant object detection in edge devices

Student:
Hallvard Enger Bjørgen

Supervisor at NTNU:
Amund Skavhaug

Trondheim/Esbjerg Spring 2024

Table of Contents

List of Figures

List of Tables

Abstract

Placeholders:
What
Why
How
Principal contributions
Principal conclusion

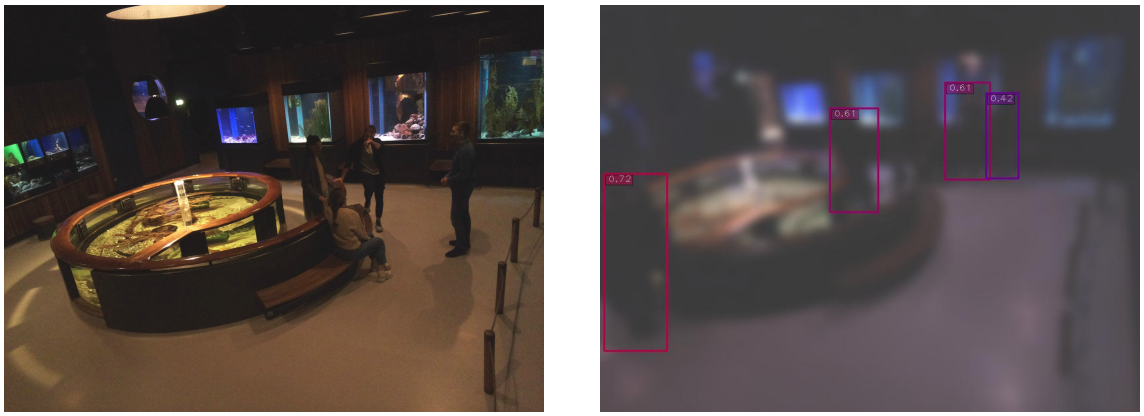
Sammendrag

Plassholdere: *Hva Hvorfor Hvordan Prinsipielle bidrag/resultater Prinsipiell konklusjon*

1 Introduction

Machine Learning (ML) is the area of artificial intelligence (AI) which involves the development of models that are trained on data to perform specific tasks. The last years have shown great improvements in model's capabilities in extracting useful information from images. Societal regulations for the technology and consumer opinions with regards to privacy and data security is struggling to keep up with the rapid increase in complexity and capabilities of the developed and deployed machine learning models.

On-device processing is emerging as a vital component of modern human detection and tracking systems, particularly as a strategy to enhance privacy and data security. The ability to detect and track humans in real-time is essential across a range of applications, from security surveillance to visitor analytics in cultural institutions. However, the deployment of these systems, especially in sensitive environments like museums and aquariums, raises significant privacy and data security concerns. This thesis explores the development and deployment of a privacy-preserving human localization system, specifically addressing the challenges posed by on-device processing where images are deleted or obfuscated post-inference. This process complicates the validation of inference accuracy, especially since many models trained on large, generic datasets may not perform equivalently in specific deployment scenarios. A dataset was collected, and analysis on live data was performed (see Figure 1).



(a) Example Image From the FIMUS Dataset.

(b) Example Image From the Live Detections.

Figure 1: Example Images From the FIMUS Aquarium.

todo bytt ut med et b med de rette innstillingene?

1.1 Problem Description

The method of human detection and tracking in public spaces has significantly evolved over the past decade, driven by advancements in computer vision and machine learning. Traditional surveillance systems typically relied on centralized processing, where video feeds were transmitted to a remote server for analysis. This approach not only raised privacy concerns due to the potential exposure of sensitive information but also required substantial human intervention, making it time-consuming, error-prone, and lacking in scalability. This thesis advocates for a shift towards *on-device* processing, which performs analytics locally on the edge device, thereby eliminating the need to transmit raw video data and significantly enhancing privacy. This is particularly relevant for environments such as museums and aquariums where privacy preservation is critical.

Additionally, the performance of object detection models is often evaluated using generic datasets like COCO, which may not accurately reflect the conditions of the deployment environment. This discrepancy can lead to suboptimal model performance in real-world scenarios, where factors like lighting conditions, camera angles, and object occlusion can significantly impact detection accuracy.

This thesis aims to investigate the impact of using deployment-specific data for the fine-tuning of object detection models, and on the evaluation of object detection models, comparing the performance metrics obtained with those from generic datasets.

1.2 Scope

To demonstrate the feasibility and effectiveness of on-device human detection and localization in a practical setting, two devices were deployed in the "Fiskeri og Søfartsmuseet" (FIMUS) aquarium in Esbjerg, Denmark. The deployment aimed to address the unique challenges of indoor, low-light environments. A dataset of 3397 images was collected and labeled, and was used to evaluate and fine-tune several object detector machine learning models. An example image from the aquarium is displayed in Figure 1. The best performing model was subsequently deployed to collect anonymous data on visitors over a month, with results visualized through heatmaps and analysis of peak visitation hours.

The scope of this project is dualistic. It encompasses demonstrating a comprehensive implementation of a privacy-preserving human localization system. It also encompasses critically assessing the validity of object detection model performances across general and specific datasets to understand the real-world impacts of scientific advancements.

The focus of this thesis is selectively deep on topics such as privacy, privacy preservation in images, and performance metrics of object detectors. These areas are emphasized due to their critical relevance and the necessity of a fundamental understanding of these topics to grasp the project's core objectives. While other subjects like object detector model architecture, edge devices, challenges in object detection, security, and decision support systems also present interesting avenues of exploration, they are not central to the thesis' primary aims.

This thesis is intended for a diverse readership, including edge-device engineers, AI and machine learning enthusiasts, non-technical social studies scholars, and policymakers engaged in crafting regulations for object detection technologies. This results in a broad scope of topics covered in the thesis, ranging from technical discussions on machine learning models to ethical considerations in deploying object detection systems in public spaces.

The project of this thesis also spanned several disciplines. It required research, development, and effort in edge-device deployment, machine learning, and data science. Choices were made to focus the scope to manage the workload effectively.

1.3 Limitations

Secure Control of Device A dataset was built of consenting individuals in an aquarium which was part of a larger museum facility. However, once development was finished and the system was tested, the devices were actively photographing individuals who had *not* given consent to be photographed. Privacy was still preserved by immediately inferencing on and deleting the images. In such an application, it is imperative to not store or upload clear, privacy-intrusive images. Therefore, an existing and already proven secure solution developed by *HallMonitor*, a company specializing in on-device processing solutions based in Esbjerg, was utilized to establish a secure communication channel with the deployed devices. The communication channel was used to extract the analytics data from the devices. This secure system setup, necessary to protect the devices from attackers, is not covered in this thesis due to its proprietary nature.

Legal Considerations The discussions and insights in this thesis may apply to global applications, but the legal considerations specifically target the European Union member countries. Further, some of the discussions may be influenced and biased by a heavily european-influenced cultural mindset and thus not be as relevant and applicable to parts outside Europe. More detailed discussions on international privacy laws beyond just the GDPR should have been included if the technology was intended for global application.

Fine-tuned Model Development The project's broad scope resulted in a limited exploration of potential improvements in model fine-tuning. This thesis evaluates the performance of various machine learning models, including models built from the three arhitectures YOLOv3, YOLOv9, and DETR. Two more object detection architectures are also mentioned, but were not (fully) implemented. These are Co-DETR, the current best-performing model on the COCO dataset, and the Faster-RCNN, another popular and good option for object detection. However, the Co-DETR was deemed too complex and resource-intensive for the project's scope to be fully implemented and evaluated, and Faster-RCNN was not prioritized due worse performance than the YOLOv9. The object detectors are discussed in Section 2.5.

Museum and Aquarium Opening Hours and Visitor Conduct The project was designed to avoid interference with the normal operations of the aquarium. Consequently, image capture for the dataset was confined to aquarium opening hours, and random visitors were not inquired whether they'd be willing to participate in the project. An early analysis of the visitation patterns revealed the aquarium was busy from the opening at 10:00 until approximately to 15:00, 2 hours before closing time. This meant most images for the dataset were captured in the two hours before closing time where there was least traffic.

The Task is Object Detection There are several tasks within the domain of computer vision, each serving distinct purposes and complexities. This project focuses exclusively on simple object detection, which involves locating objects of relevance within an image. Specifically, this thesis addresses single-class object detection with *person* as the sole class of interest. Other tasks in computer vision include person re-identification, image classification, combined image classification and localization, semantic segmentation, and instance segmentation. Re-identification involves recognizing individuals across different images and image classification is the task of classifying the image contents as a whole. The rest of the tasks are illustrated in Figure 2 to display how they differentiate from object detection.

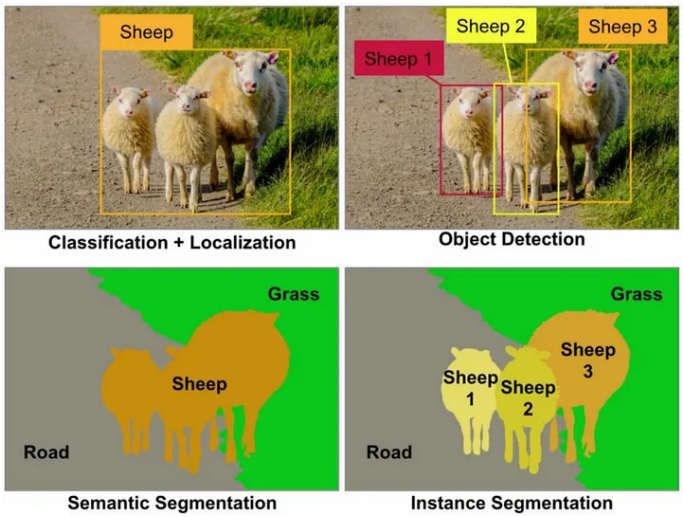


Figure 2: Image processing tasks (Murali, 2021).

The selection of a dataset must be directly aligned with the specific task to be performed, as it must contain data suited for that task. For instance, applications tasked with person reidentification require a dataset that includes the identities across images for the persons depicted in the images. An appropriate dataset for such applications, the Person Reidentification in the Wild (PRW), is detailed in Section 3.3.3.

Lighting Conditions in Aquarium Settings The environmental conditions within aquarium settings impose specific lighting requirements that significantly influence the deployment and

performance of the proposed object detection system. Aquariums typically maintain low-light conditions, presumably to minimize stress for the aquatic animals and improve vision into the tanks for the visitors. In this context, the options for enhancing illumination are often limited, making it impractical to simply increase lighting to facilitate better visual detection technologies. These subdued lighting conditions were an important feature of the thesis project.

TinyML and Frugal Devices An initial attempt was made to encompass tinyML and frugal devices in the project. TinyML is when machine learning models are aimed at deployment to heavily resource-constrained environments, e.g. "frugal devices". These are devices where the microcontroller units (MCUs) are accompanied by memory measured in kilobytes, and processor speeds measured in megahertz. Machine learning networks applied to tiny robots are subject to challenges from size, weight, area, and power (SWAP) (Neuman et al., 2022). Many of the same challenges apply even in applications where the SWAP challenges are not the main concerns. Rajapakse et al. mentions the open challenges and future directions of the next generation tinyML. Catastrophic forgetting, which is when information from previous tasks while learning new ones are forgotten, are a result of the frugal devices' computational resources and memory[1]. The first recommendation for future directions from the authors is to investigate fog computing as a means to offload tasks from the frugal devices. Exploring a system of frugal devices for obtaining sensor data on the edge and "fog devices"[2] for processing of the data would be a time-consuming objective, especially in a real-world scenario where security must be taken seriously. Therefore, tinyML and frugal devices are not elaborated in detail in this thesis, but discussed briefly in D.

Model Hyperparameter Tuning The project did not encompass hyperparameter tuning of the models. Hyperparameter tuning is the process of finding the best hyperparameters for a machine learning model. Hyperparameters are the settings of the model that are not learned from the data, such as the learning rate, batch size, and number of epochs. Hyperparameter tuning is a crucial step in the machine learning pipeline, as it can significantly impact the model's performance. However, due to time and resource constraints, hyperparameter tuning was not included in this project. Further comments on hyperparameter tuning are made in Section 3.4.1.

1.4 Research Questions

Rather than forming a single hypothesis for this rather widespread project, a set of research questions are formulated to guide the research and to provide a structured approach to the investigation.

The Research Questions are as follows:

1. What are some privacy risks associated with traditional human localization systems in public spaces, and how may a system mitigate these privacy concerns?
2. How does the validity of object detection model evaluations change when using data specifically from the intended deployment environment compared to using generic datasets?
3. What are some machine learning architectures suitable for object detection in a real-world deployment scenario?
4. How many images are needed for testing before the average precision of a machine learning model converges in a single-environment setting?

1.5 Research Objectives

The Primary Objectives are to:

[1] Catastrophic forgetting can also be seen in transfer learning, and is why freezing the backbone of a pre-trained model may be a good idea.

[2] Fog devices is an unofficial term for processing-instances located close to the edge.

1. Develop a privacy-preserving human localization system using on-device processing to minimize data transmission and enhance data privacy.
2. Demonstrate feasability and effectiveness of on-device human detection and tracking in a practical and realistic setting.
3. Investigate the effects of the dataset quality in fine-tuning of models on the performance.
4. Assess the impact of using real world-specific test data on the evaluations of object detection models, by comparing model evaluations obtained with specific test data to evaluations obtained with more general datasets.

The Secondary Objectives are to:

1. Compare the privacy and performance impacts of on-device processing against traditional centralized processing methods.
2. Investigate the feasibility of deploying the developed system in other public spaces to enhance visitor analytics.
3. Demonstate how to visualize object detection data by creating visualizations of collected data from a realistic setting.
4. Explore relevant object detection architectures to evaluate their performance in a real-world deployment sceneario.

1.6 Project Work

In order to achieve these objectives, the following project work was executed and is detailed in this thesis:

- Designed and implemented a prototype system that incorporates on-device processing for privacy-preserving human localization.
- Deployed a prototype in an aquarium to gather real-world data and analyze system performance.
- Collected and labeled a dataset of 3394 images to evaluate and fine-tune several object detector machine learning models.
- Conducted comparative studies to evaluate the effects of data specificity and quality on model accuracy.
- Performeded a comparative analysis of three object detection architectures under real-world conditions.
- Developed and utilized advanced visualization tools to represent data findings in a way that facilitates clear understanding and decision-making.

This thesis compiles and structures literature on various aspects of human localization systems, bridging multiple domains. It contributes to the field by serving as a collaborative and communicative tool, fostering common understanding among technologists, ethicists, policymakers, and the public.

1.7 Structure

The thesis is structured the following way. The descriptions include what each section will cover and why, to increase readability and coherence of the thesis.

Section 2: Literature Review The theoretical underpinnings of the project are surveyed and discussed in the literature review. This section contains text from previous work by the same authors (see 1.8.1). Privacy is a recurring subject in the literature review and throughout the rest of the thesis due to it's upmost importance in projects tasked with detecting and tracking individuals.

Section 2.1 is included to investigate the need for a human localization system and to provide a short overview of what has already been tested. Museum stakeholders opinions and IoT consumer questionnaire participants are included in this section. Section 2.2 provides a rudimentary overview of current regulations in the EU with regards to privacy, and aims to summarize the regulations most relevant for object detection of individuals. Section 2.3 provides an in-depth overview of some of the approaches and methods to preserve privacy specifically in images. Section 2.4 presents various opinions and ethical considerations related to human localization technologies. These perspectives come from a previous developer in the field of object detection, an acclaimed writer, and multiple philosophers. Section 2.5 provides a technological overview of object detector datasets, performance metrics, object detector algorithms, dark-lit environment considerations, and an overview of transfer learning and the effectiveness of fine-tuning. This section was included as an obvious necessity for this thesis. The detector algorithms are only briefly discussed as much in-depth literature is available elsewhere. A subsection of the key differences and similarities betweem the algorithms used in the project are included. Section 2.6 and Section 2.7 brings forth various third party software services and products. These are included due to their similarity and overlapping use cases with the project of this thesis and to improve awareness around potential already-existing implementations that may fit a given use case.

Section 3: Methodology The Methodology section outlines the detailed technical approaches and methods employed in this project. This section is critical for understanding how the data was collected, processed, and analyzed. It provides a foundation for interpreting the results and findings of this thesis, and a common understanding of the terms.

Section 3.1 provides an overview of the project, including the deployment of devices in the aquarium and the collection of datasets. This section explains the different machine learning models employed for training and evaluation. Section 3.2 delves into the construction of the FIMUS dataset. It covers the camera configurations, image capturing process, and labeling. This section highlights the differences between the inconsistent and consistent partitions of the dataset, detailing the specific settings and processes used to ensure the quality and relevance of the captured images. Section 3.3 briefly describes the external datasets used in the project; COCO, CrowdHuman, Person Reidentification in the Wild, and Football-players. It explains the relevance of each dataset to the project and how they were utilized for model training and evaluation.

Section 3.4 focuses on the training process of the models. The section contains information regarding the licenses of the models, and the hyperparameter tuning process (which was mostly foregone), and the use of Google Colab for cloud training with GPUs, detailing a few pros and cons. Finally, it provides an explanation regarding the lack of validation used in the training of the models, a decision based both on the lack of hyperparameter optimization in the project and issues with google colab service disconnections.

Section 3.6 restates the metrics used to evaluate the performance of the models in this thesis project, and defines them as *COCO AP* and *Vary-Both AP*. Section 3.5 presents the different models that were developed and tested in the project. It includes details about the pre-trained and fine-tuned models, their configurations, and the specific experiments conducted to measure their performance.

Section 2.4 addresses the ethical methodologies that were implemented in the project to ensure preservation of privacy. Section 3.8 discusses the use of heatmaps as a visualization tool to analyze visitor behavior patterns. It details the attempts to create heatmaps using different Python packages and the final implementation using the Supervision module by Roboflow[3].

By following this structure, the Methodology section provides a comprehensive and transparent

[3]A final heatmap is displayed in the end of this section, and in the results (Section 4.3)

account of the technical processes involved in the project, ensuring that the results are reproducible and the conclusions are drawn based on a solid and well-documented foundation.

Section 4: Project Results Presents the data collected, evaluates the system's performance, and discusses the findings.

Section 6: Conclusions and Future Work Summarizes the research contributions and outlines potential future research directions.

Appendices **Appendix A Code Snippets** serves as an easily accessible illustration of the camera configuration options ordering used to capture the images for the dataset. The specific values are displayed in Table 2. Most importantly, this appendix serves as an illustration of what order to set the camera settings to obtain consistent images. The most important takeaway is to set the shutter speed first, then let the automatic gain control settle for about 2 seconds (or until it reaches a certain threshold value), then set the exposure- and auto white balance modes to 'off', before manually setting the auto white balance gains. For the matter of reproducibility, the todo upload master thesis code Github should be used instead.

Appendix B Technical Challenges is included to mention some of the challenges faced during the project implementation which there was no room for elsewhere in the thesis. The list of challenges is far from exhaustive, but is included in case the reader faces similar problems in their implementation of similar systems. The appendix aims to build on the first primary objective of developing a privacy-preserving human localization system by providing a list of possible challenges one may face in the attempt of implementing said system.

Appendix C Camera Settings Explanation provides an easy-access way of understanding the settings that were used in this project.

Appendix D TinyML and Frugal Devices is a section of 4 paragraphs which landed outside the scope of the project, but is included in the thesis as it is considered a highly relevant topic for adjacent systems implementing less complex hardware than the device used in this project (the *Raspberry PI 4*).

1.7.1 APA-Format

In accordance with APA-style,

- Every quotation of over 40 words has been formatted as a free-standing block without quitation mark. These blocks are indented (Academic Resource Center, University of California Riverside, 2024).
- Quotations of fewer than 40 words are included in the text and surrounded with quotation marks. The page number has not been included as it is considered irrelevant with todays digital tools as one may search for the exact sentence instead and find it much faster.
- The sections and subsections within implement title case capitalization in accordance with APA-style (American Psychological Association, 2022).

1.8 Disclaimers

1.8.1 Reuse of Previous Work by the same Author

Some of the subsections in the Literature section are heavily based on an unpublished[4] preliminary study for the project of this thesis by the same authors. This preliminary study was written the semester before, and was a theoretical study of how one may achieve "Efficient, accurate, and privacy-preservant object detection in edge devices" (which was it's title). This subsections most influenced by the previous work are the subsections 2.3, 2.5, and 2.7.

1.8.2 The Use of AI Tools

Some of the code and text in this thesis has been enhanced by the use of AI tools. For the main portions of the writing process, Github Co-Pilot was considered mostly distracting and therefore disabled. In Figure 3 we see an example of one of these distracting suggestions, where it may seem the AI thinks very strongly we need more reasearch on who are not wearing masks and are not vaccinated... The somewhat humurous suggestion served only as a distraction to the writing of this thesis.

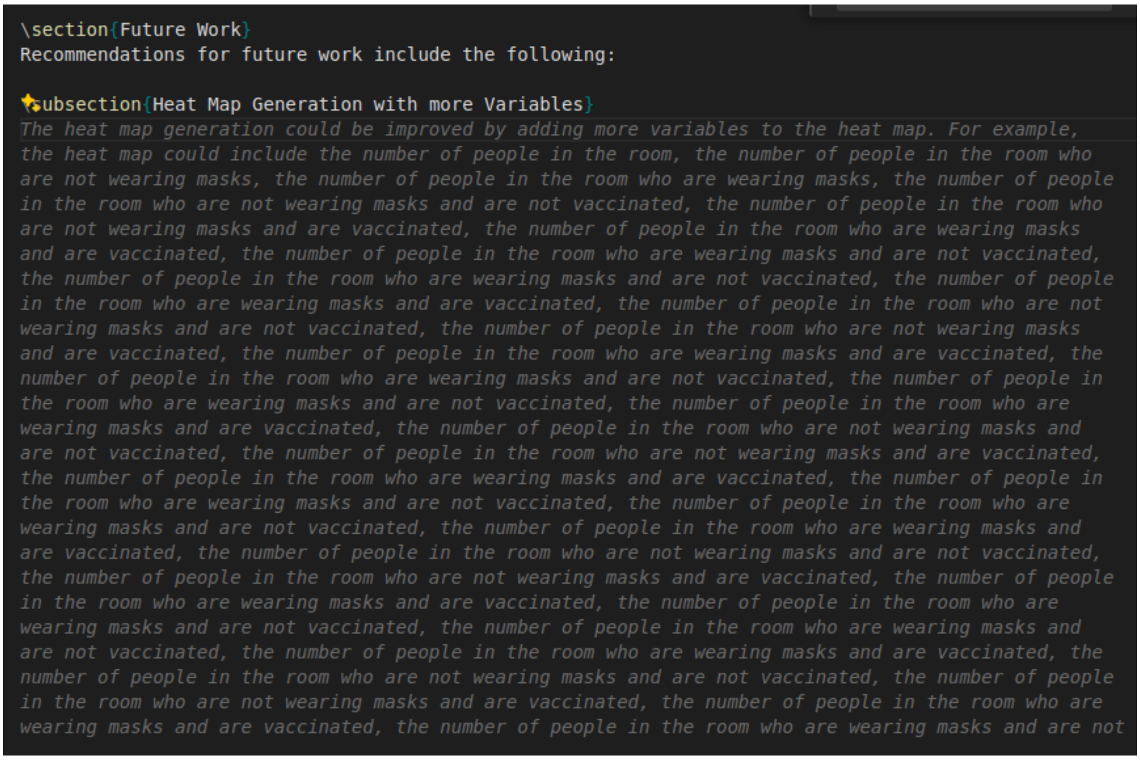


Figure 3: The Co-Pilot is mostly annoying for Latex.

