NTNU
Kunnskap for en bedre verden

Department of Mechanical and
Industrial Engineering

Master's Thesis - robotics & automation

On-Device Object Detection and Dataset Authenticity: Enhancing Privacy and Real-World Applicability

Other drafts:
The Impact of Dataset Characteristics on the Real-World Accuracy of Privacy-Preserving Human Detection Systems
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

1 Introduction **1**

1.1 Background and Motivation . . . 1

1.2 Scope . . . 2

1.2.1 Research Questions . . . 3

1.2.2 Research Objectives . . . 3

1.3 Structure . . . 4

2 Literature Review **5**

2.1 Visitor Behavior Analysis . . . 5

2.1.1 Questions asked by visitors in a mobile app . . . 5

2.1.2 Percieved value to museum stakeholders . . . 5

2.2 Individual Privacy . . . 5

2.2.1 The General Data Protection Regulation . . . 6

2.2.2 Preservation of Individual Privacy . . . 8

2.2.3 User perceptions of smart home IoT privacy . . . 9

2.2.4 Federated learning . . . 10

2.2.5 Differential privacy . . . 11

2.2.6 On-device processing . . . 12

2.2.7 Depth cameras . . . 13

2.2.8 Deletion of images . . . 13

2.2.9 Obfuscation . . . 14

2.2.10 Ethical Considerations in the Development of Localization Technologies . . 15

2.3 Object Detection . . . 16

2.3.1 Performance benchmark . . . 17

2.3.2 Real-Time Detection Transformer . . . 21

2.3.3 YOLOv9 . . . 22

2.3.4 Dark-Lit Environments . . . 22

2.3.5 Transfer Learning and the Effectiveness of Fine-tuning . . . 22

2.4 Third-Party Services . . . 23

2.4.1 Roboflow . . . 23

2.4.2 GPT-4 with Vision on Object Detection . . . 23

2.5 Third-Party Products . . . 24

2.5.1 Eufy . . . 24

2.5.2 Aqara . . . 25

2.5.3 i-PRO . . . 25
2.5.4 Viso . . . 26
2.5.5 VMukti . . . 26
2.6 Summary of Literature Review . . . 26

3 Methodology **27**
3.1 Project . . . 27
3.2 The FIMUS Dataset . . . 27
3.2.1 Camera Configurations . . . 27
3.2.2 The Image Capturing Process . . . 28
3.2.3 Labeling . . . 30
3.3 External Datasets . . . 31
3.3.1 Common Objects in Context (COCO) . . . 31
3.3.2 CrowdHuman . . . 31
3.3.3 Person Reidentification in the Wild . . . 32
3.4 Model Training . . . 32
3.4.1 Transfer Learning . . . 32
3.4.2 Hyperparameter Tuning . . . 32
3.5 Model Evaluation . . . 33
3.6 Ethical Considerations . . . 33
3.6.1 Privacy by Design . . . 33
3.6.2 Ethical Use and Data Protection . . . 34
3.6.3 Transparency and Accountability . . . 34
3.7 Heatmaps . . . 34
3.7.1 Supervision Heatmaps . . . 35

4 Results **36**
4.1 Machine Learning Models of the Project . . . 36
4.1.1 YOLOv9 . . . 36
4.2 Model Evaluation . . . 37
4.3 Data Visualization . . . 38
4.3.1 Heatmaps . . . 38
4.3.2 Peak Hours . . . 39
4.4 Third-party Services . . . 40
4.4.1 Drawbacks of Utilizing Third-Party Services . . . 41

5 Conclusion **42**

5.1 Disclaimers 42

5.1.1 Plagiarism 42

5.1.2 The Use of AI Tools 42

5.1.3 Privacy of Similar Projects 43

6 Future Work **44**

6.1 Heat Map Generation with more Variables 44

Bibliography **45**

A Code Snippets **48**

B Technical Challenges **48**

C Camera Settings Explanation **48**

D TinyML and Frugal Devices **49**

Abstract

Placeholders:
What
Why
How
Principal contributions
Principal conclusion

1 Introduction

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 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.

1.1 Background and Motivation

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.

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" 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. 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.

1.2 Scope

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 project spanned several disciplines and 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.

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.