However, some boiler-plate Latex-code and some of the sections have been sent to OpenAI's ChatGPT-4/ChatGPT-4o to verify it's quality and to get suggestions on how to enhance readability and flow. The author has tried to be transparent about the use of these tools, and has tried to ensure that the text is original and not plagiarized.

[4]The preliminary study is not publicly available, as the default practice of the educational institution is to not publish these.

1.8.3 Privacy of Similar Projects

The author of this thesis is not an expert in privacy. The methods outlined in this thesis are meant to ensure privacy of individuals, but the author cannot guarantee that the methods are foolproof. The author has tried to follow best practices and guidelines from the field and has tried to be transparent about the methods used and the limitations of the methods, but the reader should be aware that following the methods outlined in this thesis may not necessarily be enough to ensure privacy. An investigation into the privacy of similar projects is recommended before deploying a similar system in a real-world setting.

2 Literature Review

The advent of modern object detection has enabled more sophisticated and automated methods for understanding visitor engagement and flow in cultural institutions. This literature review aims to explore the current state of research on the various topics of this thesis. This includes visitor behaviour analysis, individual privacy, object detection, the influence of fine-tuning models on specialized data, and existing third-party services for deploying similar human localization systems.

2.1 Visitor Behaviour Analysis

Some of the traditional methods of analyzing visitor behaviour include surveys, manual counting, and direct observation. Today, more technology-driven and practical applications may be used to gain insights in visitor engagement and experience in a museum or acquarium setting. In this subsection, we look at an alternative to computer vision tracking systems. Afterwards, a study on the percieved value of visitor tracking to museum stakeholders is brought foreward.

2.1.1 Questions asked by visitors in a mobile app

Pérez Cortés et al. had visitors ask questions in a mobile app while moving through the museum 2023. Visitor movement through the museum was inferred from the data by leveraging question keyword content, knowledge of exhibit layout, and question timestamps. This removed the need for more costly, vision-based applications for detecting and tracking visitor movement. This study illustrates one way of conducting affordable, dependable and scalable visitor analysis without the need for costly devices.

2.1.2 Percieved value to museum stakeholders

Lanir et al. explored an alternative approach to museum visitor behaviour analysis, and its perceived value to museum curators, administrators and department heads (2017). Wearable RFID trackers[5] were given to the visitors, and beacons were positioned at positions deemed important by the museum curators. The beacons would then communicate the positions of the visitors to the system. This allowed for the collection of data on key metrics like exhibit popularity, average visit duration, and common visitor paths. The authors noted that technology-based visitor behaviour analysis was generally well-received by museum curators, offering valuable data that could enhance the visitor experience.

The study of Lanir et al. further discussed the divergent views between the curators and the administrators on the utility of visitor behaviour analysis systems (2017). Administrators and department heads generally viewed these systems favorably, citing the financial justification for expensive exhibitions: *We really need to know if this expenditure was worthwhile* (Lanir et al., 2017). On the contrary, museum curators expressed skepticism. One curator remarked:

> A temporary exhibition won't change after you deploy it, and understanding how it is used by the public would not help me in my next exhibition, since they are each very different. My main reason to analyze behaviour would be to satisfy my curiosity. (Lanir et al., 2017)

This contrast underscores the varied perspectives within museums regarding the value and implications of behaviour analysis technologies.

[5]The requirement for visitors to wear RFID trackers represents a significant drawback as it may be perceived as intrusive (although completely privacy preservant).

2.1.3 User perceptions of smart home IoT privacy

In a 2018 study, researchers conducted semi-structured interviews of 11 smart home owners were conducted to figure out user perceptions of smart home IoT privacy (S. Zheng et al.). Another research question of the study (S. Zheng et al., 2018) was user perceptions of obsolescence of the IoT devices, as there are frequent upgrades and new products on the market. Responses regarding privacy and obsoleteness of IoT devices were the following:

> I think it's more likely that a lot of these things will become obsolete... If that's what happens then I have to buy another device. It still might be worth it for the convenience. (Participant 10)

> [The security concern] is always kind of in the back of my mind because of all that IoT stuff that always goes on, and everyone says how easily hackable they are. But I think my peace of mind that I get from having them outweighs my worry of what could be potentially taken advantage of. (Participant 6)

These responses indicate that the convenience and connectedness of the devices surpass the desire to preserve privacy. This is a promising finding for the development of visual systems in museums and acquariums, as it suggests that the benefits of the system may outweigh the privacy concerns of the visitors. It also illustrates the need for regulations to prevent solutions from being developed that are too privacy-invasive, as the end-users will not prioritize privacy over convenience.

2.2 The General Data Protection Regulation (GDPR)

The general data protection regulation (GDPR) is a single set of regulations to guarantee privacy and protecion of personal data. A quick review of the GDPR should be on the agenda of anyone affiliated with systems not inherently preservant of privacy[6].

The GDPR entered into applicability in the EU on 25th of May 2018 has two major impacts. 1) It leaves individuals with more control over their data, and 2) it facilitates a level playing field for all companies; there is now a single set of data protection rules for all companies operating in the European Economic Area (EEA)[7]. The most relevant sections of the GDPR to this thesis are the regulations regarding personal data.

2.2.1 Personal Data

Personal data is any form of information that can be connected to an identifiable data subject. The following definition was given by the european parliament in 2016:

Definition of personal data, as given by EUs GDPR:

> The term 'personal data' means any information relating to an identifed or identifable natural person ('data subject'); an identifable natural person is one who can be identifed, directly or indirectly, in particular by reference to an identifer such as a name, an identifcation number, location data, an online identifer or to one or more factors specifc to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person. (The European Parliament, 2016)

Approaches to Managing Personal Data Various methodologies can be adopted to manage personal data within a system.

One approach involves transforming information so that it no longer qualifies as personal data. This can be achieved through techniques such as differential privacy, which ensures that the processed data cannot be traced back to an individual. For a detailed explanation of differential privacy, refer to Section 2.3.3.

[6] More specifically, *informational* privacy. This term is introduced in 2.3
[7] The EEA consists of all EU countries plus Iceland, Liechtenstein and Norway.

A second approach for managing personal data involves establishing lawful grounds for the processing of personal data. This necessitates adherence to legal frameworks that justify the use of personal data under specified conditions, thereby ensuring compliance with data protection regulations. Processing of personal data is permissible under the GDPR only when it satisfies at least one of the following legal bases:

Legal Bases for Processing Personal Data

1. The data subject has given explicit consent.
2. It is necessary for the performance of a contract to which the data subject is a party.
3. It is necessary for compliance with a legal obligation to which the controller[8] is subject.
4. It is necessary to protect the vital interests of the data subject or of another natural person.
5. It is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller.
6. It is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data.

Additionally, the controller is responsible for compliance with the 3 requirements summarized below and should be able to demonstrate this compliance at any given time.

1. Security Documentation In the event of a breach of personal data, the controller must document that proper precautions were made to secure the data. One of these precautions is to deleted data that is no longer needed. This rule to delete no-longer-needed data is often overlooked and violated by companies (Sandtrø, 2022).
2. Data breaches Breaches of personal data must be reported within 72 hours. Companies failing to do so are economically sanctioned, but even worse, it damages the reputation of the company. In such cases it is common to uncover more failures (Sandtrø, 2022). This is often that the company has failed to make, or failed to document, the efforts they have made to sufficiently protect the data (the first requirement).
3. Rights of the data subject The data subject has the right to be informed about how their personal data is handled. This is commonly achieved through the company's privacy declaration, which must be comprehensive and regularly updated. Additionally, companies are encouraged to proactively communicate this information to clients, for instance, via email. According to privacy experts (Sandtrø, 2022), adopting such practices is an effective way of building and maintaining trust with customers.

There are multiple other approaches for managing personal data that are more specific to the management of *images*. The ones discussed in this thesis are primarily concerned with removing the individual information of persons from the images. These methods are discussed in Section 2.3.

2.2.2 The Network & Information Security 2 (NIS2) Directive

The NIS2 Directive (European Parliament and Council of the European Union, 2022) is a more recent EU regulation that came into force in January 2023. Unlike the GDPR, which broadly addresses the protection of personal data, NIS2 is specifically targeted toward technology. As an update to the EU's cybersecurity framework, NIS2 focuses on strengthening the security of network

[8]The controller refers to the party controlling the data

and information systems. It emphasizes the critical need for robust security measures in systems that process personal data to prevent unauthorized access and data leaks.

Both NIS2 and GDPR highlight the principle of data minimization, which mandates that object detection systems process only the necessary amount of personal data for their intended function. This practice not only bolsters security but also supports privacy by minimizing potential data exposure. Adhering to these principles is vital for maintaining user trust and ensuring compliance with EU regulations, particularly when deploying object detection technologies in environments where data sensitivity is paramount.

2.3 Preservation of Individual Privacy in Images

Building on the previously introduced regulations in Section 2.2, the contents of this section aims to provice a deeper insight into the methods of preserving the individual privacy of individuals *in images*. It is heavily influenced by previous work of the same author. See the disclaimers in Section 1.8 for more details.

The first definition of privacy was given by Brandeis and Warren in 1890 as *the right to be let alone*. A more comprehensive definition of privacy that is more relevant to the modern age of digitalization and the topics of this thesis is the following:

Privacy as informational self-determination:

> Privacy is the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others. (Westin, 1967)

There are multiple dimensions to privacy. The definition of Westin covers informational privacy, which is most relevant to this thesis. This definition includes groups and institutions, however, in most legal systems, privacy is defined as a basic human right that only applies to natural persons[9]. The term *individual privacy*[10] encapsulates the individual focus of privacy as opposed to the broader interpretations of privacy that might apply to groups, organizations, or institutions. Other dimensions of privacy include spatial privacy, territorial privacy, and bodily privacy (Fischer-Hbner and Berthold, 2017). These will not be further discuss in this thesis.

Preservation of individual privacy refers to maintaining the personal space and confidentiality of individuals, ensuring that their private lives and personal integrity are not invaded or exposed (without consent).

Protection of personal data is important due to the regulations, preserving the individual privacy is also essential to maintain trust with customers, and to avoid (*delay*[11] for brief discussion on the ethicality of person localization systems development.) the onset of a dystopian society...

2.3.1 Privacy in Images

Protection of personal data in general is very similar to protection of personal data in images. Protection deals with the management and security of personal information—data that can identify an individual, such as names, addresses, and biometrics. This protection is primarily about the correct handling, processing, storage, and destruction of personal data to prevent unauthorized access, misuse, or breaches.

There are multiple methods, both pre- and post-processing, for preserving individual privacy in images. One example of a pre-processing privacy preservation method is to hide the facial regions optically during capture, which was done in a study on fall detection by X. Wang et al. (X. Wang et al.).

[9] A natural person (also sometimes referred to as a physical person) is a title used to identify an individual human being. This is different from a legal person, which can be an individual or a company (Termly, 2023).

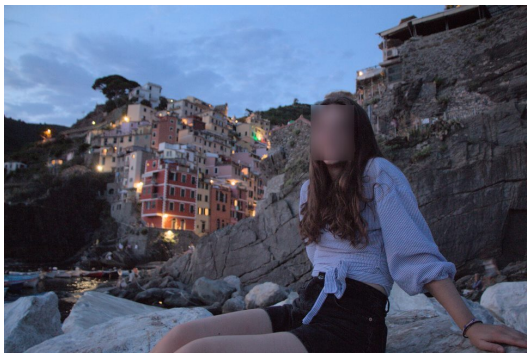
[10] Individual privacy is sometimes referred to as personal privacy.

[11] See Section 5.4

Post-processing methods include various techniques to obscure identifiable information after the data has been captured. These range from simple blurring and pixelation to more sophisticated approaches such as k-anonymity (Sweeney, 2002) and differential privacy. Six of the simple, easy-to-implement methods are shown in Figure 4, demonstrating practical implementations.

K-anonymity claimed to be a mathematically proven method for anonymyzation of personal data, but has been critizised by it's successor, the l-diversity criterion, for not being robust in the events where attackers have background data (Machanavajjhala et al., 2007). Differential privacy is a statistical disclosure control algorithm to process individual data from a group to produce close-to-real outcomes without disclosing the personal data of individuals (Y. Huang et al., 2023). Differential privacy is explained and illustrated in 2.3.3.

(a) Blurred entire image of Hong Kong street to protect privacy of citizens.



(b) Blurred face of individual by a sea town in Cinque Terre.

(c) Masked faces.

(d) Pixelated faces.

(e) Unconventional method: replace faces. May be done as effectively as the other approaches, but is likely to be seen as an unprofesional approach.

(f) Deleted image. This is the most effective and secure, but removes the possibility of verifying results and is unsuitable for most vision-based applications.

Figure 4: Six methods to enhance individual privacy in images.

There has been considerable research focused on preserving privacy within the realm of machine learning (Ravi et al., 2023). A fundamental principle shared across various use cases is that

deleting data serves as the most definitive means of ensuring privacy, assuming such measures are practicable. When only non-personal data is retained, the application achieves unequivocal security concerning privacy.

2.3.2 Federated learning

In many systems relying on machine learning, being able to utilize locally stored personal data may augment the system to perform better for the situation it was created for. However, sharing this personal data with a centralized model may not be possible due to the legal bases for processing personal data (see sec:legal-bases-processing-personal-data).

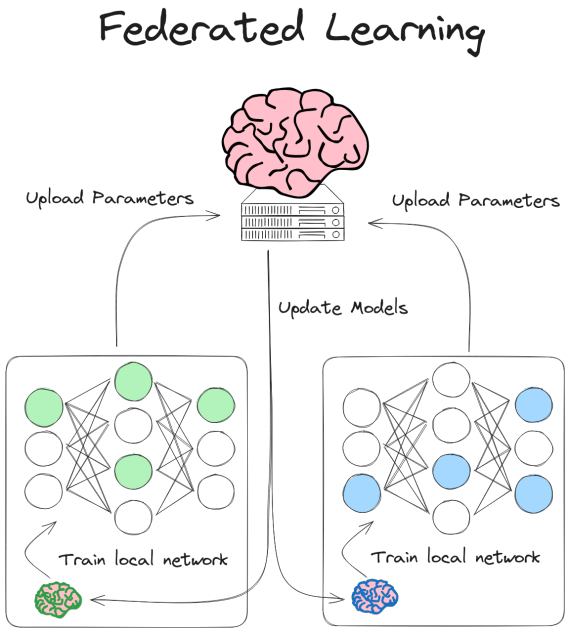


Figure 5: The federated learning process.

The concept of FL can be seen in Figure 5. Federated learning (FL), also known as collaborative learning, is a decentralized approach to training machine learning models. It does not require exchange of data from client devices to global servers. It is described in the article of Antunes et al. (2022) the following way:

> A central entity manages the learning process and distributes the training algorithm to each participating data holder. Each participant generates a local model trained with their private data and shares the resulting parameters with the central entity. Finally, the central entity employs an aggregation algorithm to combine the parameters of all local models into a single global model.

In summary, FL enables the training of ML models locally (at the location of the data) and only shares the resulting model, which is not reverse-engineerable, with the requesting party. Therefore, FL avoids the need to share the private datasets and sensitive data to others, preventing exposition to entities conducting studies and enabling data usage for broader purposes (Gu et al., 2021).

The FL process is reliant on having ground truth data on the edge for training the models correctly, but obtaining the ground truth for edge device models operating on *visual data* is difficult. The way this may be achieved, is by having a powerful edge device perform the inferences with a computationally expensive but accurate model, and using the inference results of this model as the ground truth for training a separate, possibly faster and less computationally expensive model to replace the other at a later stage. Otherwise, one could also perform the training under conditions where the ground truth is known, for example by manually inputting the number of people in an area, then having the model learn to arrive at the same count based on the camera input.

Improvement of machine learning models devices in the healthcare industry present challenges due to the sensitive nature of medical data from patients. Centralized training of machine learning models may violate laws such as the GDPR, because of the way data is being collected and used unbeknownst to the data subject (Antunes et al., 2022). To tackle these issues, Antunes et al. (2022) proposes the usage of FL[12] to tackle these issues.

Furthermore it should be noted that FL is a method to deal with the existential nature of data in edge computing devices, best described as *isolated islands*, and to use the data on edge devices before it is deleted or obscured, to improve the intelligence of the devices in privacy preservant and protective way. An important measure to take in the development of FL models is to ensure that the models are not reverse-engineerable, as the models may contain personal data. This is done by adding noise to the data, which makes it impossible to backtrack the data to the individuals. This may be done by a method such as differential privacy, which is discussed in Section 2.3.3.

2.3.3 Differential privacy

Regulations regarding personal data also applies to the events where pieces of information are aggregated to identify a person. The concept of differential privacy is to make data of individuals privacy-preservant through describing them as a group. Data from the group of people may be used, but without the possibility of backtracking the information to certain individuals. See Figure 6.

In more technical terms: Differential privacy is a statistical disclosure control algorithm to process individual data from a group to produce close-to-real outcomes without disclosing the personal data of individuals (Y. Huang et al., 2023). This means that the data is processed in a way that the results are close to the real results, but the data is not disclosed. This is done by adding noise to the data, which makes it impossible to backtrack the data to the individuals.

Differential privacy is particularly pertinent in the context of federated learning. In this approach, client devices add controlled noise to their model updates—or weights—before sending them to a centralized server. This noise addition prevents the server from being able to infer individual-specific information from the model updates. The degree of noise is regulated by a privacy parameter, often referred to as a privacy budget. This strategy allows the central server to aggregate these noisy updates from all participating nodes to update the global model. Contrary to the original statement, the noise is not removed but rather managed in such a way that the aggregated model maintains utility while protecting individual privacy (Sharma, 2023).

Note that differential privacy is a definition, not an algorithm (Dwork et al., 2011). In other words, we can have many different algorithms that satisfy the privacy demands for a given use case. For example, Dwork et al. mentions the Laplace mechanism (outlined in the same authors works from 2006) as an optimal mechanism for answering "tally" type questions differentially privately (2011). For more advanced situations, other algorithms, such as the method outlined by Blum et al. (2011), are more suitable (Dwork et al., 2011).

The big tech giants like Apple, Google and Microsoft employ differential privacy in their data collection and analysis to ensure the privacy of their users. Differential privacy is a method to ensure that the data is not personal, and thus not subject to the GDPR.

[12]Specifically, the FL method described in the works of Yang et al.(2019)

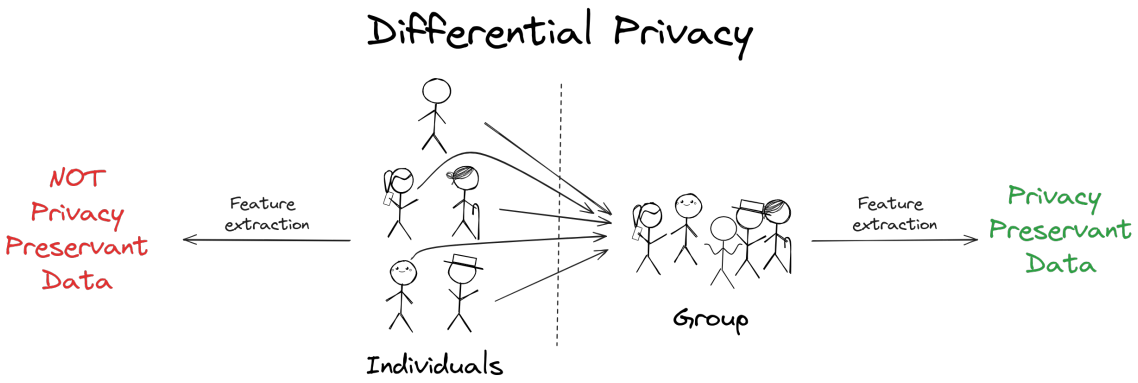


Figure 6: The differential privacy concept

2.3.4 On-device processing

According to Z. Huang et al. (2022), there are four methods for running tasks on resource-constrained edge computing devices. This is relevant in applications where user's concerns for privacy increases if data is directly transmitted to a server. These methods are seen in Table 1, and explained discussed in the following paragraphs.

Method	Advantages	Disadvantages
Data encryption	Privacy protection Fast calculation	Much bandwidth
Traditional ML	Little resource consumption	Relying on the Internet Poor robustness
Task sharing	Reducing stress on a single device	Much bandwidth Large latency
Deep learning	Privacy protection High robustness	High resource consumption

Table 1: Comparison of methods for running tasks on resource-constrained edge computing devices (Z. Huang et al., 2022)

Data encryption The first method, data encryption, would be one way of transmitting images in a more secure way. This should be done in a lossless way to maintain the image quality to preserve the accuracy of the detectors. Doing so is not trivial, and is a research field on it's own. A few methods that may function well, e.g. blurring only the faces, are discussed in Section 2.3.7.

Traditional machine learning The second method of running traditional machine learning methods, might not the greatest solution either, as they have been less accurate than the deep learning models (see Figure 13). They may, however, be a good option for devices with low computing power and memory resources as they are generally low-demanding. The methods need less data, are more transparent, but are most applicable to use cases with clear, deterministic logic. Traditional machine learning methods were the most prominent prior to 2014, while deep learning based detection models have been the completely dominant approach to image recognition tasks. Figure 7 illustrates a road map of what have been the most popular machine learning approaches to object detection. To achieve similar accuracies to those of the deep learning models but with the low computational demands of traditional machine learning, one might consider to investigate

the field of tinyML, which was scoped out of this thesis 1.3. Some considerations are, however, added in appendix D.

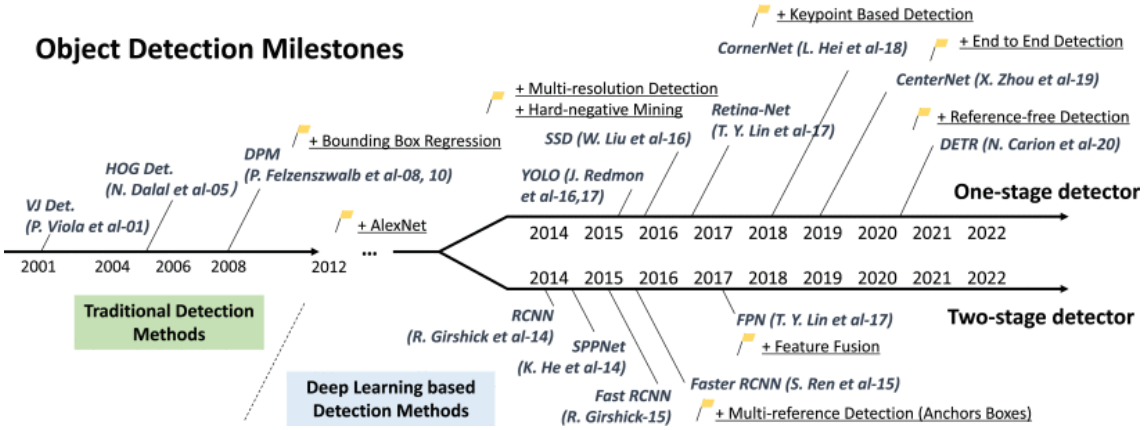


Figure 7: The evolution of object detection (Zou et al., 2023)

Task sharing The third method, sharing the workload over multiple devices, is not an uncommon practice in technology. See for example Eufy's solution with a home-base device in Section 2.7.1), where camera devices take images and send them to a more powerful computer for doing the heavy computing. This reserves privacy as the images are never sent outside of the local network, and can be achieved by simple TCP/IP[13] communication between the nodes. This gives low latency, fast networks, but introduces (1) the need of having a central hub, (2) extra work of setting up the transmission protocols, (3) another source of error and (4) the need to encrypt/decrypt images prior/post transmission to ensure security. However, due to scarcity in specialized hardware such as the GPU, this could be a nice solution, as one GPU per facility may be sufficient and achieve a higher throughput than processing large data on the CPUs of multiple edge devices.