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 to it performing worse than the YOLOv9 in multiple experiments. The object detectors are discussed in section 2.3.

Museum and Aquarium Opening Hours and Visitor Conduct The project was to not interfere with the normal operations of the aquarium. This meant the only hours to capture images for the dataset was within opening hours, and it meant not asking random visitors if they would be willing to participate in the project. By an early analysis of the visitation patterns, most visitors were there early in the morning from the opening at 10:00, until around 2 hours before closing time 17:00. This was the opportunity window for getting images collected for the dataset.

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 1 to display how they differentiate from object detection.

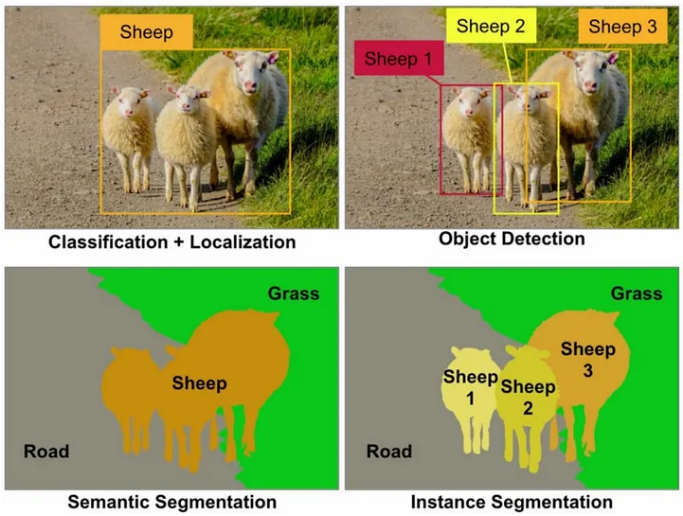


Figure 1: Image processing tasks (Murali, 2021).

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.

TinyML and Frugal Devices An initial attempt was made to encompass tinyML and frugal devices in the project. This, however, was scoped out of the thesis.

1.2.1 Research Questions

The following research questions are meant to guide the focus of this thesis.

1. What are some privacy risks associated with traditional human localization systems in public spaces, and how may on-device processing 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?

1.2.2 Research Objectives

The Primary Objectives are to:

1. Develop a privacy-preserving human localization system using on-device processing to minimize data transmission and enhance data privacy.
2. Evaluate the performance of the developed system in a real-world setting (e.g., an aquarium) to determine its effectiveness and reliability.
3. Investigate the effects of the dataset quality in fine-tuning of models on the performance.
4. Assess the impact of deployment-specific data on the accuracy and validity of object detection model evaluations, by comparing performance metrics with those obtained using generic datasets.

The Secondary Objectives are to:

1. Compare the privacy and performance impacts of on-device processing against traditional centralized methods.
2. Investigate the feasibility of deploying the developed system in other public spaces to enhance visitor analytics and security.
3. Develop a visualization tool to analyze and interpret the collected data for practical applications.
4. Explore relevant object detection architectures to evaluate their performance in a real-world deployment sceneario.

1.3 Structure

The thesis is structured as follows:

Section 2: Literature Review - Surveys existing technologies and discusses the theoretical underpinnings of the project.

Section 3: Methodology - Details the technical methods and materials used in the project.

Section 4: Results and Discussion - Analyzes the data collected, evaluates the system's performance, and discusses the findings.

Section 5: Conclusions and Future Work - Summarizes the research contributions and outlines potential future research directions.

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, including visitor behaviour analysis, individual privacy, object detection, the influence of fine-tuning models on specialized data, and already existing third-party services for deploying similar systems.

2.1 Visitor Behavior Analysis

Some of the traditional methods of analyzing visitor behavior 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, we bring forth a study on the percieved value of visitor tracking to museum stakeholders.

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 behavior analysis, and its perceived value to museum curators, administrators and department heads (2017). Wearable RFID trackers[1] 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 behavior 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 behavior 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 behavior would be to satisfy my curiosity." This contrast underscores the varied perspectives within museums regarding the value and implications of behavior analysis technologies.

2.2 Individual Privacy

This section is heavily influenced by previous work of the same author. See the disclaimers-section 5.1 for more details.

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

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 dimentions of privacy. The above quote would mainly cover informational privacy, which is most relevant to this thesis. The definition by Westin includes groups and institutions. However, in most legal systems, privacy is defined as a basic human right that only applies to natural persons[2]. The term "individual privacy"[3] 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, which are less relevant for this thesis include spatial privacy, territorial privacy, bodily privacy (Fischer-Hbner and Berthold, 2017).

The general data protection regulation (GDPR) is a single set of regulations to guarantee privacy and protecion of personal data. Informational privacy is mostly concerned with protection of personal data (we will define personal data in section 2.2.1), and informational privacy is what may be violated by the experiment of this thesis. Therefore, the GDPR is highly relevant for this thesis. Regardless, a quick review of the GDPR should be on the agenda of anyone affiliated with systems not inherently preservant of informational privacy.

2.2.1 The General Data Protection Regulation

This section serves as a summarization of some aspects of the GDPR relevant to the thesis.

The GDPR entered into applicability in the EU on 25th of May 2018 as a way for people to have more control over their data, and for having a level playing field for all companies. There is now one set of data protection rules for all companies operating in the European Economic Area (EEA). The EEA consists of all EU countries plus Iceland, Liechtenstein and Norway. The most relevant parts of the GDPR are the regulations regarding personal data.

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).

Regulations regarding personal data also applies to the events where pieces of information are aggregated to identify a person. It is possible to store information about individuals without it

[2] 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).

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

being personal data. This can be done in several ways, one of them being by the method of differential privacy. Differential privacy is explained in section 2.2.5.

Another approach to the personal data is to process it the right way.

Legal Bases for Processing Personal Data Processing of personal data is permissible under the GDPR only when it satisfies at least one of the following legal bases:

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 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 1) responsible for compliance with the 3 requirements summarized below, and 2) 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.

The 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 and information systems throughout the Union. 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.

2.2.2 Preservation of Individual Privacy

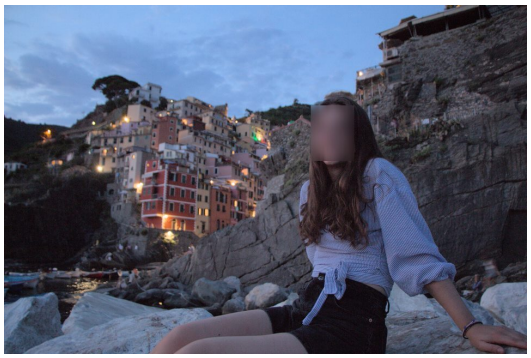
Preservation of individual privacy refers to maintaining the personal space and confidentiality of individuals, ensuring that their private lives and personal integrity are not invaded or exposed without consent. This involves considerations beyond just data, including behaviors and surveillance. Protection of personal data specifically 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. While protection of personal data is important due to the regulations, preserving the individual privacy is essential in object detection of persons.

There are multiple methods, both pre- and post-processing, for preserving individual privacy. 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.).

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 2, 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 (ma2007l-diversity). 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 2.2.5.

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

Figure 2: Six methods to enhance individual privacy in images.

There's also been much research with regards to preserving privacy in the context of machine learning (Ravi et al., 2023). Common to all use cases is the principle that deleting the data is the most definitive method of ensuring privacy, provided that such an action is feasible. When only non-personal data remains stored, the application is unequivocally secure in terms of privacy.

In addition to not protecting the identity of individuals, it is often important that the individuals *feel* their privacy is being preserved.

2.2.3 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 (Zheng et al.). What the researchers found, shows promise to IoT edge computing visual devices; the convenience and con-

nectedness of the devices surpasses the desire to preserve privacy. Another research question of the study 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 (Zheng et al., 2018)**
> "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).

2.2.4 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).

The concept of FL can be seen in figure 3, and is best described in the article of Antunes et al. (2022): "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). 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".