Deep learning As opposed to traditional machine learning, This section outlines some methods to retain the privacy of individuals by using different sensors or implementing neural network on the edge devices, often referred to as on-device processing or edge computation. Which term of on-device processing and edge computation is used may be dependent on which aspect of the concept the author chooses to emphasize; the actual process that is happening on a device, or the architectural decision of making the computation on the edge.

2.3.5 Depth cameras

A widely used approach within the domain of anonymous fall detection, is to use of RGB depth cameras to capture depth information (X. Wang et al., 2020). As only depth information is captured, the data remains completely anonymous from the start.

2.3.6 Deletion of images

In an investigation of an already existing internet of things (IoT) system for wildlife monitoring: 'Where's The Bear', relying on motion-triggered cameras, three challenges of visual systems in such applications were discussed Elias et al., 2017. The drawbacks were (1) the transmission of enormous numbers (sometimes millions) of images over low-bandwidth networks, which tend to happen in automatically (motion-) triggered applications, (2) motion sensors triggered by weather conditions or by animals that were not of interest, and (3) redundancy of images taken of the same individual animal. While the 2nd and 3rd drawbacks are not applicable to this project, the 1st is.

[13]Transmission Control Protocol/Internet Protocol is a set of standardized rules that allow devices to communicate with each other on a network.

Elias et al. proposed a solution to this challenge: edge computing. Edge computing, also referred to as on-device processing, encapsulates similar concepts but emphasizes slightly different aspects of the computing approach. While *on-device processing* specifically indicates that the computational tasks are carried out directly on the device itself, *edge computing* underscores that these tasks are performed close to the data sources, i.e., at the *edge* of the network.

The deployment of visual systems in public spaces presents challenges related to privacy, not only because of the immediate access to private data, but also due to the recent breakthroughs in object detection allowing the extraction of sensitive information from visual data. The altogether only completely safe way to ensure complete and total privacy of data, is to not have the data at all.

Edge computing and on-device processing allows for the image is obscured or deleted right after analysis without ever leaving the edge device. In this way, only the anonymous analysis results are communicated online. This would mean that the personal data (1) exists *just* while the analysis is running, (2) is never sent online, and (3) is thus a lot less vulnerable to attacks. The perpetrator's device would need to be physically connected to the device and the attack would need to happen in real time. In those cases, the perpetrator could quite likely just as well take the photo himself. This is an approach to achieve low-latency, high bandwidth, high availability, low cost communications and fast response to/from the sensors.

The images would in some cases benefit in multiple ways from being obscured instead of deleted. This approach is discussed in the following paragraph.

2.3.7 Obfuscation

Another way to remove the privacy concern is by obscuring the images after analysis in such a way that individuals may never be identified.

Obfuscation is the action of making something obscure, which means to conceal or make unclear, implying it has been done intentionally. To obscure an image is often used interchangeably with "to blur", but they are not the same. To blur means to make something indistinct or hazy, suggesting something is unclear or out of focus. One might say an image has been obscured by blurring the image, or it may be done by other methods such as masking or pixelate the faces of individuals. These methods are illustrated in Figure 4.

Blurring the faces In a 2019 study, faces were detected with a thermal-detecting camera and then photos were captured with an RGB camera, blurring the area the face was detected by the thermal camera (Ma et al.). This approach is privacy preservant as long as all faces are blurred, but may fail if the algorithm does not detect all faces. In those cases, however, most humans would likely also struggle to identify a person based on the face. On the contrary, in many cases, blurring the entire image would compress the image, making it faster and easier to transfer, and be the faster option than having to detect all faces in an image.

Perceptions of privacy enhancements methods A questionnaire study of 328 students indicated that blurred images were not considered by the students to provide satisfactory privacy protection (Edgcomb and Vahid, 2012). Participants were given 18 randomly ordered videos, and were asked to rate the privacy on a Likert[14] scale from 1-5. The obfuscation methods, or privacy enhancements as they called them, and the results are displayed in Figure 8. The results show that blurred images were only considered privacy preservant for 23 percent of participants. Regardless, an important notion is that the images of this survey are from within a private home, posing higher demands and expectations with regards to privacy than what is typically done in a more public space.

[14]Likert scale: A scale of odd options, where the participant may answer a neutral middle-option and distribution should be equally distributed in both directions thereafter. An often used questionnaire scale in psychology research.

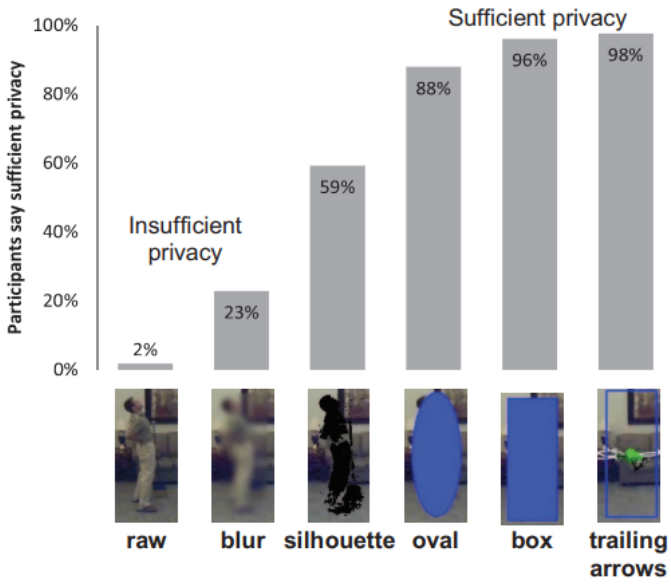


Figure 8: Privacy enhancements methods in the study of Edgcomb and Vahid (2012).

2.4 Ethical Considerations in the Development of Human Localization Technologies

As we advance the capabilities of technologies such as YOLOv9 for human localization, it becomes imperative to consider the ethical implications of our developments. The narrative of George Orwell's dystopian novel *1984* serves as a reminder of the potential societal consequences of intelligent, extensive and automated surveillance. Orwell's portrayal of a society where history is constantly rewritten and individual privacy is obliterated highlights the dangerous path we might tread if these technologies are misused by those in control of political power.

"The Party seeks power entirely for its own sake. We are not interested in the good of others; we are interested solely in power, pure power." (Orwell, 1949). This may remind of some politicians, i.e. american presidents, who may decide to deploy human localization devices to keep population under control while tightening the grip on the population...

2.4.1 Joseph Redmon Quit Computer Vision Development

Joseph Redmon, the creator of the initial versions of YOLO, decided to cease his work on the project due to its military applications. This illustrates a profound ethical stance. Redmon's choice underscores the responsibility of developers in considering the broader impacts of their work. The resignation marks a critical point in the discourse on the moral responsibilities of researchers and developers in the field of artificial intelligence and machine learning. The discussion of how to responsibly regulate and develop AI applications is still ongoing, and the decisions made by individuals like Redmon are crucial in shaping the future of the field.

Joseph Redmon's Twitter Posts:

> I stopped doing CV research because I saw the impact my work was having. I loved the work but the military applications and privacy concerns eventually became impossible to ignore. (Redmon, 2020a).
> But basically all facial recognition work would not get published if we took Broader Impacts sections seriously. There is almost no upside and enormous downside risk. (Redmon, 2020a).
> [...] I bought in to the myth that science is apolitical and research is objectively moral and good no matter what the subject is. (Redmon, 2020a).
> If you worked in a knife factory and a guy came in and thanked you for making knives because he killed many people with those knives and then he showed you a video of himself killing people with a knife you made how would you feel then about working in your knife factory? (Redmon, 2020b).

2.4.2 Philosophical Perspectives

Immanuel Kant and Deontological Ethics Immanuel Kant's deontological ethics emphasizes the importance of adhering to moral rules or duties. According to Kantian philosophy, actions are morally right if they are in accordance with a moral rule or principle, regardless of the consequences. Applying Kantian ethics to the development of localization technologies suggests that developers have a duty to uphold principles such as privacy, autonomy, and human dignity. The use of these technologies must be guided by the categorical imperative: to act only according to that maxim which can be universally applied. This means creating technologies that respect individual rights and can be ethically justified as a universal practice.

Utilitarianism and Ethical Consequences Utilitarianism, a consequentialist theory primarily developed by Jeremy Bentham and John Stuart Mill, posits that the rightness or wrongness of actions depends on their outcomes, specifically their contribution to overall happiness or utility. In the context of localization technologies, utilitarianism would require a careful assessment of the potential benefits and harms. While such technologies can enhance safety, efficiency, and convenience, they also pose significant risks to privacy and individual freedoms. Developers must strive to maximize the overall good while minimizing potential harms, ensuring that the societal benefits outweigh the risks and negative consequences.

Modern philosophers Modern philosophers, computer scientists and artificial intelligence researchers like (respectively) Sam Harris, Stuart Russell, and Eliezer Yudowsky also discuss the implications of AI. They are, however, discussing topics such as the long-term risks of artificial general intelligence, the control problem, and the value alignment problem. These are not so relevant for a human localization system without capabilites of decisionmaking. They are all voicing a concern, however, for the lack of regulations for the rapid-growing field of AI, where politicians oblivious to the nature of AI is making the regulations.

2.4.3 Practical Ethical Framework for Development

Transparency and Accountability In developing technologies capable of tracking and analyzing human behaviour, transparency and accountability are paramount. Developers must ensure that the design, implementation, and deployment processes are transparent, allowing for public scrutiny and informed consent. Clear guidelines and regulations should be established to hold developers and users accountable for the ethical use of localization technologies. This includes regular audits, impact assessments, and the involvement of diverse stakeholders in decision-making processes.

Privacy Safeguards and Data Protection Privacy safeguards are critical in mitigating the ethical risks associated with localization technologies. Robust data protection measures must be implemented to secure personal information from unauthorized access and misuse. Techniques

such as anonymization, encryption, and differential privacy can help protect individual privacy while allowing for the beneficial use of data. Legal frameworks like the General Data Protection Regulation (GDPR) in the European Union set important precedents for protecting personal data and ensuring privacy rights.

Continuous Monitoring and Ethical Auditing Continuous monitoring and ethical auditing are essential to ensure that localization technologies are used responsibly. Regular assessments should be conducted to evaluate the ethical implications of these technologies, identifying and addressing potential risks and unintended consequences. This involves establishing independent oversight bodies and ethical review boards to provide ongoing guidance and recommendations for ethical practices in the development and deployment of localization technologies.

Learning from History Just as Orwell warns against the dangers of forgetting or altering history, the AI community must remember the lessons from pioneers like Redmon. We must strive to develop technologies that do not compromise ethical standards for convenience or profitability. Historical examples of technological misuse and ethical failures should inform current practices, guiding the development of localization technologies in a manner that prioritizes ethical considerations and societal well-being.

Conclusion The development of localization technologies presents complex ethical challenges that require us to be vigilant and proactive. By embedding ethical considerations into the fabric of our technological innovations, we can avoid the dystopian futures forewarned by Orwell and ensure that these tools serve to support and enhance human society, rather than diminish it. The integration of philosophical insights, practical ethical frameworks, and lessons from history will help guide the responsible and ethical development of localization technologies, fostering a future where technological advancements align with the values and principles of a just and equitable society.

Ethical Framework for Development In developing technologies capable of tracking and analyzing human behaviour, we must establish robust ethical frameworks that prevent misuse and ensure that advancements enhance societal welfare without infringing on individual rights and freedoms. This involves transparent development processes, clear privacy safeguards, and continuous monitoring of technology deployment.

Learning from History Just as Orwell warns against the dangers of forgetting or altering history, the AI community must remember the lessons from pioneers like Redmon. We must strive to develop technologies that do not compromise ethical standards for convenience or profitability.

Conclusion The development of localization technologies presents complex ethical challenges that require us to be vigilant and proactive. By embedding ethical considerations into the fabric of our technological innovations, we can avoid the dystopian futures forewarned by Orwell and ensure that these tools serve to support and enhance human society, rather than diminish it.

2.4.4 Object Detection Performance Benchmark Datasets

There are multiple benchmark datasets for machine learning applications. The area of facial emotion recognition alone has at least five benchmark datasets (Saurav et al., 2022). For the task of object detection, the Common Objects in Context (COCO) dataset (Lin et al., 2014) has been widely used since it's introduction in 2014, with it's 330 000 annotated images.

Another well-known, widely adopted dataset for classification, object detection and segmentation is the PASCAL Visual Object Classes (VOC) (Everingham et al., 2010). The PASCAL VOC

websites include several challenges, i.e. VOC2005 through VOC2012, for researchers to benchmark their detectors. Even though the challenges have completed, one can still evaluate new methods on their datasets.

A third dataset is the CrowdHuman dataset. This may be the most relevant for a detector aiming to detect persons, as it consists of 24 370 images with in total 400 000 human instances in diverse occlusions and variations.

For any use case implementation however, it is vital to have a dataset that is relevant to the problem at hand. For a detector aiming to detect persons in a dark-lit museum, the most relevant dataset would be one with images from dark-lit museums.

In real-world applications there are licenses for using datasets for training a model. Testing and benchmarking a solution against a certain dataset is typically free to do, but the datasets are often under a license which forbids commercial use.

2.4.5 Object Detection Performance Benchmark Metrics

Machine learning can be seen as a gamified[15] version of statistics and software engineering. Object detection is a subset of machine learning. Modifications and new advances in object detection methods may be instantly evaluated by running inference on benchmark datasets and compare them to the other state of the art (SOTA) models.

Partly due to the aforementioned gamified nature of machine learning models, which metrics are deemed important may have a significant impact on the development of the models. There are competitions on the data science platform Kaggle, where data and machine learning specialists may compete for the best scores. The developers of the best-performing models are awarded prize money in many of the competitions. The target variables for the competitions are what drives development. According to Zou et al., the developments primarily pursue two main goals: enhancing prediction accuracy and increasing computational efficiency (2023. Additionally, the evaluation of object detectors extends to more, harder-to-measure, abilities. This can be their ability to transfer their capabilities to new domains, such as learning to detect a new category it has not previously been trained for. There's not yet been a focus on energy efficiency, which needs to happen soon, should development continue for AI in the current pace (Luccioni, 2023).

The most used measurement of performance for an object detector model is the *mean Average Precision* (mAP) for varying values of *IoU thresholds* (Zou et al., 2023).

The average precision is the average when taking the average of precision values under various recalls. The mean is when this is averaged for all the object classes in the dataset. The IoU represents how well the predicted box fits to the ground truth. The average precision may be calculated fixing the IoU threshold, fixing the confidence threshold, or varying them both. More on this later.

First the thesis provides an overview to understand the concepts of true positives, false positives, false negatives, the confusion matrix, precision and recall. These are easiest to explain if the task is image classification and not object detection. For 2.4.5 and 2.4.5, we will use the example of image classification, but the concepts are the same for object detection, with the difference that the bounding box positioning is also taken into account.

Understanding TP, FN, and FP, and the Confusion Matrix For a machine learning model dealing with a regression problem[16], the metrics usually used to evaluate it's performance is the number of true positives, false negatives and false positives.

[15]Gamification is the practice of applying typical elements of game play (e.g. point scoring, competition with others, rules of play) to an activity, typically as an online marketing technique, to encourage engagement with a product or service (Dictionary, 2023).

[16]Object detection is also a regression problem, as the model is simply relating the independent variable input image pixels to a dependent variable output of the bounding boxes and classes.

These may be defined as follows:

1. True Positive (TP): The number of instances correctly identified by the model as positive. For instance, if your model is tasked with identifying people in images, a true positive would be an instance where the model correctly identifies a person.
2. False Negative (FN): The number of instances where the model incorrectly identifies a positive instance as negative. Using the same example, this would be a situation where the model fails to identify a person who is actually in the image.
3. False Positive (FP): The number of instances where the model incorrectly identifies a negative instance as positive. This could occur if the model identifies a person in an image where there is no person.

The confusion matrix is a table used to illustrate these numbers. An example of a confusion matrix is shown in Figure 9.

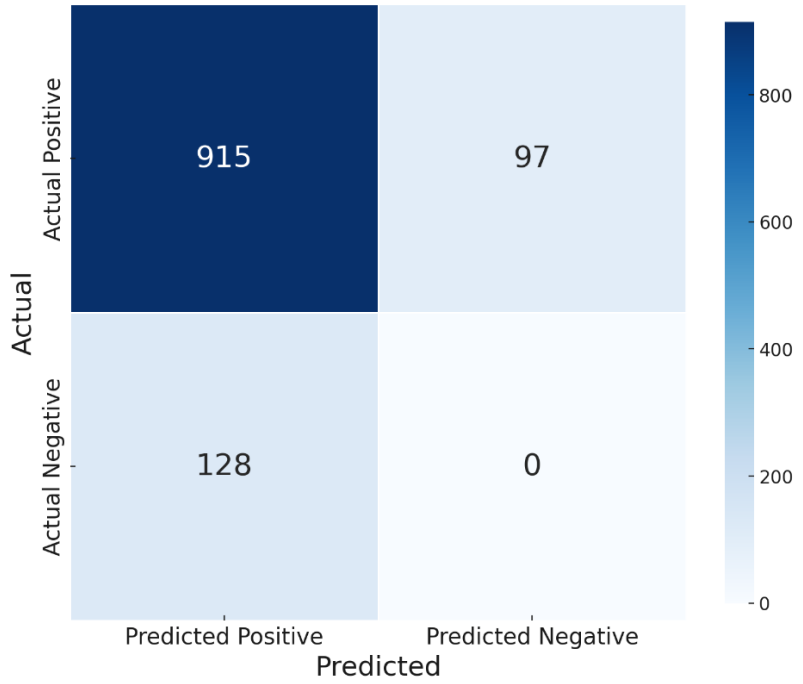


Figure 9: Example Confusion Matrix.

The confusion matrix in 9 displays that the model has detected 915 people correctly, failed to detect 128 people, and incorrectly detected 97 people where there were none. For classification tasks, it is common to have the table show which class the model has detected, and which class the object actually is. For single class object detection, the confusion matrix is sufficient as-is.

Further the TPs, FNs and FPs are used to calculate the precision, recall and F1 score of a machine learning model.

Understanding Precision and Recall For a balanced metric of precision and recall we also have the F1 Score, combining the two in a single value. Here's a breakdown of each:

Precision: Measures the accuracy of positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives. High precision relates to a low rate of false positives. For object detecion of persons, precision would be how accurate the model is when it claims to detect a person.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

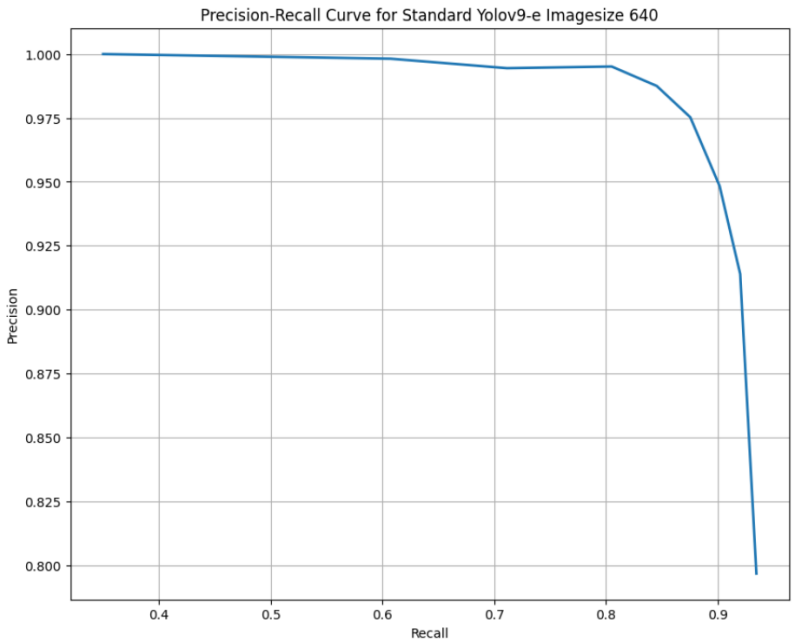


Figure 10: Example Precision-Recall Curve.

Recall (Sensitivity or True Positive Rate): Measures the ability of the model to find all the relevant cases within a dataset. It is the ratio of correctly predicted positive observations to the all observations in actual class. High recall relates to a low rate of false negatives. For object detection of persons, recall would tell us how many of the actual persons in the image the model was able to detect.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

F1 Score: The weighted average of Precision and Recall. This score takes both false positives and false negatives into account. It is particularly useful when the class distribution is uneven. F1 Score is best if there is some sort of balance between Precision and Recall in the system.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

To assess a regular machine learning model's performance, the precision-recall curve is common practice (see Figure 10 for an example). The precision-recall curve is a graph that shows the trade-off between precision and recall for different thresholds for confidence in the object class. As you allow your model to be more uncertain in it's inferences[17], the number of hallucinations will also increase and thus the precision drops. The area under this curve is the average precision (AP) of the model.

The area under the Precision-Recall curve is the average precision (AP) of the model. This can be expressed as follows:

$$\text{AP} = \sum_n (R_n - R_{n-1}) P_n \tag{4}$$

where R_n is the recall at the n n-th threshold, R_{n-1} is the recall at the previous threshold, and P_n is the precision at the n n-th threshold.

Alternatively, AP can be represented as an integral:

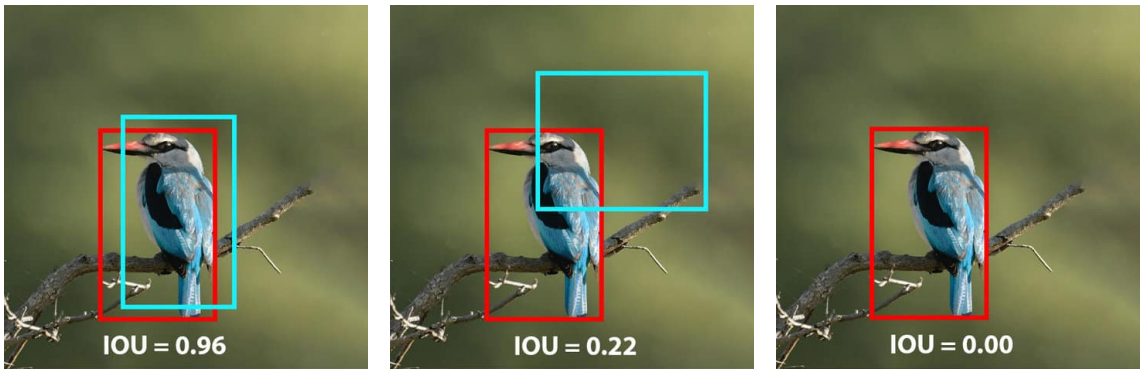
$$\text{AP} = \int_0^1 P(R) dR \tag{5}$$

[17]For object detection, there are at least three ways of allowing the model to be more uncertain. Fixing the confidence threshold and vary the IoU, or fixing the IoU and vary the confidence threshold, or by averaging over both thresholds.

where $P(R)$ is the precision as a function of recall R.

Understanding the IoU metric Accuracy in object detection refers to both detecting the object *and* it's location accurately. Combining both in one metric would simplify benchmarking. The precision, recall and f1-score all neglect the positioning precision of bounding boxes.

For assessing localization accuracy, the Intersection over Union (IoU) is calculated. This compares the predicted bounding box and the ground truth bounding box in a way so boxes need to fit as closely to the ground truth bounding box as possible to get the best score (which is 1.0). See Figure 11.



(a) True positive. (b) False Positive (c) False Negative.

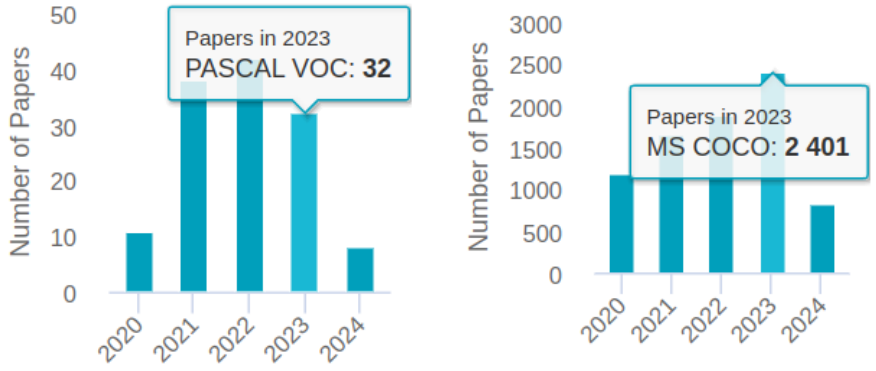
Figure 11: Intersection over Union (IoU) (OpenCV, 2022). The red boxes are the ground truth, and the blue are the model predictions.

A high IoU values equates to having a good fit with the ground truth bounding box. If the IoU value is over a certain threshold, we define the detection to be a true positive (see 11a). If the predicted bounding box has little overlap, we identify this as a false positive (see 11b). This may also be called a hallucination. If we don't have detections for a ground truth bounding box (see 11c), we have a false negative. A fourth case is where the predicted bounding box fully overlaps with the ground truth, but covers a larger area. In this case, we have a low IoU due to the high area of union, and thus a false positive.

The equation for calculating the IoU of a predicted bounding box and a ground truth bounding box is as follows:

$$\text{Intersect over Union} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{6}$$

Conclusion of Performance Metrics The Pascal Visual Object Challenge (VOC) was a standard way of measuring performance. Here, the IoU value was fixed (typically at 0.5), while the confidence in detections was averaged over multiple confidence thresholds. Today, this metric is seldom used (see 12).



(a) Pascal VOC Approximate Usage in 2023: 32.

(b) COCO Approximate Usage in 2023: 2401.

Figure 12: PASCAL VOC and COCO Approximate Usages the Previous Five Years.

The single-most important metric for object detection in the COCO dataset challenge was the mean Average Precision (mAP) over 10 IoU thresholds of .50:.05:.95. This rewards the detectors with more accurate localization (Lin et al., 2014).

Following the introduction of MS-COCO datasets in 2014, researchers started to pay more attention to the accuracy of object localization instead of using a fixed IoU threshold[18] (Zou et al., 2023).

The third method of measuring AP would be to vary both the confidence threshold and the IoU threshold. This would consider both the classification *and* localization accuracy.

2.5 Object Detection Algorithms

This subsection includes a brief summarization of the evolution of object detection, including the transition from traditional methods to morw modern methods such as the YOLO series and vision transformers.

The evolution of object detection can be divided into two major historical phases: before and after 2014, as illustrated in Figure 13. Prior to 2014, traditional object detection methods, such as the Viola-Jones detectors (Viola and Jones, 2001), Histogram of Oriented Gradients (HOG), and Deformable Part-Based Models (DPMs) were prevalent[19]. During this era, *mixture models* were developed to improve detection granularity by recognizing the different parts of the same object, such as the doors and windows of a car.

[18] A fixed IoU threshold is typically set at 0.5 or higher. Which value is best dependends on the accuracy demands of the sceneario, and is why retaining the ability to adjust the threshold is a good idea when implementing an object detector.

[19] These are just some honorable mentiones of some of the most sucessful and widely adopted models of the time (Li et al., 2012)

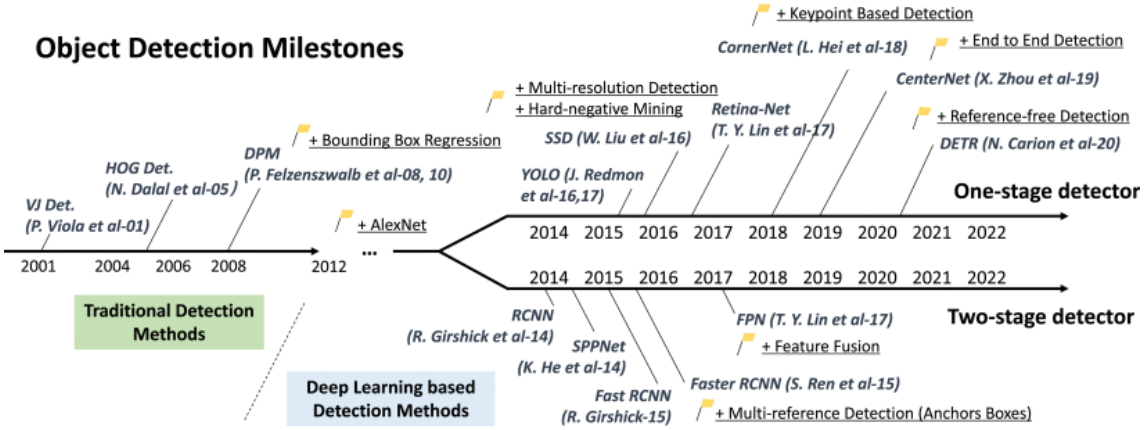


Figure 13: The evolution of object detection (Zou et al., 2023)

Despite these advancements, it was not until the introduction of Region-based Convolutional Neural Networks (R-CNN) in 2014 that the accuracy of object detection systems began to improve significantly. This paradigm shift marked a substantial advancement in the field, leveraging deep learning techniques to enhance detection performance dramatically (Zou et al., 2023). The period following 2014 has seen rapid progress, introducing sophisticated object detectors like You Only Look Once (YOLO) and Detection Transformers (DETR).

2.5.1 You Only Look Once (YOLO)

The YOLO (You Only Look Once) object detection algorithm is renowned for its efficiency in real-time object detection. YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell. Unlike traditional object detection algorithms that require multiple passes through a network, YOLO processes images in a single pass. This approach significantly enhances detection speed. The YOLO algorithm was the first one-stage (single pass) detector, and led the way for the development of other various popular networks such as the RetinaNet and DETR.

YOLOv3 YOLOv3 marked a significant advancement in the YOLO series by integrating multi-scale predictions and a deeper feature extractor. A deeper feature extractor refers to the use of more layers in the convolutional neural network (CNN), which allows the model to capture more complex features from the images. These improvements improved speed and accuracy from the first versions of YOLO.

YOLOv9 YOLOv9 represents what was the latest and most advanced version in the YOLO series at the beginning of this thesis project. It features numerous optimizations for faster training and increased accuracy, especially in challenging conditions such as low light and occlusions. YOLOv9's architecture is streamlined to reduce computational overhead, enabling it to perform well even on less powerful devices. This version also benefits from enhanced post-processing techniques that refine the accuracy of its predictions.

YOLOv10 YOLOv10, built with ultralytics and RT-DETR, is the current latest addition to the series. The commit message "add yolov10" was made on 23rd of May, signalling the first date of the release.

2.5.2 Detection Transformers (DETR)

The Detection Transformer (DETR) is an innovative machine learning method introduced by the Facebook (Meta) Research team. DETR leverages a transformer encoder-decoder architecture, similar to those used in natural language processing models. This architecture enables the model to handle complex object detection tasks by processing global information within images, rather than relying solely on local features. See Figure 14 for an illustration of the architecture.

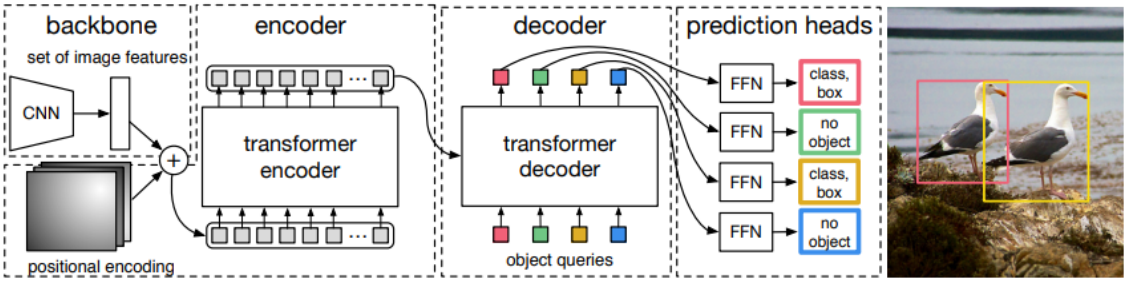


Figure 14: The architecture of the DETR model (Carion et al., 2020).

In the paper where DETR was first introduced, Carion et al. demonstrated that DETR outperforms several competitive baselines on tasks such as panoptic segmentation[20]. They achieved these results by integrating a simple segmentation head trained on top of a pre-trained DETR.

The conclusions of their study highlight the potential of DETR:

> We presented DETR, a new design for object detection systems based on transformers and bipartite matching loss for direct set prediction. The approach achieves comparable results to an optimized Faster R-CNN baseline on the challenging COCO dataset. DETR is straightforward to implement and has a flexible architecture that is easily extensible to panoptic segmentation, with competitive results. In addition, it achieves significantly better performance on large objects than Faster R-CNN, likely thanks to the processing of global information performed by the self-attention. This new design for detectors also comes with new challenges, particularly regarding training, optimization, and performances on small objects. Current detectors required several years of improvements to cope with similar issues, and we expect future work to successfully address them for DETR. (Carion et al., 2020)

The conclusions encapsulates the core innovations and findings of the DETR model. However, the authors also acknowledge the challenges that come with this new approach. Training and optimizing DETR is complex, especially when dealing with small objects. The initial performance issues on training and optimalization present challenges in terms of being able to fine-tune the models to a specialized dataset. Despite these challenges, the authors express optimism that DETR can overcome these initial hurdles, much like how earlier detectors evolved over time.

2.5.3 Comparison of YOLO and DETR

While DETR offers a groundbreaking approach by utilizing transformers for object detection, its complexity and the need for domain-specific training models can limit its adaptability and scalability. In contrast, the YOLO series, particularly YOLOv9, provides a more robust solution for real-time applications. YOLO's ability to quickly process images, coupled with continual improvements in both speed and accuracy, makes it a more practical choice for diverse and dynamic environments.

The introduction of YOLOv9 highlights a notable weakness in the DETR series. C. Wang et al. pointed out:

[20] This is a challenging pixel-level segmentation task where an image is divided into meaningful regions.

However, since it is extremely difficult for DETR series object detector to be applied to new domains without a corresponding domain pre-trained model, the most widely used real-time object detector at present is still YOLO series. (2024)

This assessment underscores the flexibility and widespread adoption of the YOLO architecture in various operational contexts. In contrast, DETR's specialized and computationally intensive requirements make it less versatile for broader applications without significant adjustments and domain-specific training. Thus, while DETR presents a novel and highly effective approach, YOLO remains the preferred choice for real-time, adaptable object detection tasks.

2.5.4 Dark-Lit Environments

Being able to detect and locate people in dark-lit environments have been previously attempted, usually for security concerns in public spaces. Park et al. developed a system for detecting people in dark-lit environments using a convolutional neural network (2020). They modified the three input channels which usually take RGB to take as input instead (i) the original infrared image, (ii) a difference image from the previous frame, and (iii) a background subtraction mask. Their dataset is vastly different from the setting for this thesis, as the individuals in their photos were far away from the cameras. However, they found that their system was able to detect people in dark-lit environments with an accuracy of 90%. This is a promising result, as it shows that it is possible to detect people in dark-lit environments using infrared imaging and CNNs. They used FLIR cameras, which make images from heat. Doing inferences on pure infrared images may be harder because the infrared radiation may be less prevalent than the heat a FLIR camera may capture. The FLIR cameras are expensive, and thus not considered viable for this project.

The article "YOLO in the Dark - Domain Adaptation Method for Merging Multiple Models" by Sasagawa and Nagahara presents a novel approach for improving object detection in low-light conditions (2020).

The key findings are:

- Domain Adaptation Method: The proposed method merges pre-trained models from different domains using glue layers and a generative model, enabling adaptation to new tasks without additional datasets.
- Model Components: The YOLO-in-the-Dark model combines the "Learning-to-See-in-the-Dark" model with the YOLO model, enhancing object detection capabilities in low-light images.
- Generative Model: By creating latent features from existing datasets, the generative model trains glue layers efficiently, reducing the need for new data and computational resources.
- Performance: The YOLO-in-the-Dark model effectively detects objects in raw short-exposure low-light images with fewer computing resources than traditional methods.

These findings highlight the model's potential for efficient, high-performance object detection in challenging lighting conditions, making it a valuable tool for human detection applications in real-world low-lit environments.

2.5.5 Transfer Learning and the Effectiveness of Fine-tuning

Transfer learning is the process of transferring knowledge from a source domain to a different but related target domain. In practice, this means having a pre-trained model fine-tune on a dataset that is specialized for the task at hand. Extending a model's capabilities to learn to correctly identify a new object class or improving the detection accuracy are typical examples of transfer learning use cases. Recent research on transfer learning for object detection models demonstrates significant accuracy gains when fine-tuning pre-trained models compared to training from scratch.

Wei et al. introduced Feature Corrective Transfer Learning (FCTL) in their study (2024). This approach enhances object detection in non-ideal visual conditions by incorporating a feature similarity loss during training. The Non-Ideal Image Transfer Faster R-CNN (NITF-RCNN) model, developed using this method, showed improved detection accuracy in challenging environments by aligning feature maps between ideal and non-ideal images.

Another approach used a generative model to create synthetic training data, which was then used to pre-train an object detector (Author(s), 2023). This pre-trained detector was subsequently fine-tuned on a limited real dataset. This method, applied to detect cars in urban environments and fish in underwater settings, resulted in improved detection accuracy compared to using real data alone. The key advantage was leveraging the large synthetic dataset to enhance the detector's initial training phase before fine-tuning it on the actual data, yielding better performance in both domains. An illustration of the process is depicted in figure 15.

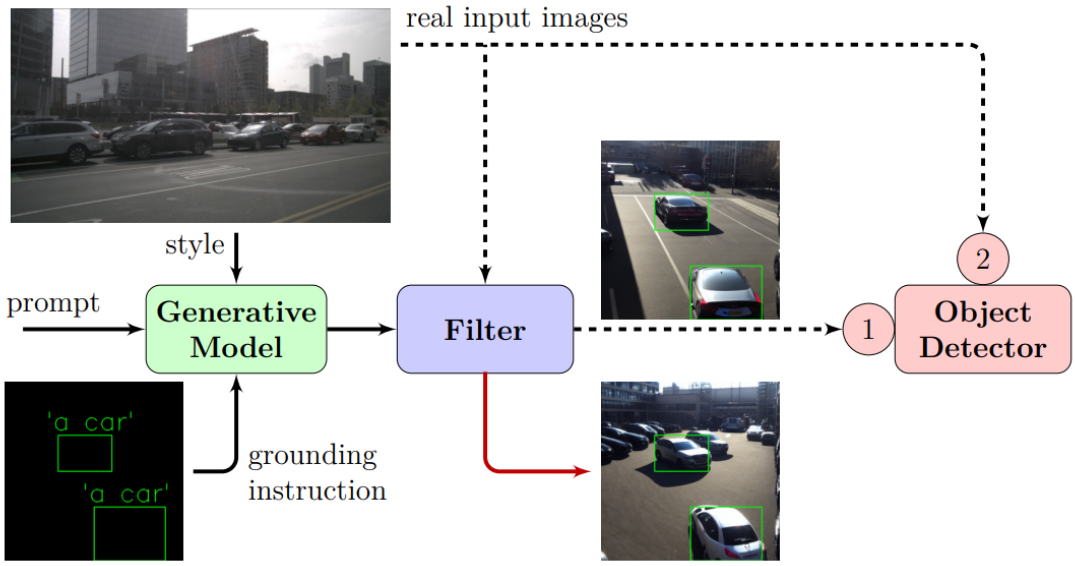


Figure 15: Transfer learning for object detection with generative models (Author(s), 2023)

Description for Figure 15 as given by the authors:

> We employ a L2I pretrained model to generate images for transfer learning to an object detector. We can filter out suboptimal generated images based on benchmark metrics. For instance, the image along the red arrow is discarded because the generative model has depicted many cars outside the bounding boxes designated in the grounding instruction. With the remaining generated images, we pretrain the object detector, followed by a fine-tuning on the real dataset. Dashed lines indicate the data used for training the models. (Author(s), 2023)

These findings highlight the effectiveness of transfer learning in improving the accuracy of object detection models, particularly when fine-tuning pre-trained models to adapt to new or challenging environments.

2.6 Third-Party Services

Roboflow is a platform designed to simplify and enhance the process of building and deploying machine learning models, particularly in the domain of computer vision. The platform offers comprehensive tools for data management, model training, and deployment, making it highly valuable for applications requiring precise object detection, including the localization and detection of persons.

2.6.1 Roboflow

Roboflow's ecosystem comprises several key components that streamline the development of computer vision models:

- **Data Management**: Roboflow provides tools for annotating, organizing, and augmenting image data. These features facilitate the creation of high-quality datasets that are essential for training accurate models. Datasets created with these tools are then stored and hosted on the Roboflow server, open for other people to use.
- **Pre-trained Models**: The platform offers a wide range of pre-trained models optimized for various tasks. Users can leverage these models to accelerate the development process, especially when combined with transfer learning techniques to adapt these models to specific tasks. This also means that any model you create yourself will be available for your potential industry competitors.
- **Model Training and AutoML**: For users without deep technical expertise in model architecture, Roboflow's AutoML capabilities offer an automated way to generate models tailored to their unique datasets. This enables a quick and easy-to-grasp way of implementing machine learning for a use case.
- **Deployment**: Roboflow enables seamless model deployment via APIs, allowing models to be integrated into applications effortlessly. This API-driven approach supports both cloud-based and local deployments, ensuring flexibility according to user needs with regards to inference speed due to network latency and data privacy and security.

The platform's ability to manage and process data through a user-friendly interface allows for rapid iteration and experimentation, reducing the time from concept to deployment.

Use Case: Detection of Persons

Roboflow excels in scenarios requiring the detection of specific objects within varied environments, such as detecting persons in crowded or complex scenes. The platform supports the deployment of models capable of identifying and localizing persons with high accuracy, which is crucial for applications in security, retail analytics, and urban planning.

One application would be using Roboflow to train models on the CrowdHuman dataset todo denne beskrevet tidligere?. Users can train custom models using this dataset, fine-tuned for scenarios such as monitoring museum traffic. On Roboflows website, there are multiple guides for how such applications may be implemented.

2.6.2 OpenAIs Generative Pretrained Transformer 4 (GPT-4) with Vision

The well known large language models (LLM) have been generalized to perform more tasks, and are thus applicable to than just text. The GPT-4 with Vision is one such large *multimodal* model (LMM). Numerous solutions already incorporate OpenAI's chatGPT as a fundamental component of their product. Expanding the role of GPT to include visual processing could potentially yield additional benefits. LLMs with vision may enable applications capable of semantically understanding scenes. This could mean the application may for example understand when a riot is about to break out in a bar street in England, or when a fish tank feeding is taking place in the acquarium, and what the crowds general reactions are to the show [21]. This may allow automated applications to provide insights to their users so they don't need to analyze the data. The resulting solution may be faster, less error prone and more scalable than the *surveillance-system with human interference*-paradigm we have today for public surveillance and intelligence.

[21] There has been considerable research focused on detecting the mood of people. This requires high resolution images of good quality. One model would then detect people or faces, and another would get cut-outs of those faces to detect the mood of each individual.

One issue arises from the generative nature of the GPTs. It is not given that a model performing well one day will be as good the next. Many experiments are performed to measure the performance of the LMM, and some show promising results. However, most experiments are frozen in time and will not reflect how well the model may perform from one day to the next. This may result in models performing well when tested, but no longer doing their jobs post-deployment.

To tackle this issue, a website has been dedicated to measure how the GPT-4 with Vision[22] performs across a range of experiments. The website is made by the team at Roboflow, but let's other users submit their experiments for daily checkups through git pull requests. Out of 13 of the experiments currently posted, 5 have failed every day the last 7 days, and 2 have failed at least once in the last 7 days. One of the experiments, counting fruits in a bowl, is alternating every day between success and failure. This proves the point that generative models may still be considered too unreliable for many applications.

Further, in May 2024, OpenAI introduced it's newest edition of the renounced ChatGPT series; the ChatGPT-4o. *o* is for omnimodal, and refers to it's ability to perform in a multitude of modalities, including vision. This model was tested for the task of object detection, but rendered unsatisfactory performances[23]. A review of ChatGPT-4o, including a more in-depth description of the experiment on object detection, is found here. The experiment is displayed in Figure 16. ChatGPT-4o, misplaced two bounding boxes when prompted to detect the dog in the image.

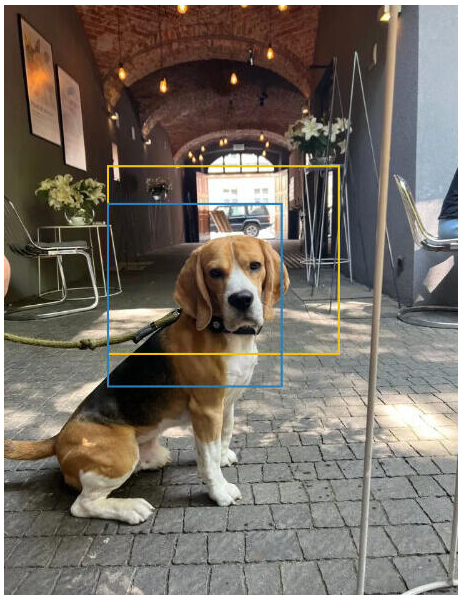


Figure 16: ChatGPT-4o Object Detection Experiment (Leo Ueno, 2024)

The two preceding sections highlights the strengths and capabilities of third-party services such as Roboflow and GPT4-V, but there are some more downsides not yet mentioned that need to be evaluated before moving forward with a third-party option. Further discussion of third-party services are found in Section 5.3.

2.7 Third-Party Products

Intelligent visual edge image/video devices seem to have an endless number of applications. In the following section we will take a look at some actors and their products to get an overview of the market of edge AI cameras.

The target of products such as the 'EufyCam 3' and the 'Aqara FP2 mmWave' is the private

[22]Previously called GPT-4V https://platform.openai.com/docs/guides/vision.

[23]The other GPT models *Gemini*, *GPT-4 with Vision*, and *Claude 3 Opus* were previously tested for the identical task, and failed.

smart home sector. The i-PRO's WV-S71300-F3 has many of the same functionalities but targets enterprises instead. These products can be seen in Figure 17 and are presented in the following paragraphs.

(a) EufyCam 3 (b) Aquara FP2 (c) i-PRO WV-S71300-F3

Figure 17: Smart cameras from Aquara, Eufy and i-PRO

2.7.1 EufyCam 3

The EufyCam 3 is a battery driven camera with solar a panel, only requiring two hours of sunlight to become fully charged. The EufyCam has functionality for face recognition. This is a self-learning AI which improves with time, up to an accuracy of 99.9% Eufy, 2022. To do this, the cameras communicate to an edge computing 'home base' to perform the machine learning tasks, and to save images to a hard drive. You may see an image of the EufyCam product in Figure 17a.

The EufyCam deals with low light situations in two different ways: a motion activated spotlight, turning on to film in the dark with a self-provided source of light, and a black and white night vision using six infrared LEDs to capture the video. There's also functionality to set activity areas, and to detect animals or cars.