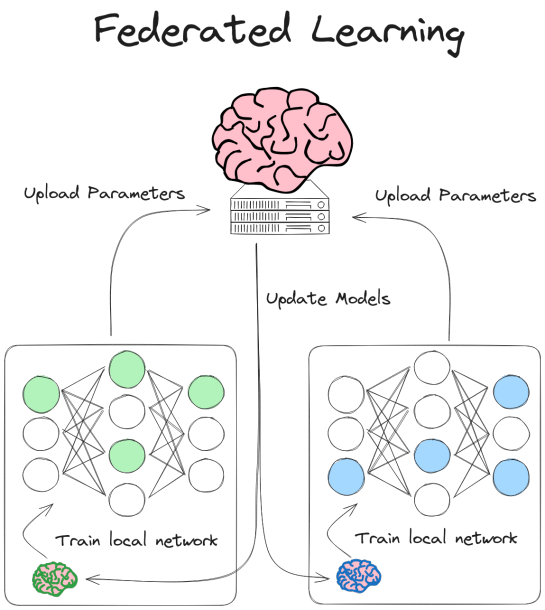


Figure 3: The federated learning process
Federated learning (FL), also known as collaborative learning, is a decentralized approach to training machine learning models. It doesn't require an exchange of data from client devices to global servers.

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[4] 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 2.2.5.

2.2.5 Differential privacy

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 4.

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.

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

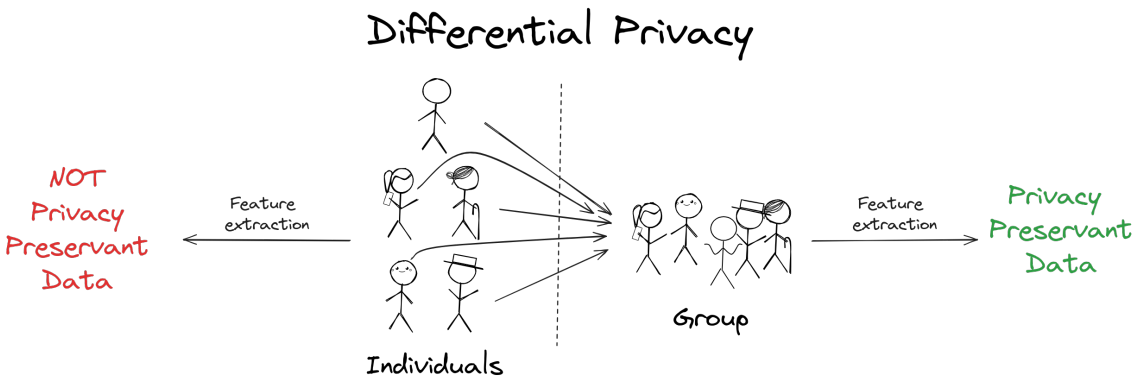


Figure 4: The differential privacy concept

2.2.6 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 2.2.9.

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 7). 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 5 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 1.2. Some considerations are, however, added in appendix D.

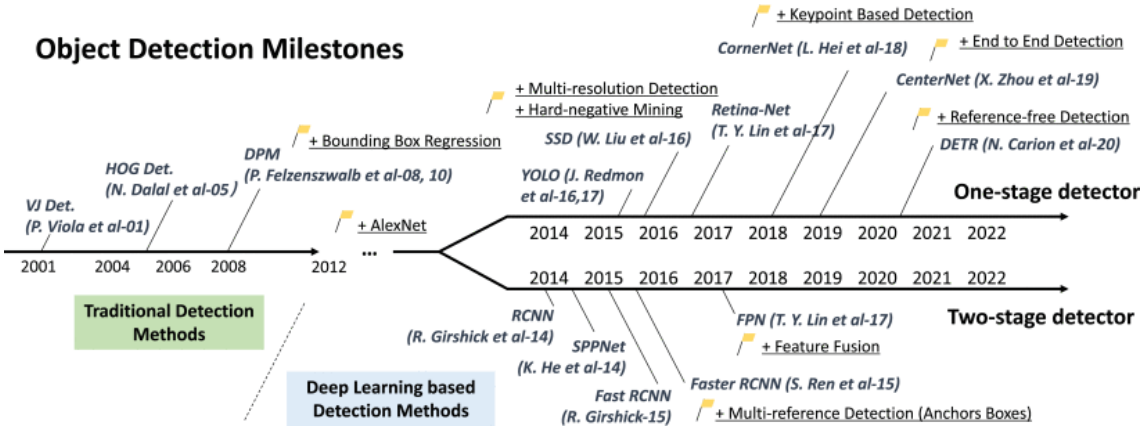


Figure 5: The evolution of object detection (Zou et al., 2023)

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 2.5.1), 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[5] 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.