Eufy's product does not assure privacy by deleting or obscuring images, but rather keeps them on the local area network (LAN). The user's privacy is preserved through storing the videos and images on a private in-house 1TB device, communicated from the camera devices via 2.4GHz WiFi.

The EufyCam 3 product is an edge computing device, and the Prasidh Chhadbria, director of Harvard Undergraduate AI, highlight three main advantages of the edge computing approach (2022). (1) Time saved, as the edge devices do not have to constantly send a lot of data to the cloud; (2) Save on cost in terms of cheaper local storage, rather than more expensive cloud storage; (3) Privacy. The data is not sent to another server, but exists on the device itself so you have more control over where your data is going.

2.7.2 Aqara Presence Sensor FP2

The Aqara FP2 utilizes multiple passive infrared (PIR) sensors rather than a RGB camera to make detections. The FP2 may detect falls, and can localize up to 5 persons in an area, but a device may only do one of the functionalities at a time. This is due to the fact that fall detection requires the device to be mounted in the ceiling, while presence detections are only accurate when the device is mounted on the wall.

However, the devices has more functionalities. With the ability to set rules for separate zones in the area, one may toggle lights only where a person is located, for example over a workbench. Making this application part of a visual system would possibly facilitate for more applications such as being able to automatically label specific items in the area. One could combine the use of zones with a visual computing module where it would only compute and analyze data, not only when something in the frame is moving (which is typical for wildlife detection cameras (2.3.6)), but when certain rules are triggered, such that a person has been located in a certain zone for a specified

amount of time.

2.7.3 i-PRO

Another big actor is i-PRO, providing AI network cameras to the market with edge computed people counting, face detection, and people attribute search (See their product: product wv-s71300-f3). The applications of the cameras they offer are often video monitoring and security features. i-PRO has informational web pages about surveillance policies and security. Most, if not all, of their cameras are NDAA compliant as well, which is a requirement to use them on american federal ground with regards to who produces the hardware of the system. Trusted manufacturers is a requirement for products capable of breaching privacy. See i-PRO's website for more information.

i-PRO had a big project where over a 100 cameras were installed in a arts museum in Monaco, where their cameras AI VMD that will give intrusion alerts when movement is detected in areas that should not be accessed, and virtual line crossing, giving alerts when people have crossed a digitally set line of an image. They also had AI scene change detection, detecting any changes in the image in a fixed part of the scenery. Also, AI people detection was used so to generate details about the visitors, so that guards that were interested in specific individuals had to opportunity to track specific individuals. These applications also generated statistics so the costumer had an overview of knowing in real time how many visitors were in the museum and even in the separate rooms or in front of each gallery. The cameras used in this solution were all fish-eye models, illustrating how fish-eye lenses may be the way to go for inside-application areas.

2.7.4 Viso

Viso.ai offers products for nearly every use case from abandoned luggage at airports, real time video stream weapon detections, detection of stopped vehicles, to parking space information. Their no-code platform (Viso Suite) enables a fast pipeline for developing new applications out of existing software. Viso.ai also has a lot of great articles on their web page regarding visual computing topics (see for example Boesch, 2023). The Viso Suite is marketed as a way to *Automate manual work, reduce development costs, solve scaleability, privacy and security end-to-end, accelerating every step of the enterprise computer vision development life cycle.* (This thesis is not sponsored).

2.7.5 VMukti

Not only does VMukti have some of the longest and most confusing product names on the market (*Real-time Edge AI based Smart Cloud Camera*), but also some of the biggest fishes in their pond of costumers. This pond includes Google, Amazon Web Services, and Microsoft[24]. One of their products, the Real-time Edge AI based Smart Cloud Camera, provides the user with a live stream of video from the camera. This may create privacy issues should the wrong user get access to the video stream, and it is likely demanding more power and network bandwidth than what it would take to only communicate the results of an analysis. VMukti's other product, the *Edge AI Based 5MP PTZ ANPR Bullet Camera VM-72BPTZ5AIVE* is listed with cutting edge technologies, including *local data processing, filtered data transfer to the cloud, and faster decision-making.* However, it's hard to figure out from their website what data is processed locally, and what their decision-making is faster than.

VMukti delivers solutions for surveillance of vehicles, school buses, healthcare, shopping malls, smart cities, warehouses, campuses, examinations, premises, elections and banking. For outside monitoring, VMukti offer cameras that may connect through the mobile network, for monitoring outside remote locations.

[24] Also Azure, which is owned by Microsoft

2.8 Summary of Literature Review

The literature review covers a wide range of topics crucial to the development and implementation of on-device processing systems for human detection and tracking in sensitive environments like museums and aquariums. The following paragraphs provide a summarized overview of the key areas discussed:

Visitor behaviour Analysis: The review explores alternative, cost-effective methods such as mobile apps and RFID tracking for visitor behaviour analysis. These technologies provide scalable solutions without the privacy concerns associated with video-based systems. Different perceptions among museum stakeholders (curators vs. administrators) highlight the diverse needs and expectations regarding visitor analytics technologies.

Privacy Concerns and Data Regulation: Insights into user perceptions of privacy in smart home environments reveal a trade-off between convenience and privacy concerns, indicating similar challenges could arise in public visitor contexts. The section on GDPR highlights the requirements for personal data protection and the legal bases for processing such data, which are directly applicable to any human localization system developed. Together with the NIS2 directive, the GDPR promotes the principle of data minimization, pushing for a solution that deletes unnecessary data as soon as the data is redundant.

Technical Aspects of Human Detection: The review delineates the privacy advantages of on-device processing over cloud-based systems, emphasizing the importance of local data processing to mitigate privacy risks. Exploration of technologies like federated learning and differential privacy illustrates advanced methods for protecting individual privacy while utilizing data for machine learning.

Impact of Dataset Specificity: The effectiveness of model performance when fine-tuned on specialized versus general datasets is discussed, which ties directly to the thesis objective of evaluating model performance in specific deployment environments.

Third-Party Services and Technologies: Roboflow and GPT-4 with Vision are evaluated for their utility in building and deploying object detection models, with considerations on their implications for privacy and data security in real-world applications.

3 Methodology

Two devices were deployed in the aquarium of "Fiskeri- og Søfartsmuseet" in Esbjerg. The devices were set in the corner of the room to capture the largest area of the room as possible. The environment, angles and final achieved image quality can be seen in Figure 18.

(a) Image taken from the 'left' device

(b) Image taken from the 'right' device

Figure 18: Images Displaying the Camera Deployment Environment and Angle

3.1 Project Outline

A dataset was collected from the devices in the aquarium to investigate the effects of dataset quality in fine-tuning of models on the performance. The details of dataset construction is found in Section 3.2. Here, we explain how the dataset consists of three partitions: *Inconsistent*, *Consistent-1*, and *Consistent-2*.

Multiple models were created to evaluate the effects of dataset quality. These include the following:

- Pre-trained "standard" models (DETR, YOLOv3, YOLOv9)
- YOLOv9 models fine-tuned on the inconsistent partition
- YOLOv9 models fine-tuned on Consistent-1[25]
- YOLOv9 models fine-tuned on the external dataset PRW
- YOLOv9 models fine-tuned on the external dataset CrowdHuman

All models were evaluated on the Consistent-1 and Consistent-2 partitions, hereby referred to as *Consistent*. Additionally, the standard models were evaluated with differing hyperparameters for input image size, and the YOLOv9 models that were fine-tuned on the inconsistent partition was evaluated with 5, 15 and 50 epochs.

3.2 The FIMUS Dataset

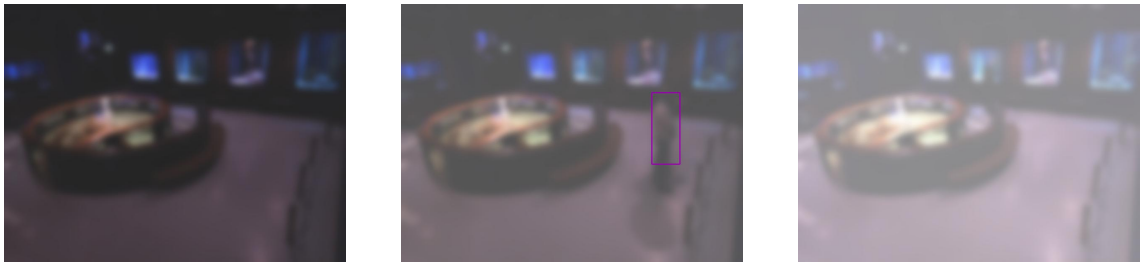
This subsection provides an in-depth explanation of the FIMUS dataset construction, including the camera configurations, the image capturing process, and the labeling process.

[25] Whilst all the other models were evaluated on the Consistent dataset, this model was only evaluated on the Consistent-2 dataset.

3.2.1 Camera Configurations

Mechanical adjustment of the aperture was ignored The Raspberry PI camera v2.1 aperture can be modified by rotating the lens with a mechanical tool,configuring its depth focus. This, however, is for very close focuses. In its default position at 0 degrees, the focus is set at "infinity". Turning the lens to 45 degrees will focus the camera at 32cm. Some applications may need this, i.e. production line systems or automated recycling facilities, but it is vastly shorter than the types of applications discussed in this thesis. All the rest of the camera settings are configured programmatically through the picamera API class. The camera settings used for the consistent images are detailed in Table 2.

Inconsistent images For the inconsistent images, the exposure mode and auto white balance mode were set to 'auto'. The automatically set values resulted in variation of image color temperature, exposure speed, and brightness, and generally lower quality images. To achieve an adequate level of brightness to label the images, the postprocessing-property *brightness* of the picamera was utilized. This resulted in artificially bright images. The brightness value was found experimentally and remotely by applying brightness and blurring the images before transmission. The brightness values of 60, 70, and 80 are displayed in Figure 19. A brightness value of 65 was used for the images in the inconsistent dataset. Finally, the last value set for the inconsistent dataset was resolution of the images, which was set to maximum (3264x2464).



(a) Brightness 60 (b) Brightness 70 (c) Brightness 80

Figure 19: Brightness values experimentation.

Further, only one person is present in each image of the inconsistent partition of the dataset. This is to simulate the probable real-world scenario of having a single technician tasked with fine-tuning a detector. The inconsistent partition is suitable for testing how a poorly captured but highly relevant dataset may function as training data for model fine-tuning. Most research in the field trains the model on a vast amount of available data, not specific to the real-world sceneario. The transfer learning experiments where models are fine tuned, the images for fine tuning are almost always of great quality. The results from this analysis aims to measure the performance when highly relevant but low quality images are utilized for fine-tuning.

Consistent images The "Consistent-1" and "Consistent-2" partitions have consistent image characteritics[26]. For these images, the camera settings were explicitly set to experimentally proven values to achieve the best image quality. The camera settings may be seen in Table 2. The consistent partition of the dataset contains images with 1-4 persons in each image. The consistent images are split in two partitions to facilitate experiments using one partition as training data and the other for evaluation. This is suitable for testing how a well-captured and highly relevant dataset may function as training data for model fine-tuning.

The settings used are found in Table 2. Note that the ordering matters when setting the picamera properties. The ordering used to achieve consistent image capturing for project of this thesis is displayed in Figure 37 in appendix A.

[26]Their differences are detailed in Section 3.2.4

PI Camera Property	Value
awb_gains	(1.5, 1.5)
awb_mode	off
brightness	55
contrast	0
exposure_compensation	0
exposure_mode	off
exposure_speed	79989
framerate	6
iso	640
sensor_mode	3
shutter_speed	80000
resolution	(3264, 2464)

Table 2: Camera settings for the image capture of consistent images.
See appendix C for a more detailed explanation of the camera settings.

3.2.2 The Image Capturing Process

Images were captured using a script that sequentially captured images, storing them directly onto a 32GB micro SD card installed in the device. This local storage approach was adopted to eliminate data transmission costs and potential security risks associated with potentially transmitting sharp, identifiable images over the internet. Instead, should unwitting individuals wander into the frames during the capturing process, these images were deleted manually once they had been transferred to the computer.

The class *Image* from the python package *PIL* was used to store the images, and to address the limited storage capacity on the computer storing the dataset images, the images were stored with a save quality value of 90.

The dataset was built capturing images while no other visitors were present in the aquarium expect those who'd volunteer to participate. This was due to the restriction detailed in the project scope (Section 1.3). A way to cancel image capturing was needed in case visitors entered the room. The simplest way of achieving this would be to pull the plug. This was challenging, however, as the devices and their power supplies were mounted high on the wall. The selected approach was to SSH[27] into the devices to start and stop the image capturing process.

An attempt was made to implement the lightweight messaging protocol MQTT for the devices to subscribe to a topic, providing them commands to perform preset operations such as capturing and storing sharp images. Through the use of a phone application such as *EasyMQTT* (for iOS), this could've allowed for command transmissions to the devices from the phone, simplifying the image capturing process and allowing for image capturing without having to bring a laptop. However, the development progress stagnated due to significant resources being diverted to resolving issues related to authentication token generation and configuring a broker accessible via a public domain name. The SSH tunneling approach was thus deemed to be sufficient.

Every once in a while, when a lot of visitors entered the room, the devices were demounted and the SD card plugged into the computer to extract the captured images. This resulted in slightly different angles when remounting the device, as finding the same configuration was challenging.

[27](Secure Shell (SSH) is not detailed in this thesis, see Section 1.3)

Setbacks in the image capturing process There were a few setbacks for the image capture process that are worth mentioning. These were: readjustments of camera settings, corrupted images, and nonlinear increase in capture time when increasing the shutter speed. The setbacks are detailed below.

Firstly, settings had to be completely readjusted between the environment the device was tested to the environment the devices were deployed. See Figure 20 to see how a shutter speed of 20 000 is in the office versus in the aquarium. This was after auto-settings were disabled and the images in theory should only vary slightly due to the slightly lower light levels in the aquarium. However, the Raspberry PI camera v2.1 seems highly sensitive to its settings.

(a) Shutter speed 20 000 in the office

(b) Shutter speed 20 000 in the aquarium

Figure 20: The effect of the same shutter speed in different environments.

A few images were corrupted and thus could not be used. This was another minor setback. One anonymous employee of a company employing the same camera in their devices, revealed that they had experienced the same issue. The occurences of corrupted images seems highly random, and only happened 4-5 times in total for the dataset capture. Examples of corrupted images can be seen in Figure 21. These were easy to detect and thus did not create further trouble.

(a) Corrupted image 1

(b) Corrupted image 2

Figure 21: Examples of rolling shutter artefacts.

Without going into further detail on the Raspberry PI camera, the image in Figure 21a reveals the camera has a rolling shutter[28]. The way images is captured with a rolling shutter is by capturing the image from the top to the bottom. The camera is constantly renewing rows of image data.

[28]The rolling shutter was confirmed by looking in the camera hardware documetation.

The corrupted rows in 21a are purple, while at the bottom the camera has resumed it's correct operation. The corrupted images are similar in size to the normal images, so the way to remove them and retake an image would be to inspect the image data values instead to determine it is mainly purple. The corrupted images were removed from the dataset. Today, Raspberry PI offers a camera with a global shutter to alleviate the aforementioned bug. As mentioned on their website, the global shutter camera is *a specialised 1.6-megapixel camera that can capture rapid motion without introducing artefacts typical of rolling shutter cameras. Ideal for fast motion photography and machine vision applications.*

The last setback, vastly more impactful than the previous, was the nonlinear increase of capture time when camera settings were modified between the capture of Inconsistent and Consistent. The reasons are still unknown due to lack of time to investigate why. The prioritization was to capture images over further developing on the image capture script, already having spent more than anticipated resources on getting the configurations set up correctly. The total time to capture an image went from approximately 2 seconds per image for the capturing of the inconsistent images, to approximately 8 seconds for the consistent images. This resulted in the relatively bigger inconsistent dataset partition of 2637 images compared to the consistent partition of 757 images.

The inconsistent images *Total number of images: total 2637. 1 subject. Captured over 3 days.*

The first iteration of image capture, and what resulted in the *inconsistent* partition of the dataset, was made with non-optimized camera configurations. To sufficiently brighten the images, the picamera.brightness attribute was set to 65. This is a postprocessing operation, which gave brighter but also artificially lit images. Also, the camera would sometimes focus on the bright fish-tanks in the museums, rendering the rest of the image rather dark. This was an effect of the awb mode and exposure mode being set to auto, and led to images of varying brightness and color. These images were still included in the dataset however, as images seen as suboptimal to the human eye may still be useful to the training of detectors. These images may be used to inspect the impact of captured image quality on inference performance.

The images were then used to build a proof of concept for the project pipeline, verifying and developing the steps needed for a successful project.

The consistent-1 images *Total number of images: 292. 1-4 subjects.*

For the second image-capturing session, the camera configurations had been more thorougly tested to obtain more consistent images in terms of colors and brightness. This means using non-auto auto white balancing and exposure settings, and reducing the amount of post-processing brightness adjustment. Also, some friends were invited in this session. Due to a reduced post-processing brightness augmentation, the exposure speed had to be increased to get sufficient light in the images. This meant more unclear outlines of moving subjects in the frame. It also meant more time was spent capturing and storing each image. This increased from $1.3\frac{s}{image}$ to $6.3\frac{s}{image}$, which means the time available for image capturing was spent less productively than with the previous camera configuration. Depending on the impacts of image consistency on inference accuracy vs. amount of training data, capturing with a higher exposure speed and then post-processing the images to be brighter might be the better solution. Also, as pointed out by TODO insert mikkels master, augmenting the brightness might only slightly impact the model performance. This is because a model may see slight differences in pixel-level values invisible to humans, thus enabling it to still recognize the patterns of human outlines. A sufficiently bright image would still be required for the sake of model verification and ground truth obtainment (by human annotators).

The camera was repositioned three times during this iteration of image capturing. This is a drawback as it complicates the process of mapping the person positions in the images to real world locations in the aquarium. This is because the positions are represented as x,y values from the corner of the image, and for a person standing at exactly the same position in two images, the x,y-values will differ if the camera position has moved. Serving as a dataset for machine learning applications and not for analytics generation based on real world positions, this was not an issue.

NoIR camera During the 2nd iteration of image capturing, 60 images with a Raspberry Pi NoIR camera module version 2.1 were captured to determine its efficacy in enhancing human detection under low-light conditions. The "no" in NoIR signifies it's lack of an infrared filter. It was hypothized by the author that this meant the camera could then operate with a lower shutter speed, which showed promising results in initial tests. However, once deployed in the aquarium, this proved to be wrong. The NoIR camera is said to give the ability to look in the dark *with infrared lightning*. Despite its potential, the noir camera was used as a regular camera module thereafter, capturing a different angle than the first device, for the remaining image capturing iterations. The 60 images were not used in the project, as the models trained on inconsistent data had already been trained.

The consistent-2 images *Total number of images: 465. 1-2 subjects.*

Similar to the consistent-1 images, but with 1-2 subjects instead of 1-4. In this iteration of image capture, the left camera was also used to capture images for the dataset. The dataset characteristics are detailed in 3.2.4.

3.2.3 Labeling

The detector requires precise ground truth positions of persons for training, validation, and testing. This data is obtained through a process known as image labeling or annotation.

To expedite the labeling process, the images were initially processed using a pre-trained YOLOv9 model on the COCO dataset, rather than manually labeling each image. Out of the 2939 images in the first-iteration dataset, the model prosuced 1863 detections that needed verification. This includes modifications, deletions, and additions to the annotations. The remaining 1076 images, which had no initial detections, required manual labeling from scratch.

Additionally, validation of the annotations uncovered specific errors: in 74 images, moving seaweed in one of the fish tanks was mistakenly identified as a human due to its human-like movement and shape (see Figure 22). In another instance, a person carrying a ladder was incorrectly recognized as one person carrying another.

(a) Seaweed and (presumably) a person

Figure 22: Sometimes, the seaweed is deemed 'more' person than the human.

Label Studio "Label Studio" was used to label the images. This online tool allows for setting up a machine learning backend for automatically generating predictions for the images, to speed up the process. The setup of this backend was not trivial, however, and another approach was taken. The images were inferred on, the labels converted to label-studio json format, and then imported. This was a less 'automatic' approach but nevertheless effective. The label-studio tool was the used to modify, delete, and add annotations to the images. Finally, the annotations were exported and converted to the YOLO format. The code for converting yolo-to-label-studio and label-studio-to-yolo is found in Other/Code/Utils on https://github.com/Hallvaeb/masterthesis.

3.2.4 Dataset Characteristics and Applications

The FIMUS dataset consists of in total 3394 images, of which 2637 are in the inconsistent partition, 292 are in consistent-1 and 465 are in consistent-2 (Consistent total: 757 images). The dataset is well suited for the task of measuring and analysing the impact of image quality on the performance of object detectors, and whether a model performance on a general dataset is a better or worse indication of real-life performance than a specialized but poorly captured dataset.

The rough standard for train-validation-test splits is 60-80% training data, 10-20% validation data, and 10-20% test data. If the images in Inconsistent are used for training and validation, and the images in Consistent are used for testing, we would get a have 78% data for training and validation, and the remaining 22% of data for testing.

Another application of the dataset would be to use Consistent-2 for training and Consistent-1 for testing to measure the impact of a small, but highly relevant dataset from two different angles on a fine-tuned model performance on images from the same environment and with the same settings. This would give a split of 61% for training and validation, and 39% for testing.

Consistent-1 and Consistent-2 differences The differences in angle and colors are visually represented in Figure 23 and are explained below.

Consistent-1 is from a group of friends of 4 people in almost all the images, moving around the aquarium and talking in a group in various location. All images are taken from the right device, i.e. nearly the same angle. The subjects are 3 persons of approximately 1.80m height, 2 female and 2 males and various light and dark clothes, all wearing pants. Consistent-1 are the closest representation of the images the device will be capturing in the experimental setting.

Consistent-2 has many images with a single subject, then two subjects. The images are primarily from the right device, but contains some taken from the left device. The subjects in Consistent-2 are both approximately 1.80m, one wearing glasses. One wearing a white tshirt, the other wearing a black tshirt, and both wearing shorts.

(a) Consistent-1 'right' image. Captured with a regular camera.

(b) Consistent-2 'left' image. Captured with a noIR RPI camera.

Figure 23: Example representative images of the Consistent-1 and the Consistent-2 partitions. Both include 'right' images, but only the Consistent-2 includes 'left' images.

3.3 External Datasets

This project utilizes multiple external datasets for developing and testing the object detection models. Each dataset was selected based on its relevance to the project, specifically for containing labeled images of the person class, and they vary in the number of images, capturing angle, and image diversity.

3.3.1 Common Objects in Context (COCO)

The COCO dataset is a large dataset of 118 000 images and 80 different classes. The COCO-2017 train dataset was used to pre-train the models. The COCO-2017 validation dataset was used to evaluate the performance of the finalized models, as is industry standard.

(a) Example Image 1

(b) Example Image 2

Figure 24: COCO Dataset Example Images

Figure 24 is a great example of the widespread nature of the COCO dataset images. This makes for a great dataset for pre-training, as the trained model will have knowledge of a wide array of objects. It may then be wise to fine-tune such a model to a more specific use case, so the model can see more of the specialized data.

COCO was introduced in the article of Lin et al. (2015).

3.3.2 CrowdHuman

CrowdHuman, the largest dataset used, focuses exclusively on images where people are the main subject, contrasting with COCO's broader class range. This dataset was employed to assess how additional data might enhance model performance, with experiments conducted across various training data volumes.

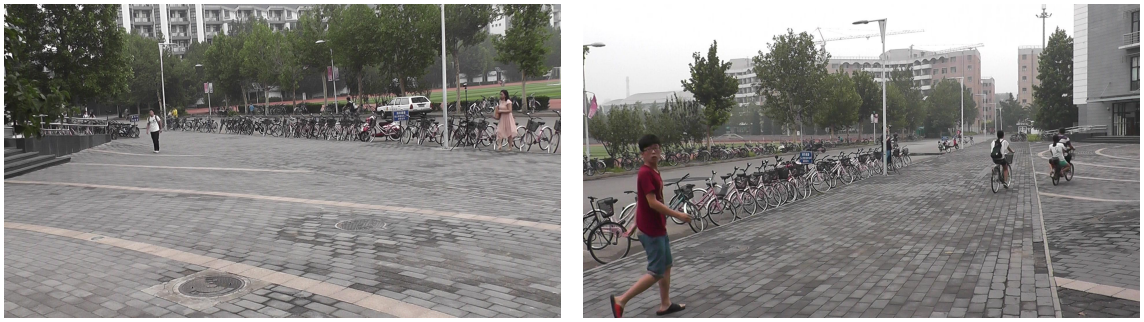
(a) Example Image 1 (b) Example Image 2

Figure 25: CrowdHuman Dataset Example Images

The CrowdHuman dataset was presented in the article of Shao et al. (2018).

3.3.3 Person Reidentification in the Wild

Person Reidentification in the Wild comprises 11,816 images of pedestrians and aligns closely with our application needs as it exclusively contains images of people. This dataset's relevance is heightened by the presence of occlusions and the similar scale of persons to those detected in the aquarium setting. The dataset contains 932 individuals, annotated in 34 304 separate annotated boxes. Although designed to facilitate the development of reidentification applications, this functionality was not utilized in this project (refer to the project scope in Section 1.3 for details).



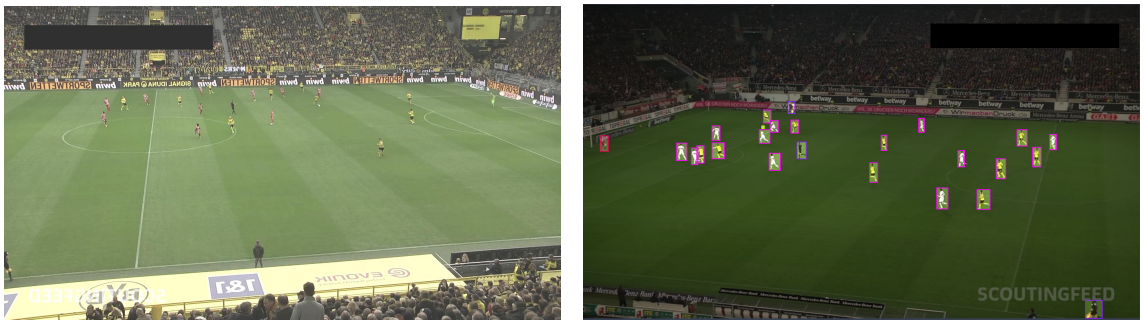
(a) Example Image 1 (b) Example Image 2

Figure 26: PRW Dataset Example Images

The PRW dataset was presented in the article of L. Zheng et al. (2017).

3.3.4 Football Players Detection

The football-players dataset, characterized by its uniform perspective and consistent lighting and quality, was introduced to determine whether model improvements derive solely from specializing to single-class data or if the specialization's quality and relevance are crucial. It also provides a clear contrast in dataset characteristics, aiding in attributing performance differences to dataset nature rather than other confounding factors. A weakness that may confuse a model under training may be that the audience are not labeled. This is illustrated in Figure 27b. Therefore, fine-tuning a model on this dataset may result in a model that ignores perons of such tiny scale.



(a) Example Image (b) Example Labeled Image

Figure 27: Football Players Detection Dataset Example Images

The Football Players Detection dataset is available at Roboflow.

3.4 Model Training

In the thesis project, YOLOv9 was pre-trained on the COCO dataset and then optimized by fine-tuning on specialized data. The process of fine-tuning a pre-trained model is known as transfer learning (see Section 2.5.5). The rest of this section outlines the processes and choices taken in the training process of these models.

Licenses A successful effort was made to create a system that is free and open to use, but some conditions apply. *YOLOv9* is under a GPL-3.0 License. This is a copy-left licensing, meaning it is free to use but has the requirement that any derivative works are released under the same rights. Another algorithm discussed in this thesis, the YOLOv3, is under a free-to-use license (AGPL-3.0), but changing the code is not allowed. The final object detector algorithm discussed, DETR, is under an Apache-2.0 license, which permits users to use and modify the code to fit their needs. This could thus provide a solution for a company interested in keeping their solution hidden from competitors.

3.4.1 Hyperparameter Tuning

One approach to tune the hyperparameters is to utilize Autogluon, an auto machine learning library. Installation may be tricky, but one may use this guide for installation: AutoGluon guide. Note that in some cases, 'pip install autogluon' must be evaluated twice. Further, this guide could be used to tune model hyperparameters.

Another great guide for hyperparameter tuning was found on Kaggle. However, this would entail optimizing the hyperparameters for each of the datasets used in this project. Hyperparameter tuning is a time-consuming process, and the author of this thesis did not have the necessary means to perform this task.

For this project, a simpler solution and less effective approach to hyperparameter tuning was adopted. This was to use the standard out-of-the-box hyperparameters. This is one source of bias in this project, as some standard parameters might be optimized for a certain dataset size. The fine-tuned models would achieve better performance had the hyperparameters been tuned. Due to the large scope of this project however, hyperparameter tuning/hyperparameter optimization was not prioritized.

One modification was made to the hyperparameters however, which was the number of epochs. A major challenge and setback for the model training was a mistake made in choosing a slightly premature YOLOv9, for which the validation process was not yet implemented. This means that the models were essentially trained blindly, without providing the data to indicate whether the

models overfitted or could benefit from more training. This is apparently still (24-05-2024) a github issue for YOLOv9. A method to manually compute losses for the models was not implemented. Instead, the models were trained for 5, 10, 20 and 50 epochs, which are quite short periods.

Mosaic data augmentation was used to enhance the datasets for the training process. This consists of several processes to create more training data from the the available images. Example images resulting from mosaic data augmentataion are seen in figure **??**.

(a) Example Mosaic Image From the FIMUS Inconsistent Dataset Partition

(b) Example Mosaic Image From the PRW Dataset

Figure 28: Mosaic Data Augmentation

3.4.2 Google Cloud Services

Google Colab was used to train the models on VMs with GPUs to speed up the training process. The training thus took about 4.5 minutes per epochs for the FIMUS inconsistent dataset partition with 2636 images, and nearly 15 minutes per epoch for the 11815 images dataset PRW. Training on the CrowdHuman dataset took nearly 20 minutes per epoch, with it's 15000 training images.

The pros with Google Colab is the possibility to borrow computational power without having to invest in expensive computers. For this project, 200 compute units, costing about 25USD, were sufficient for the training process, making it a cost-effective solution for developing a specialized system. Google Colab also allows for seamless file exchange by mounting a Google Drive to the VM, though the solution necessitates giving explicit consent to Google Drive that they may see and download all your data, including photos on Google Foto. For the convenience the service brings, however, this potential privacy infringement may be acceptable[29]. Another pro is the possibility to co-operate on notebooks. However, personal experience suggests that collaboration can be problematic, as changes made by one user require saving and refreshing the notebook for others to see

[29]This perspective is ironic and hypocritic, see Section 5.4

There were also some cons with using Google Colab. As many cloud services, the optimal usage requires a connection to the VM. This connection would often be lost, even with the Google Colab page open using a Google Colab Pro subscription. The Google Colab computer is stateless, meaning that when this happens all data is lost. This resulted in hours of training progress being lost, having to connect to a new machine and download weights and training files all over again before initiating training. This is the motivation behind why the models in this project were trained in 10 epoch-intervals, saving the weights file to consistent storage (Google Drive) every 10th epoch of training.

3.4.3 Validation Data

Since the hyperparameter tuning process of this thesis project was not of focus, the choice was made to not use validation data during training. Validation data is used to assess the models training process and to identify overfitting/underfitting. After the hyperparameters, such as the number of training epochs, has been set, the best practice is to train the model once more, using all available data to train. Google Colab was used to train the models on GPUs, and the initial plan was to train multiple more object detection models. The initial training was on the Consistent-2 dataset partition containing only 465 images and so the decision was made to drop the validation data for this model. Then, after multiple retries where long training sessions were lost due to disconnects to the Google cloud server and computing resources and time running out, the decision was made to train without the validation data. The fine-tuned models could then be assessed on the test set rather than the validation set. This was more overhead as it generated a lot of models that all needed to be evaluated, but it also means that once the models are complete there's no need to rerun with the optimal number of epochs and all the data from train and validation sets to achieve the best model.

3.5 Model Presentation

The resulting models of this thesis project were the following:

1. YOLOv3 (not fine-tuned)
2. YOLOv9 (not fine-tuned)
3. YOLOv9 Fine-Tuned on FIMUS Inconsistent
4. YOLOv9 Fine-Tuned on FIMUS Consistent-2 (and will be evaluated on Consistent-1)
5. YOLOv9 Fine-Tuned on CrowdHuman
6. YOLOv9 Fine-Tuned on PRW
7. YOLOv9 Fine-Tuned on Football Players Detection
8. DETR with a ResNet50 backbone (not fine-tuned)
9. DETR with a ResNet101 backbone (not fine-tuned)

The final YOLO model is a standard *YOLOv3* model from Ultralytics. This was added to see the performance gain from upgrading an older version to a newer YOLO version, and increase validity of the results by testing that our experiments gains results in accordance with previous research showing YOLOv9's superiority over YOLOv3.

The last 2 models are built on the DETR architecture. The first is built with a ResNet50 backbone, while the other implement a more complex ResNet101 backbone. These are included as an alternative approach to the YOLO algorithm.

3.6 Model Evaluation

As mentioned in 2.4.5, there have been multiple ways for object detection model evaluation. The most widely used has been to fix the confidence threshold, and average over 10 IoU thresholds from 0.5 to 0.95 in steps of 0.05. This is hereby denoted as COCO AP. For this thesis, both COCO AP and the more computationally expensive where both confidence and IoU thresholds are varied has been implemented to see if there's a different outcome for model evaluation based on which version of the evaluation metric is chosen. The more computationally expensive version is denoted as Vary-Both AP.

What input image size is optimal depends on the dataset and use case, and should be tested for a given sceneario. According to James Gallagher, to increase the input image size will augment the accuracy of a model:

> We trained our model on images with a size of 640, which allows us to train a model with lesser computational resources. During inference, we increase the image size to 1280, allowing us to get more accurate results from our model. (James Gallagher, 2024)

This postulates that even though a model is trained on images with size 640, more accurate results may be obtained by increasing the input image size during inference. This hypothesis was tested in the project, see Section **??** where input image size 320, 640 and 1280 were compared in terms of accuracy and inference latency. Except from this experiment, the models in this project inferenced with input image size of 640.

The pre-trained weights were available in multiple sizes. The largest weights-file, called 'yolov9-e', is what has been used for this project. These weights are available for download on the Yolov9 Github repository. An assessment regarding the differences in accuracy of the different available pre-trained weights was also made. This includes the available weights as of May 2024, yolov9-m, -c, and -e.

3.7 Ethical Considerations

In the deployment of advanced machine learning technologies for visitor localization and engagement analysis, this research proactively addresses privacy concerns through the implementation of image obscuration techniques. These methods ensure that no personally identifiable information is captured or communicated, thus significantly reducing privacy risks associated with visitor tracking in cultural spaces such as museums and aquariums.

3.7.1 Privacy by Design

At the forefront of our ethical approach is the principle of "privacy by design." This concept involves integrating privacy into the development and operation of our tracking technologies from the outset, rather than as an afterthought. By employing image obscuration techniques, such as real-time pixelation or silhouette generation, we ensure that the visual data processed by our system remains anonymous. This method effectively eliminates the possibility of identifying individual visitors from the captured data, thereby safeguarding their privacy.

The application of these privacy-preserving techniques negates the need for explicit consent from visitors for two primary reasons. First, the anonymization process occurs instantaneously as the data is captured, meaning no identifiable information is ever stored or analyzed. Second, the focus of the research is on aggregate behavior patterns rather than individual actions, further distancing the study from privacy concerns.

3.7.2 Ethical Use and Data Protection

Ensuring the ethical use of technology extends beyond privacy considerations to include the responsible handling and protection of any data generated by the system. Although the data is anonymized, we are committed to maintaining high standards of data protection. This includes secure data storage, limiting access to authorized personnel, and employing robust data management policies that comply with relevant data protection laws and guidelines.

The utilization of anonymization techniques also reflects our commitment to minimizing any potential impact on visitor behavior and the overall museum or aquarium experience. By ensuring that the tracking system is unobtrusive and does not compromise privacy, we aim to maintain the integrity of the visitor experience, allowing individuals to engage with exhibits without concern for their privacy.

3.7.3 Transparency and Accountability

While the technical approach effectively addresses privacy concerns, maintaining transparency about the use and purpose of tracking technologies is still essential. Information about the tracking system and its privacy-preserving nature will be made available to visitors, ensuring they are informed about how data is used to enhance the visitor experience.

Furthermore, the project will adhere to an ongoing ethical review process, ensuring that all aspects of the research remain aligned with ethical best practices and respond to evolving technological and societal standards.

In summary, by prioritizing privacy through the use of image obscuration techniques and adopting a comprehensive ethical framework, this research aims to advance the understanding of visitor engagement in a manner that is both innovative and respectful of individual privacy rights. This approach sets a precedent for the ethical application of machine learning technologies in cultural institutions, balancing the benefits of visitor behavior analysis with the imperative of protecting privacy.

3.8 Heatmaps

Heatmaps are a powerful visualization tool that can provide insights into visitor behavior patterns and engagement levels within a museum or aquarium setting. By aggregating anonymized data from the tracking system, heatmaps can reveal areas of high visitor activity, peak visitation times, and popular exhibit locations. These visual representations offer valuable information for museum staff and curators, enabling them to optimize exhibit layouts, plan interactive experiences, and enhance visitor engagement. For this project, heatmaps were attempted to be created using 3 different python packages.

On the attempts to make heatmaps The first attempt was made asking chatGPT 4 to provide the code. The AI chose to draw circles using the python package "OpenCV", which without modifications did not render satisfactory results. Instead of tweaking a suboptimal solution, the attempt was then made to use the modules from Ultralytics to create the heatmap. The results of the first attempts may be seen in Figure 29.

Ultralytics, a company from Los Angeles, is the same company that developed YOLOv5 on which the YOLOv9 is built upon. They also have premade modules for creating heatmaps. However, the solution neccessitates a detector model to make inferences live, and has no optional arguments to pass your own inferences. An attempt was made to modify the code and pass mock-data in the right format, but the result was unsatisfactory. The heatmaps were thus created with another module instead.

(a) Heatmap Draft 1: ChatGPT-4 solution

(b) Heatmap Draft 2: Ultralytics solution

Figure 29: Heatmap Development Drafts

3.8.1 Supervision Heatmaps

Supervision is a module created by Roboflow, to make reusable and user friendly computer vision tools. It is designed to be model agnostic. The github repo may be found here.

The solution incorporating Supervision rendered satisfactory results. An example image is provided in Figure 30. This solution supports generating heatmaps from data in a pandas dataframe, allowing for filtering the dataframe to generate the preferred heatmaps based on any variable. This could be the interesting times of the day aggregated over a month, (e.g. every weekday from 10-11), or for a given time interval (e.g. week 39, 2024).

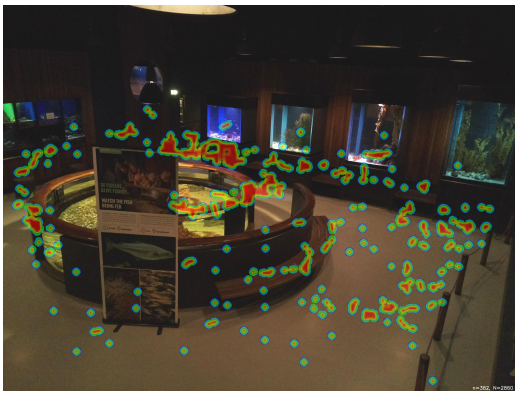


Figure 30: Final Heat Map Example. This is for one day in May.

4 Project Results

This section presents the results of the human localization system. This includes a diverse range of experiments relevant for the objectives of this thesis. The results are discussed with regards to their implications and significance in the context of the research questions and objectives outlined in Section 1.4, and in the broader context of the field.

4.1 Object Detection Model Evaluation

The characteritics of the dataset partitions allow us to execute several experiments. The experiments are designed to answer the research questions and objectives outlined in Section 1.4. Various evaluation jupyter notebooks used in this thesis to evaluate the models are found at this github.

COCO Average Precision The tables 3, 4, and 5 shows the AP scores when the models are evaluated with "COCO AP". This means to only average over 10 IoU thresholds, while keeping the confidence threshold fixed. As mentioned in Section 2.4.5, this has been the de-facto method of calculating average precision since it's introduction with the COCO dataset challenge in 2017. The confidence threshold is kept fixed at 0.5, which is the default value for the COCO challenge.

4.1.1 Fine-Tuning On Consistent-2, Testing On Consistent-1

The test-set should cover as many situations as possible that the detector may face after deployment, and be as diverse as possible. The Consistent-1 and Consistent-2 are similar, but have some minor differences (differences were detailed in Section 3.2.4). These differences may affect the results, and obscure whether the results are due to the differences in the dataset sizes or the differences in the dataset content. Regardless, using a smaller test-set reduces validity of the results. It is necessary, however, when a portion of the test-set is needed for fine-tuning. Since the same data must not be used for training and validation or testing, the models displayed in 3, which were fine-tuned on Consistent-2, are only evaluated on Consistent-1.

Model	AP50	AP75	AP90	mAP50-95
Not Fine-Tuned	0.982	0.959	0.935	0.959
Fine-Tuned 5 Epochs	0.969	0.881	0.645	0.815
Fine-Tuned 10 Epochs	0.976	0.889	0.660	0.823
Fine-Tuned 20 Epochs	0.971	0.883	0.648	0.821

Table 3: COCO APs comparison of various YOLOv9 models fine tuned on Consistent-2 (465 images) and evaluated on Consistent-1 (292 images).

As we will see in the following tables, the models are all achieving relatively high scores for object detector models, but the model which was not fine-tuned is outperforming the others. This may be due to the low number of epochs they are trained for. Another reason to the fine-tuned models performing poorly may be due to the simplicity of the dataset. The incredibly high score for the not fine-tuned YOLOv9 model is the reason to why further investigations with more epochs was . The following section uses a larger test-set to evaluate the models.

4.1.2 Larger Test-set

Table 4 illustrates that adding images from the Consistent-2 to the test-set *reduces* the scores for YOLOv9 while it *increases* the scores for YOLOv3. These fluctuating scores when introducing

more test data indicates one of two things. Either the scores have yet to converge, or the newly introduced test data is different from the previous test dataset.

Model	AP50	AP75	AP90	AP50-95
YOLOv3 (n=292)	0.985	0.900	0.608	0.826
YOLOv3 (n=757)	0.804	0.742	0.510	0.681
YOLOv9 (n=292)	0.982	0.959	0.935	0.959
YOLOv9 (n=757)	0.805	0.787	0.769	0.787

Table 4: Performance Metrics of Object Detection Models on 292 images vs 757 images

Whether the scores have yet to converge or the introduced data provides new situations for the detector to be evaluated on is not trivial. Regardless, the results indicate that the test-set should be as large as possible to ensure the results are valid.

4.1.3 Input Image Size

We experimented with the input image size for model inference with the pre-trained, not fine-tuned version of YOLOv9 to find the optimal value. 320, 640 and 1280 were tested. The input image size affects accuracy and inference latency of the models. The inferences were performed using an 8-core AMD® Ryzen 7 4700u CPU with 16GiB RAM[30], resulting in higher inference latencies than what are usually reported for similar models. In addition to the hardware, other simultaneuos tasks on the computer may affect inference latency. Therefore, no other user input was given during these model inference runs. However, background processes and other programs were not terminated during the process, so the numbers are only not valid outside this thesis. More valid results could have been achieved by running the inference many times and taken the average inference latency.

The weights file also affects the inference latency. The YOLOv9 is released with three sets of pre-trained weights. These are called yolov9-m, -c, and -e (as of 03.06.2024). The results of Table 5 were achieved using the yolov9-e weights.

Model	Input Image Size	mAP50-95	Inference Latency
YOLOv9	320	0.820	302ms
YOLOv9	640	0.945	840ms
YOLOv9	1280	0.877	3336ms

Table 5: COCO APs comparison of Yolov9 models with various Input image sizes evaluated on Consistent

Table 5 reveals the effects of input image size on model performance. Performing model inference with an input image size of 1280 was hypothesized to augment the accuracies, but this was not the case in our experiment. Introducing a much higher inference latency and poorer performance than of the 640 image size, the 1280 is clearly the worst option. This is likely due to the scale of the persons in the FIMUS Consistent dataset partition, which does not necessitate a higher input image size than 640. Inferring with size 320 may be a viable option in applications where speed is important.

[30]The inference machine hardware specification is mentioned here, and not in the methodology section, as this is the only place in the thesis where speed is discussed and the inference machine hardware is of relevance. The machine's hardware does not affect inference accuracy.

4.1.4 Model Average Precisions on Consistent Dataset:

The models were evaluated using COCO AP *and* Vary-Both AP. This increases reproducibility as further development and evaluations may freely choose between the two. The rankings of the models is similar for the two evaluation metric methods, proving the choice in evaluation metric is not detrimental to the process of finding the optimal model for a deployment sceneario. The results are discussed most in-depth in the COCO AP section.

COCO AP Figure 31 displays how the AP of the models varied with number of epochs. The best model was achieved at 70 epochs. Another notable discovery from this graph is that we hit a local maximum at 20 epochs. Had we need continued training past 60 epochs we would not have know the model would improve at 70 epochs.

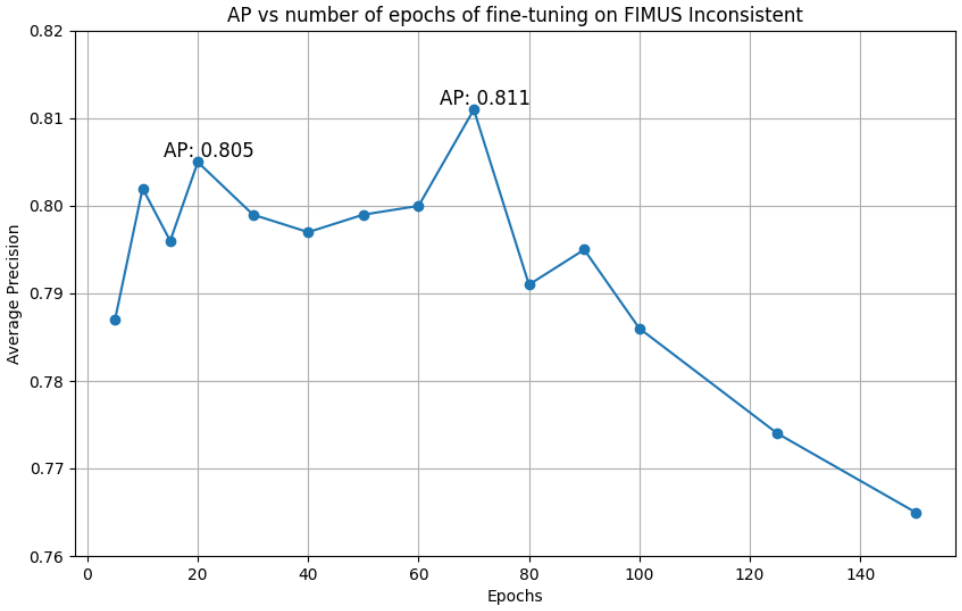


Figure 31: COCO APs Over Number of Epochs Fine Tuned on Inconsistent.

The following table (6) displays how well the models performed on the Consistent dataset.

Model	AP50	AP75	AP90	AP50-95
YOLOv9-m Not Fine-Tuned	0.957	0.908	0.768	0.865
YOLOv9-c Not Fine-Tuned	0.932	0.885	0.757	0.845
YOLOv9-e Not Fine-Tuned	0.972	0.949	0.912	**0.945**
YOLOv9 CrowdHuman 5ep	0.968	0.916	0.707	0.834
YOLOv9 CrowdHuman 10ep	0.964	0.914	0.730	0.831
YOLOv9 Inconsistent 20ep	0.950	0.850	0.662	0.805
YOLOv9 Inconsistent 70ep	0.954	0.867	0.657	0.811
YOLOv9 PRW 5ep	0.802	0.681	0.347	0.605
YOLOv9 PRW 10ep	0.969	0.856	0.269	0.721
YOLOv9 Football 5ep	0.214	0.143	0.143	0.143
YOLOv9 Football 10ep	0.286	0.190	0.143	0.176
YOLOv3	0.804	0.742	0.510	0.681
DETR-50	0.776	0.711	0.369	0.624
DETR-101	0.818	0.762	0.408	0.665

Table 6: COCO APs comparison of various models on Consistent (757 images).

These are disappointing results, showing no improvement in the models from implementing the fine-tuning. The models that were not fine-tuned performed best on the Consistent dataset, showing how freezing the backbone and fine-tuning the head on inconsistent, highly relevant data was a destructive practice in this case and did not improve the model accuracies.

Out of the fine-tuned models, the best performant one was the YOLOv9 model trained on the CrowdHuman dataset for 5 epochs. This is not unexpected: The dataset contains a lot of diversity and probably more instances of humans to learn from than the Inconsistent FIMUS dataset. The models that were fine-tuned on Inconsistent were trained for a tenfold more epochs to try to make up for this fact, but the accuracies failed to improve. These results indicate that rather than producing a sub-optimal specialized dataset for fine-tuning a model, the better option may be to use a larger and better dataset.

Should the positioning of the bounding boxes be of less importance, we see that fine-tuning on the PRW dataset is still a better option than fine-tuning on our FIMUS Inconsistent dataset partition. This is revealed by the AP50 score of 0.969 after just 10 epochs, while training on FIMUS Inconsistent did not achieve similar results in it's 150 epochs of training.

Further, we see that the models fine-tuned on the football player dataset were really bad-performing. They both completely missed the persons in the aquarium. This is likely largely due to the difference in scale, resulting in a fine-tuned model not able to predict the humans in the aquarium setting. A review of the labels indicate that the model had close to no inferences with a confidence score higher than 0.1, and the highest confidence at 0.425. An attempt was made to improve the scores by normalizing the detections, which resulted in the reported AP50-95 of 0.176 for the 10 epoch model, and 0.143 for the 5 epoch model. Prior to the confidence score adjustment, both models scored 0.0.

The fixed confidence threshold was raised for the DETR models since they report vastly higher confidence scores than many other model architectures. Therefore, the confidence scores were fixed at 0.99. A comparison of the inference latencies of the models were not conducted, as the YOLO model is still clearly the better option. For Table 7, the confidence threshold was varied similar to every other model.

Another surprising result from Table 6 is the YOLOv9-m performance relative to the larger weights of c and e! These letters correspond to different pre-trained weights, with various number of parameters. The converted version[31] of YOLOv9-e weight file has a size of 117.2MB, YOLOv9-c and -e fills only respectively 51.4MB and 40.7MB of space.

Vary-Both AP Figure 32 displays how the AP of the models varied with number of epochs. Equally to the COCO AP, the best model was the standard YOLOv9-e. We recall the Vary-Both AP references to the AP score achieved when evaluating models with a varying IoU threshold of 10 values from 0.50-0.95 *and* confidence threshold from 0.10 to 0.90 in 20 steps.

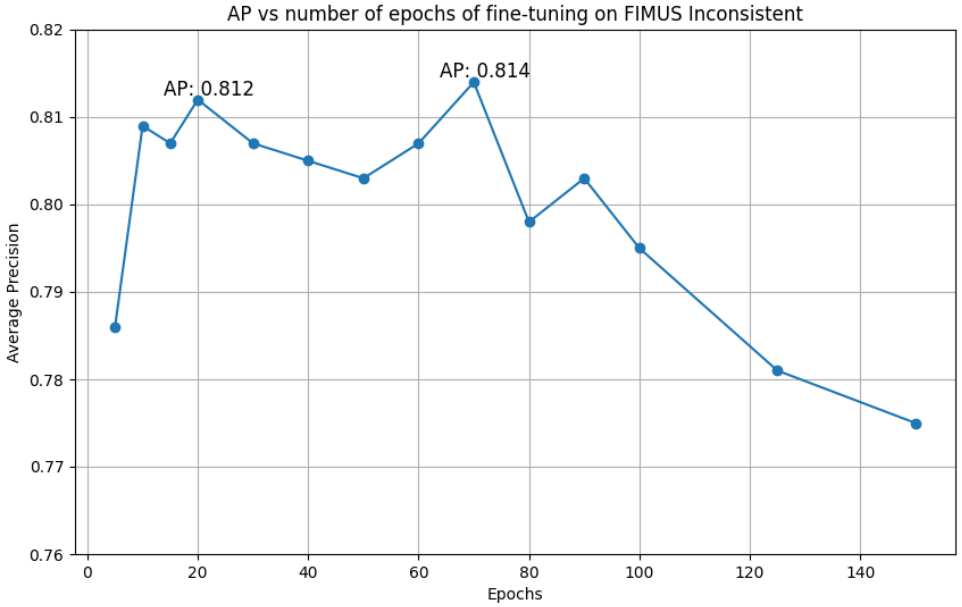


Figure 32: Vary-Both APs Over Number of Epochs Fine Tuned on Inconsistent.

The following table (7) displays how well the models performed on the Consistent dataset. Similarily to the results in Table 6, the YOLOv9-e was the most accurate, achieving an AP50-95 of 0.913.

[31]There's also converted vs not converted models. The reparameterization functionality to convert a model consists of trimming layers meant to speed up and augment model training, which is not needed for running model inference. The converted models achieve the same results but with lower inference latency, smaller size. They should not, however, be used for fine-tuning.

Model	AP50	AP75	AP90	mAP50-95
YOLOv9-m Not Fine-Tuned	0.895	0.855	0.746	0.821
YOLOv9-c Not Fine-Tuned	0.881	0.842	0.740	0.810
YOLOv9-e Not Fine-Tuned	0.942	0.918	0.876	**0.913**
YOLOv9 on Inconsistent for 20ep	0.948	0.854	0.680	0.812
YOLOv9 on Inconsistent for 70ep	0.948	0.867	0.674	0.814
YOLOv9 on PRW for 5ep	0.808	0.706	0.427	0.633
YOLOv9 on PRW for 10ep	0.924	0.792	0.252	0.677
YOLOv9 on CrowdHuman for 5ep	0.908	0.863	0.688	0.788
YOLOv9 on CrowdHuman for 10ep	0.944	0.902	0.740	0.822
YOLOv9 on Football for 5ep	0.221	0.177	0.175	0.179
YOLOv9 on Football for 10ep	0.514	0.457	0.443	0.452
YOLOv3	0.780	0.728	0.528	0.672
DETR ResNet50	0.414	0.355	0.164	0.315
DETR ResNet101	0.594	0.526	0.245	0.460

Table 7: APs Comparison of Various Models on Consistent (757 images), Varying Both Thresholds.

Results are similar to those in Table 6. See 4.1.4 for a brief presentation of the results. Note that for this evaluation, the DETR models were also varied from 0.1 to 0.90 in confidence thresholds, resulting in much worse average precisions.

DETR Fixed Confidence Thresholds' Effects on Average Precision The poor performances of the DETR models motivated the an investigation of the labels. The DETR models infer with much higher confidences, making the fixed threshold at 0.5 way too low to score well on the COCO metric. An assessment was made thus made to find the optimal confidence level for the model. This was found at 0.991, nearly doubling the AP score from 0.315 to 0.625 for the DETR with a ResNet50 backbone, and improving the score from 0.460 to 0.668 for the model with the more complex ResNet101 backbone.

Model	AP50	AP75	AP90	mAP50-95
DETR ResNet50 Conf 0.50	0.414	0.355	0.164	0.315
DETR ResNet50 Conf 0.95	0.755	0.660	0.326	0.585
DETR ResNet50 Conf 0.99	0.776	0.711	0.369	0.624
DETR ResNet50 Conf 0.991	0.773	0.713	0.373	0.625
DETR ResNet101 Conf 0.50	0.594	0.526	0.245	0.460
DETR ResNet101 Conf 0.95	0.795	0.719	0.353	0.627
DETR ResNet101 Conf 0.99	0.818	0.762	0.408	0.665
DETR ResNet101 Conf 0.991	0.818	0.767	0.419	0.668

Table 8: APs for DETR When Fixing the Confidence Threshold at Various Values (757 images).

The results presented in Table 8 are in accordance with the results of Carion et al. They claim it is performing well on panoptic segmentation, a task where pixel-level detail is important. This is the reason to the confidence values are high for the task of object detection. For the experiments in Table 6, a threshold of 0.99 was used instead of 0.991, because experimentally finding the optimal confidence threshold post-inference on a test-set is likely overfitted to the testing data and not so easy to optimize for in a practical setting.

4.2 The Broader Context of the Results

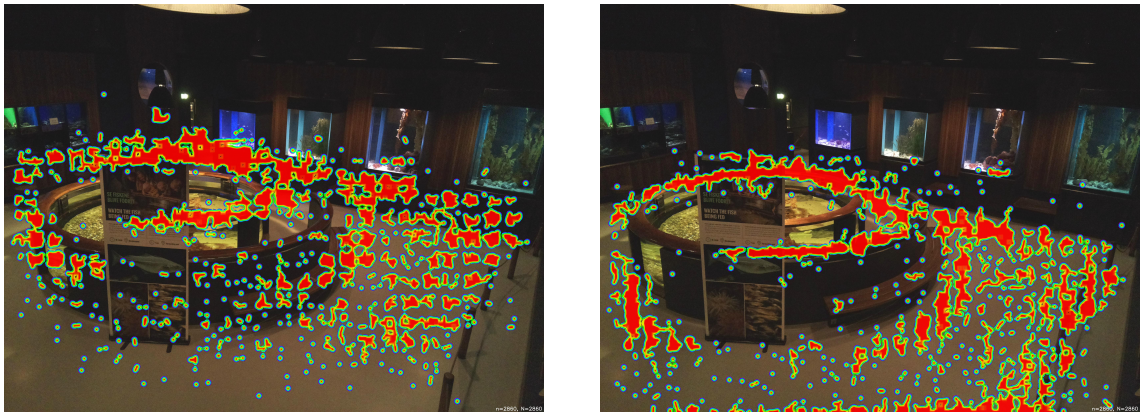
todo skrive om broader implications of the findings. Are these findings only relevant for the "snever" område of yolov9, what in it s architecture makes it so these resutls are interesting for all of object detection?

4.3 Data Visualization

The data may be visualized a multiple of ways. The explored methods in this thesis are by creating heat maps and bar charts to visualize the data.

4.3.1 Heatmaps

As mentioned in Section 3.8, heatmaps are a powerful visualization tool that can provide insights into visitor behavior patterns and engagement levels within a museum or aquarium setting. Heatmaps for the month of may are illustrated in 33. These heatmap generation code for the two heatmaps are identical, apart from one variable: the position where detections are mapped to. In 33a and b, the detections are mapped to the respectively the middle and the bottom center of the detection bounding box. This single modification has the largest difference on the edges of occlusions, such as (for the images in Figure 33) the railing of the fish tank in the center.



(a) Final Heatmap: Supervision, sampled from the *middle* of detections

(b) Final Heatmap: Supervision, sampled from the *bottom* of detections

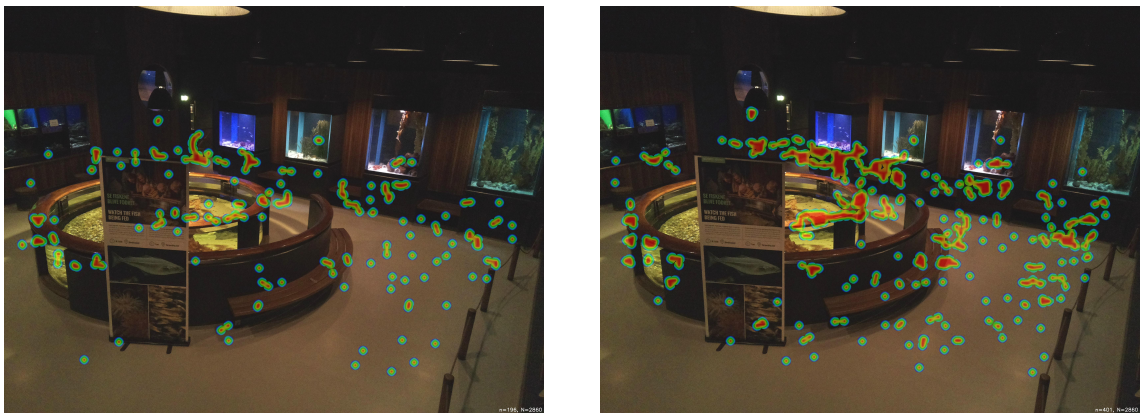
Figure 33: Final Heatmaps

There's another, more important takeaway from the small modification. While seemingly similar, the heatmap sampling detections from the middle of the bounding boxes (33a) reveal a weakness in our detector which is invisible in the other heatmap: the lamp is sometimes classified as a person. On the other hand, the other heatmap (33b) reveal another weakness. The seaweed in the second fish tank from the right is sometimes also classified as a person.

Apart from revealing weaknesses from the detector models[32], these heatmaps may provide valu-

[32]These weaknesses in our detector models could be revealed by looking at the annotated images. However,

able insights with regards to which areas of the facility are being used the most. There may be difficulties, however, in correctly inferring what are the reasons for the variations. For periods less than a day, these variations are likely due to randomness. The more interesting numbers in this context would be to see the total number of visitors throughout the day, which is better visualized in the bar charts in Section 4.3.2. Two heatmaps for separate days are illustrated in Figure 34.

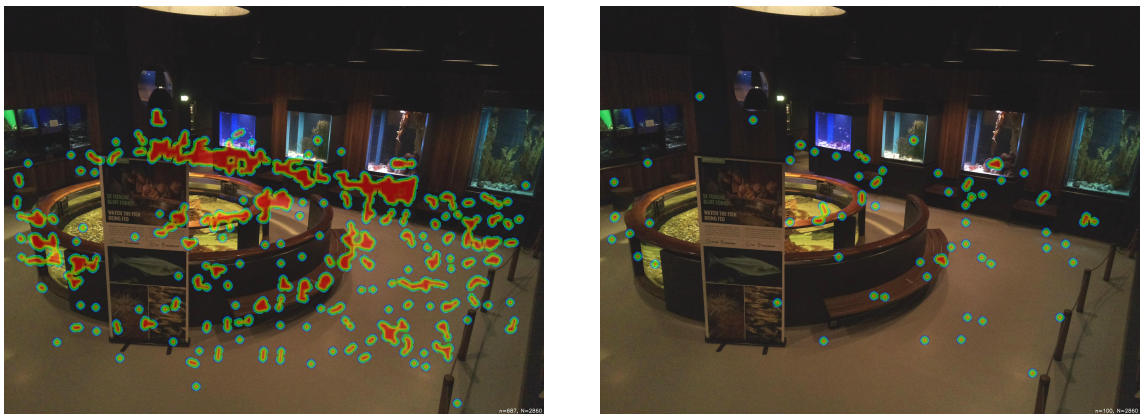


(a) Heatmap Wednesday 8th of May, 2024

(b) Heatmap Saturday 11th of May, 2024

Figure 34: Daily Heatmap

Heat maps may also visualize the hours throughout a day, accumulating all data for a specific time period each day to see if the visitor engagement changes based on the time. This is one example of introducing a variable, namely the time of day, to filter the detections. For an area where other variables such as the temperature, the noise level or the weather is also known, this could be used instead to filter the detections and illustrate how visitor engagement changes based on these factors. This usage would naturally, require some months-worth of data to be valid. For this project, only a months-worth of localization data has been stored to make the analysis. An illustration of heatmaps where the time of day has been used to determine which detections are presented in the heatmaps are displayed in Figure 35.



(a) Heatmap 13:00-14:00

(b) Heatmap 16:00-17:00

Figure 35: Hourly Heatmap

The heatmaps in Figure 35 reveal another use case for heatmaps. The relative difference between the two heatmaps is likely due to randomness, but with a larger number of detections one might be able to look for patterns. This could be that the heatmap for 13:00-14:00 could show a higher

looking at the annotated images is not possible for a on-device processing image-deleting device. In this case, one would need to display/plot the detections onto a base-layer image (heat map), or make use of obfuscation discussed in Section 2.3.7 to illustrate and reveal model weaknesses.

number of detections in front of the fish tanks, while the heatmap for 16:00-17:00 could show a higher number of detections on the benches. This could have easily been overlooked, had a manager of the museum only passed through the museum in the day and never in the evenings, resulting in him not thinking so many benches were neccessary.

4.3.2 Peak Hours

Another tool is to analyze the average number of detected persons per hour. This provides insights into room utilization during different times of the day.

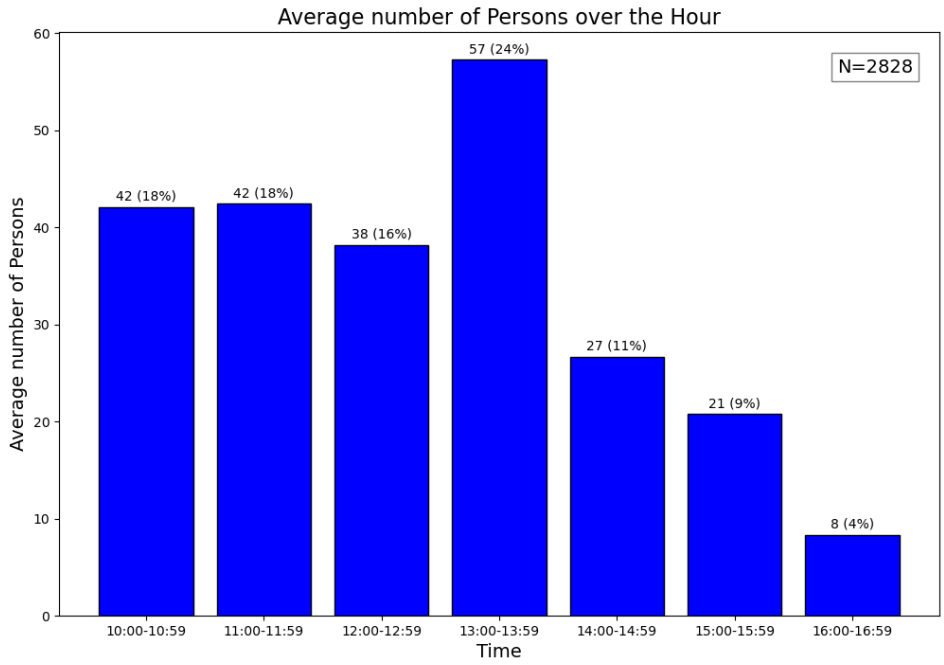


Figure 36: Peak Hours Analysis

For instance, a lower number of detections during early opening hours, despite high visitor entry, could indicate that the room temperature is not yet optimal, affecting visitor comfort. This hypothesis could be tested by using the number of visitors as the dependent variable and adjust the room temperature to see the effects. In the absence of other confounding variables[33], this could be an interesting causal relationsship to investigate to infer the perfect room temperature. However, due to the requirement in such an investigation for the high volume of data to rule out the possibility of randomness confounding the results, this investigation is likely unrealistic.

Further, comparing visitor detections of summer vs winter months, normalized for the total visitors in the facility, could provide deeper insights. It would enable the possibility of gauging the relative popularity of different areas. Indoor environments, typically maintained at constant temperatures, might offer different levels of comfort compared to the naturally fluctuating conditions of outdoor areas.

Understanding these dynamics can guide decisions on environmental controls, such as adjusting heating levels to enhance visitor comfort and potentially increase engagement in specific areas of the facility. Such adjustments could directly influence the overall visitation experience, making the

[33]Which may be a difficult and nearly impossible task in a real-world setting. Altough providing high ecological validity, such a setting makes it nearly impossible to infer valid results due to the high chance of unforeseen or even random events affecting the results.

whole facility more favorable for a visit regardless of seasonal factors.

5 Reflections & Overall Discussions

This chapter will discuss the broader implications of the results, summarizing the results and their significance for the development of similar systems, and whether or not the approach in this thesis is a viable solution for the presented problem. To maintain flow, the research questions from the introduction of this thesis are restated and addressed in the end of this discussion.

5.1 Visitor Behaviour Analysis

5.2 Privacy in Images

In the edge computing paradigm, the developer has the option to choose what should happen to the image post-analysis. Having the device delete the image and only send the anonymous analysis results is what makes the device the most privacy preservant, but obfuscation may be a preferable approach in some cases. While the former is advantageous when the technology is verified or a certain degree of analysis error is acceptable, the latter makes it easier to develop and verify the technology as one may to some degree verify the analysis results. Obscuring the images also has the advantages of transparency regarding the origin of analysis results, which in some cases proves beneficial as the end-users may want to verify and understand the data. Some approaches to obfuscating an image were displayed, where blurring the whole image certainly is the easiest option to avoid having to detect faces.

Edgcomb and Vahid reported blurred images to not be privacy preservant (Edgcomb and Vahid). However, one could argue the results of this study are invalid due to multiple reasons.

- The age of the paper. Opinions may have changed since the publication in 2012, i.e. people may be either more or less sceptical.
- Due to the way the questionnaire had multiple methods of preserving privacy where some were clearly more privacy preservant than the others, participants may have felt urged to select blurred videos to be privacy violating due to the fact they were much less privacy preservant than some others, albeit they were not really in fact privacy violating had they been evaluated by themselves.
- One could also argue about the demographic of participants, and that the results of the study would possibly not apply to Scandinavian countries where the general trust to public institutions is significantly higher than in the United States. This builds on a hypothesis that people in Scandinavian countries may be more inclined to trust obfuscation methods to protect privacy, which is yet to be explored.

This discourse will prove relevant in addressing the research question of how a system may mitigate privacy concerns, and in the primary objective of developing a privacy preservant system.

5.3 Third-party Services

As we've seen in Section 2.6, third-party services offer convenient solutions that may align perfectly with specific requirements for object detection systems, providing a quick and efficient path to implementation. There are potential drawbacks, however. These are listed below.

5.3.1 Drawbacks of Utilizing Third-Party Services

1. **Complete Control Over the System:** Developing your own application allows for full customization in terms of software architecture, data processing, and system integration. This total control facilitates the optimization of the system to meet specific performance and

operational requirements. In addition, a system built separately would have the benefit of being independent from the performance and existence of Roboflow.

2. **Data Privacy and Security:** On-device processing ensures that all data processing is kept on-device, enhancing data security and privacy. Roboflow offers local deployment, but this comes as part of their more expensive business-level subsxription plan.

3. **Cost Efficiency:** Managing your own system can be more cost-effective in the long run, particularly if the application demands extensive processing power or high throughput, as it eliminates recurring costs associated with third-party platforms. Roboflow's plans include costs related to "inference credits", making the system great for small applications but less likely to be a good fit for bigger enterprise solutions looking to leverage the margins. GPT4-V may be accessed via Azure's OpenAI service, which is also priced by how much the service is used and how

4. **Performance Optimization:** Owning the inference system allows for hardware and software optimizations that are not possible when using third-party services. This can lead to better performance, especially in terms of processing speed and latency.

5. **Scalability and Integration Flexibility:** Implementing your own solution allows for easier scaling and integration with existing IT infrastructure, which is beneficial for maintaining seamless data workflows and supporting business growth without being limited by external platform constraints.

While leveraging third-party services can expedite development, it is imperative for researchers and practitioners in the field of object detection, particularly in contexts such as person detection where privacy may be of cencern, to carefully weigh these considerations. Exploring alternative methods of implementation, including developing systems from scratch, can offer greater flexibility, control, and potential for innovation.

5.4 On the Ethicality of Person Localization Systems Development

History has shown Kant's categorical imperative to function as a guiding principle in smaller groups, but these principles fail once the size of societies passes a certain threshold where internal in- and out groups are forming. This is apparent, as wars and failure of governments to provide basic humanitary aid to those in need is still an issue. Enabling further mass public control through automated human localization devices may be a bad idea. This is also the case in areas where conflicts are not the current issue, but ways of tipping an election through improved intelligence or influence on the mass public may be the threat.

People may use Kant's deontological ethics to argue why they don't want systems to gather intelligence on them, without being able to accurately reflect about the consequences; posing arguments such as "it is just wrong" or similar. As seen in Section 2.1.3, this principle will work until the convenience of devices surpass the desire to preserve privacy. This constitutes a need for a stronger motivation to uphold the individual privacy, once technology capabilities surpass these desires. We already agree to surveillance in many public spaces for surveillance reasons. We have also accepted having devices in our homes and pockets listening for a "Hi Siri", "Okay Google", or "Alexa". The future may also include devices that will locate you in your room, detect if you fall, or turn the TV on once you move into the living room area. Not to speak about the attention-encapsulating experience these intelligent automations may entail, they would also pose a threat to privacy in the cases where the companies have legal bases for processing of personal data where "it is necessary for the performance of a contract to which the data subject is a party" (see Section 2.2.1).

Redmon's quote "[...] I bought in to the myth that science is apolitical and research is objectively moral and good no matter what the subject is." reflects a deontological approach to ethics, which may fail to consider the impacts of actions.

As proven by the third-party services and products, however, there are also many positive impacts of computer vision applications, e.g. self-driving cars, automated fall detection, and automated

waste filtering for recycling. Democracy is designed so it is up to the people themselves to decide whether the benefits outweigh the risks. To put it in utilitarian terms: will the overall happiness or utility be improved or worsened by the development of said systems and applications?

Drawing from the literature on utilitarianism and ethical consequences, it is evident that the deployment of these technologies necessitates a balanced and comprehensive approach. Utilitarianism suggests that we must weigh the potential benefits of increased safety, efficiency, and convenience against the risks to privacy and individual freedoms. The key challenge for developers and policymakers is to maximize the net positive impact—enhancing societal well-being while minimizing adverse outcomes. This perspective underscores the importance of rigorous impact assessments and ethical considerations in the development and implementation phases of these technologies.

Modern philosophical discourse, as highlighted by thinkers like Harris, Russell, and Yudkowsky, also offers critical insights, particularly regarding the broader ethical and regulatory challenges posed by AI. Although their primary concerns revolve around the long-term implications and governance of artificial general intelligence, their emphasis on the need for informed and effective regulation is highly pertinent. While ethical considerations are crucial, it's important not to stifle innovation.

In conclusion, the ethical deployment of computer vision technologies requires a multi-faceted approach. Utilitarian principles call for maximizing societal benefits while minimizing harms, and modern philosophical discourse on AI regulation highlight the necessity for informed policymaking. By integrating these diverse ethical frameworks, society can better navigate the complexities of technological advancement, fostering developments that are both beneficial and just.

5.4.1 Broader Impacts

The practical demonstation of basic-level computer vision technology in the project of this thesis, plays a tiny role in advancing technology, while the thesis addresses critical ethical and privacy concerns in a much more comprehensive and possibly impactful way. The increased efficiency and convenience[34] outweighs potential privacy infringements in the live aquarium experiment of the project. This ethical consideration aims to convey the message of the need for a thoughtful and comprehensive approach in developing and implementing technology. Developers bear a significant responsibility to ensure ethical performance, as the public often fails to recognize these issues and regulations lag behind.

The rapid pace of AI development necessitates that regulators possess a nuanced understanding of the technologies they seek to govern, ensuring that laws and policies are both protective and conducive to innovation. Developers should be encouraged to pursue creative solutions while adhering to ethical guidelines. A balance between innovation and ethics must be achieved through collaborative efforts between technologists, ethicists, policymakers, and the public, and this thesis may serve as a collaborative and communicative medium for this.

By upholding the aforementioned ethical standards and evaluating the potential outcomes of the project, we can justify the development of the system in this thesis project. Recognizing that the benefits to societal advancement and the contributions to ethical discourse make such progress acceptable and valuable.

5.5 Methodology

Despite it's Apache-2.0 license, the DETR showed too bad performance in object detection to be part of a satisfactory system.

[34]Hopefully also safety in the future. Also not mentioned is the personal desire to complete the master thesis degree.

5.6 Results

5.7 Research Questions

1. What are some privacy risks associated with traditional human localization systems in public spaces, and how may a system mitigate these privacy concerns?

   Privacy risks include unauthorized data collection, misuse of personal data, and lack of transparency in data handling. On-device processing mitigates these concerns by keeping data localized, enhancing security and privacy, and allowing users greater control over their data.

2. How does the validity of object detection model evaluations change when using data specifically from the intended deployment environment compared to using generic datasets?

   The validity of object detection model evaluations improves when using data from the intended deployment environment because the model can better learn the specific features and variations present in that context. Generic datasets may not capture these nuances, leading to less accurate evaluations and performance.

3. What are some machine learning architectures suitable for object detection in a real-world deployment scenario?

4. How many images are needed for testing before the average precision of a machine learning model converges in a single-environment setting?

6 Conclusion

Summarization of the thesis and its contributions to the field.

7 Future Work

Recommendations for future work include the following.

7.1 Data Visualization Tools

7.1.1 Heat Map Generation with more Variables

Other variables could have been integrated in the solution to create more insights. This could be temperature, weather or light data, measuring if the visitation patterns change based on any of these variables. Once the initial technology of being able to localize humans in a space in a privacy preservant way, experimenting with more data variables to improve insights could prove beneficial.

7.1.2 Zones

Another great visualization would be to visualize the number of detections in front of each exhibition. This would allow for an easy-to-grasp and useful visualization of where visitors most frequently stand, giving indications to what exhibition is most popular (or time-consuming).

7.1.3 Queue Formation Areas

Bibliography

Academic Resource Center, University of California Riverside. (2024). Formatting long/block quotes [Accessed: 2024-05-29]. https://arc.ucr.edu/writing/formatting-longblock-quotes (cit. on p. 7).

American Psychological Association. (2022). Title case capitalization [Accessed: 2024-05-29]. https://apastyle.apa.org/style-grammar-guidelines/capitalization/title-case (cit. on p. 7).

Antunes, R. S., André da Costa, C., Küderle, A., Yari, I. A., & Eskofier, B. (2022). Federated learning for healthcare: Systematic review and architecture proposal. *13*(4). https://doi.org/10.1145/3501813 (cit. on pp. 15, 16).

Author(s). (2023). Transfer learning with generative models for object detection on limited datasets. *arXiv*. https://arxiv.org/abs/2402.06784 (cit. on p. 31).

Blum, A., Ligett, K., & Roth, A. (2011). A learning theory approach to non-interactive database privacy (cit. on p. 16).

Boesch, G. (2023). *Yolov7: The most powerful object detection algorithm (2024 guide)*. https://viso.ai/deep-learning/yolov7-guide/ (cit. on p. 35).

Brandeis, L. D., & Warren, S. (1890). The right to privacy. *Harvard Law Review*, *4*(5), 193–220. https://doi.org/10.2307/1321160 (cit. on p. 13).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. https://doi.org/https://doi.org/10.48550/arXiv.2005.12872 (cit. on pp. 29, 59).

Chhabria, P. (2022). *Expert a 1m53s 16x9*. https://www.youtube.com/watch?v=w78U7w33NTI&t=5s (cit. on p. 34).

Dictionary, O. E. (2023, July). Gamification (n.) https://doi.org/10.1093/OED/7320229446 (cit. on p. 23).

Dwork, C., McSherry, F., Nissim, K., & Smith, A. D. (2006). Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, 265–284 (cit. on p. 16).

Dwork, C., McSherry, F., Nissim, K., & Smith, A. D. (2011). Differential privacy - a primer for the perplexed. https://api.semanticscholar.org/CorpusID:2583736 (cit. on p. 16).

Edgcomb, A., & Vahid, F. (2012). Privacy perception and fall detection accuracy for in-home video assistive monitoring with privacy enhancements. *SIGHIT Rec.*, *2*(2), 6–15. https://doi.org/10.1145/2384556.2384557 (cit. on pp. 19, 20, 63).

Elias, A. R., Golubovic, N., Krintz, C., & Wolski, R. (2017). Where's the bear? - automating wildlife image processing using iot and edge cloud systems. *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 247–258. https://ieeexplore.ieee.org/document/7946882 (cit. on pp. 18, 19).

Eufy. (2022). *What makes eufycam 3c stand out?* https://eu.eufy.com/pages/security-eufycam3c (cit. on p. 34).

European Parliament and Council of the European Union. (2022). Directive (eu) 2022/2555 of the european parliament and of the council of 14 december 2022 on measures for a high common level of cybersecurity across the union, amending regulations (eu) no 910/2014 and (eu) 2016/679 and directive (eu) 2018/1972 and repealing directive (eu) 2016/1148 [Accessed: 2024-03-15]. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022L2555 (cit. on p. 12).

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. https://doi.org/10.1007/s11263-009-0275-4 (cit. on p. 22).

Fischer-Hbner, S., & Berthold, S. (2017). Chapter 53 - privacy-enhancing technologies. In J. R. Vacca (Ed.), *Computer and information security handbook (third edition)* (Third Edition, pp. 759–778). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-803843-7.00053-3 (cit. on p. 13).

Gu, R., Niu, C., Wu, F., Chen, G., Hu, C., Lyu, C., & Wu, Z. (2021). From server-based to client-based machine learning: A comprehensive survey. *ACM Comput. Surv.*, *54*(1). https://doi.org/10.1145/3424660 (cit. on p. 15).

Huang, Y., Li, Y. J., & Cai, Z. (2023). Security and privacy in metaverse: A comprehensive survey. *Big Data Mining and Analytics*, *6*(2), 234–247. https://doi.org/10.26599/BDMA.2022.9020047 (cit. on pp. 14, 16).

Huang, Z., Yang, S., Zhou, M., Gong, Z., Abusorrah, A., Lin, C., & Huang, Z. (2022). Making accurate object detection at the edge: Review and new approach. *Artificial Intelligence Review*, *55*(3), 2245–2274. https://doi.org/10.1007/s10462-021-10059-3 (cit. on p. 17).

James Gallagher, P. S. (2024). How to train yolov9 on a custom dataset [Accessed: 2024-05-21]. *Roboflow Blog*. https://blog.roboflow.com/train-yolov9-model/ (cit. on p. 50).

Lanir, J., Kuflik, T., Sheidin, J., Yavin, N., Leiderman, K., & Segal, M. (2017). Visualizing museum visitors' behavior: Where do they go and what do they do there? *Personal and Ubiquitous Computing*, *21*(2), 313–326. https://doi.org/10.1007/s00779-016-0994-9 (cit. on p. 10).

Leo Ueno, T. L. (2024). GPT-4o: The comprehensive guide and explanation [Accessed: 2024-05-23]. *Roboflow Blog*. https://blog.roboflow.com/gpt-4o-vision-use-cases/ (cit. on p. 33).

Li, Q., Niaz, U., & Merialdo, B. (2012). An improved algorithm on viola-jones object detector. *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 1–6. https://doi.org/10.1109/CBMI.2012.6269796 (cit. on p. 27).

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). Microsoft coco: Common objects in context. (Cit. on p. 45).

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P., & Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, *abs/1405.0312*. https://doi.org/10.48550/arXiv.1405.0312 (cit. on pp. 22, 27).

Luccioni, S. (2023). Ai is dangerous, but not for the reasons you think author [Accessed: 2024-05-17]. https://www.youtube.com/watch?v=eXdVDhOGqoE&ab_channel=TED (cit. on p. 23).

Ma, C., Shimada, A., Uchiyama, H., Nagahara, H., & Taniguchi, R.-i. (2019). Fall detection using optical level anonymous image sensing system. *Optics & Laser Technology*, *110*, 44–61. https://doi.org/10.1016/j.optlastec.2018.07.013 (cit. on p. 19).

Maayah, M., Abunada, A., Al-Janahi, K., Ahmed, M. E., & Qadir, J. (2023). Limitaccess: On-device tinyml based robust speech recognition and age classification. *Discover Artificial Intelligence*, *3*(1), 8. https://doi.org/10.1007/s44163-023-00051-x (cit. on p. 73).

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, *1*(1), 3–es. https://doi.org/10.1145/1217299.1217302 (cit. on p. 14).

Murali, N. (2021). *Image classification vs semantic segmentation vs instance segmentation*. https://nirmalamurali.medium.com/image-classification-vs-semantic-segmentation-vs-instance-segmentation-625c33a08d50 (cit. on p. 3).

Neuman, S. M., Plancher, B., Duisterhof, B. P., Krishnan, S., Banbury, C., Mazumder, M., Prakash, S., Jabbour, J., Faust, A., de Croon, G. C., & Reddi, V. J. (2022). Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots. *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 296–299. https://doi.org/10.1109/AICAS54282.2022.9870000 (cit. on pp. 4, 73).

OpenCV. (2022). Intersection over union (iou) in object detection and segmentation [Accessed: 2024-05-29]. https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/ (cit. on p. 26).

Orwell, G. (1949). *1984*. Secker; Warburg. (Cit. on p. 20).

Park, J., Chen, J., Cho, Y. K., Kang, D. Y., & Son, B. J. (2020). Cnn-based person detection using infrared images for night-time intrusion warning systems. *Sensors*, *20*(1). https://doi.org/10.3390/s20010034 (cit. on p. 30).

Pérez Cortés, L. E., Ha, J., Su, M., Nelson, B., Bowman, C., & Bowman, J. (2023). Gleaning museum visitors' behaviors by analyzing questions asked in a mobile app. *Educational technology research and development*, *71*(3), 1209–1231. https://doi.org/10.1007/s11423-023-10208-1 (cit. on p. 10).

Rajapakse, V., Karunanayake, I., & Ahmed, N. (2023). Intelligence at the extreme edge: A survey on reformable tinyml. *ACM Comput. Surv.*, *55*(13s). https://doi.org/10.1145/3583683 (cit. on pp. 4, 73).

Ravi, S., Climent-Pérez, P., & Florez-Revuelta, F. (2023). A review on visual privacy preservation techniques for active and assisted living. *Multimedia Tools and Applications*, *Online First*, 1–16. https://doi.org/10.1007/s11042-023-15775-2 (cit. on p. 14).

Redmon, J. (2020a). Joseph redmon's twitter [Accessed: 2024-05-09]. *Twitter*. https://twitter.com/pjreddie/status/1230524770350817280?lang=en (cit. on p. 21).

Redmon, J. (2020b). Joseph redmon's twitter [Accessed: 2024-05-09]. *Twitter*. https://twitter.com/pjreddie/status/1230524770350817280?lang=en (cit. on p. 21).

Sandtrø, J. (2022). *Webcast: Hvordan unngå å bryte regelverket (gdpr)*. SuperOffice Norge on Youtube. https://www.youtube.com/watch?v=FB2P-ijCIKw&ab_channel=SuperOfficeNorge (cit. on p. 12).

Sasagawa, Y., & Nagahara, H. (2020). Yolo in the dark - domain adaptation method for merging multiple models. *Proceedings of the European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-030-58589-1_21 (cit. on p. 30).

Saurav, S., Saini, A. K., Saini, R., & Singh, S. (2022). Deep learning inspired intelligent embedded system for haptic rendering of facial emotions to the blind. *Neural Computing and Applications*, *34*(6), 4595–4623. https://doi.org/10.1007/s00521-021-06613-3 (cit. on p. 22).

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2018). Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123* (cit. on p. 46).

Sharma, P. (2023). *Role of weight transmission protocol in machine learning*. https://www.tutorialspoint.com/role-of-weight-transmission-protocol-in-machine-learning (cit. on p. 16).

Sweeney, L. (2002). K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570. https://doi.org/10.1142/S0218488502001648 (cit. on p. 14).

Termly. (2023). Natural person [Accessed: 2024-05-15]. https://termly.io/legal-dictionary/natural-person/ (cit. on p. 13).

The European Parliament. (2016). *Eu directive 2016/679 general data protection regulation (gdpr)*. Official J Eur Union 2014. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679 (cit. on p. 11).

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I. https://doi.org/10.1109/CVPR.2001.990517 (cit. on p. 27).

Wang, C., Yeh, I.-H., & Liao, H.-Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information (cit. on pp. 29, 30).

Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly Fall Detection Systems: A Literature Survey. *Frontiers in Robotics and AI*, *7*. https://doi.org/10.3389/frobt.2020.00071 (cit. on pp. 13, 18).

Wei, C., Wu, G., & Barth, M. J. (2024). Feature corrective transfer learning: End-to-end solutions to object detection in non-ideal visual conditions. *arXiv preprint arXiv:2404.11214*. https://arxiv.org/abs/2404.11214 (cit. on p. 31).

Westin, A. (1967). *Privacy and freedom*. Atheneum. (Cit. on p. 13).

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, *10*(2). https://doi.org/10.1145/3298981 (cit. on p. 16).

Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., & Tian, Q. (2017). Person re-identification in the wild. (Cit. on p. 46).

Zheng, S., Apthorpe, N., Chetty, M., & Feamster, N. (2018). User perceptions of smart home iot privacy. *Proc. ACM Hum.-Comput. Interact.*, *2*(CSCW). https://doi.org/10.1145/3274469 (cit. on p. 11).

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, *111*(3), 257–276. https://doi.org/10.1109/JPROC.2023.3238524 (cit. on pp. 18, 23, 27, 28).

A Code Snippets

```
class CameraHandler:
    def __init__(self, awb_gains, awb_mode, brightness, contrast, exposure_compensation,
    exposure_mode, image_format, iso, meter_mode, resolution, sensor_mode, shutter_speed, framerate)
    :
        # Create camera with the arguments
        cam = picamera.PiCamera(resolution=resolution, sensor_mode=sensor_mode,
        framerate=framerate)
        cam.iso = iso

        # Wait for the automatic gain control to settle
        time.sleep(2)
        cam.shutter_speed = shutter_speed
        cam.exposure_mode = exposure_mode
        cam.awb_mode = awb_mode
        cam.awb_gains = awb_gains
        cam.exposure_compensation = exposure_compensation
        cam.brightness = brightness
        cam.contrast = contrast
        self.image_format = image_format
        cam.meter_mode = meter_mode
        self.picamera = cam
```

Figure 37: CameraHandler class initialization.

B Technical Challenges

Pi out of memory on image capture Context: tweaking camera settings, leading to larger size images Error: mal port enable failed to enable connected port... Out of resources. Solution: allocating more memory to the GPU by going through raspi-config Performance Options -¿ GPU Memory

AWB Gains Context: setting picamera.awg gains has no effect Error: no change Solution: set awb gains after awb mode has been set to off, and capture an image. The control seems to not set before after capturing an image. Thus, setting these values and then checking the values, it might seem the modification has not been made altough it will show on the images.

C Camera Settings Explanation

- *awb_gains* - Set the auto white balance gains red and blue. Set as a (red, blue) set. Each value may range from 0.0 to 8.0. Typical is 0.9-1.9. Only has an effect when *awb_mode* is 'off'. IMPORTANT: awb and exposure mode must be set to off BEFORE setting the *awb_gains*.
- *awb_mode* - Auto white balance. Default is auto. Disabling auto white balance mode allows for manual setting of AWB gains, ensuring consistent image color temperature. 'off' 'auto' 'sunlight' 'cloudy' 'shade' 'tungsten' 'fluorescent' 'incandescent' 'flash' 'horizon'
- *brightness* - Adjusts the post-processing brightness of the image. Default is 50, representing no adjustment. 0 to 100.
- contrast - Adjusts the post-processing contrast of the image. Default is 0, representing no adjustment. -100 to 100.
- *exposure_compensation*- Adjusts the exposure compensation level. Range is -25 to 25. Default is 0.

- *exposure_mode* - Disabling auto-exposure allows for manual control over exposure settings. 'off' 'auto' 'night' 'nightpreview' 'backlight' 'spotlight' 'sports' 'snow' 'beach' 'verylong' 'fixedfps' 'antishake' 'fireworks'.
- *exposure_speed* - Indicates the effective exposure speed, which may differ from the set shutter speed after adjustments.
- *framerate* - Sets the number of frames per second captured by the camera.
- *iso* - Sets the ISO sensitivity of the camera sensor. Values: 100, 200, 320, 400, 500, 640, 800. 0 is auto.
- *metering_mode* - Sets the metering mode. 'average' 'spot' 'backlit' 'matrix'. Backlit is the largest area. Default is average.
- *sensor_mode* - Controls the sensor mode, where '3' typically corresponds to standard image capturing.
- *shutter_speed* - Sets the shutter speed in microseconds. 0 to 6000000. Default 0. 0 is auto. Max 6s.
- *resolution* - Sets the resolution of the image frame.

D TinyML and Frugal Devices

As mentioned in Section 1.3, TinyML is when machine learning models are aimed at deployment to heavily resource constrained environments, e.g. what is called frugal devices. These are devices where the microcontroller units (MCUs) are accompanied by memory measured in kilobytes, and processor speeds measured in megahertz.

Machine learning networks applied to tiny robots are subject to challenges from size, weight, area, and power (SWAP) (Neuman et al., 2022). Many of the same challenges apply even in applications where the SWAP challenges are not the main concerns. Rajapakse et al. mentions the open challenges and future directions of the next generation tinyML. Catastrophic forgetting, which is when information from previous tasks while learning new ones are forgotten, are a result of the frugal devices' computational resources and memory. The first recommendation for future directions from the authors is to investigate fog computing as a means to offload tasks from the frugal devices.

Maayah et al. (2023) explore the ways of speech processing on microcontrollers to improve car AI systems. They employed their trained and optimized model to an Arduino Nano 33 BLE. The model achieved accuracies in above 85 percent on recognizing whether the voice was that of an adult or a child, and to detect whether the speech was a replay (synthetic) or "live".

Furthermore, Rajapakse et al. discuss some of the challenges in industrial IoT environments with several smart object devices, where having the devices share a collective dataset of anomalies within a manufacturing environment would be advantageous for utilizing collective learning to improve the ML models in each of the devices (2023). This means the devices will all learn from observations of the other devices, such that the training period from when a network of devices is deployed within a new environment to when they are fully functioning with regards to accuracy in their predictions is reduced. See more about this in section 2.3.2 about federated learning as a way of implementing a collective learning network for the edge devices.