2.2.7 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.

2.2.8 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.

[5]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.

2.2.9 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 2.

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[6] scale from 1-5. The obfuscation methods, or privacy enhancements as they called them, and the results are displayed in figure 6. 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.

[6]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.

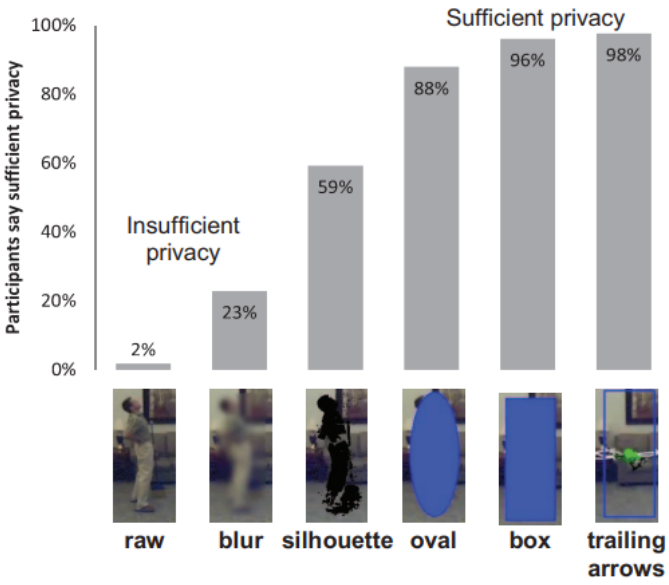


Figure 6: Privacy enhancements methods in the study of Edgcomb and Vahid (2012)
The study indicates that the methods of blurring an image is not perceived privacy preservant. However, as we will see in section 2.2.9, the results of this study might not be a reason by itself to throw the method of blurring an image out the window.

2.2.10 Ethical Considerations in the Development of 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.

1984 on greed of 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." -George Orwell, 1984.

Joseph Redmon 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 very much 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.-Joseph Redmon, Twitter (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.-Joseph Redmon, Twitter (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.-Joseph Redmon, Twitter (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? -Joseph Redmon, Twitter (Redmon, 2020b).

Ethical Framework for Development In developing technologies capable of tracking and analyzing human behavior, 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.

2.3 Object Detection

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 7. 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[7]. 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.

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

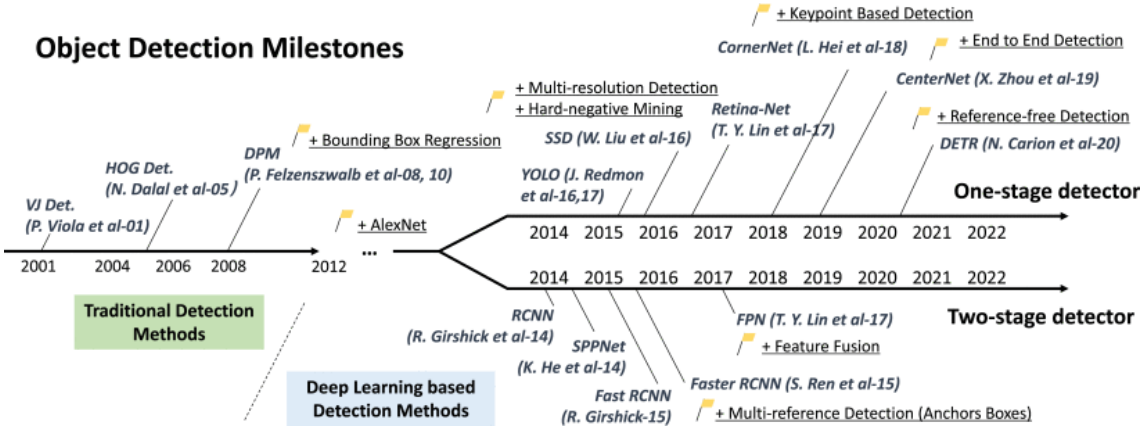


Figure 7: The evolution of object detection (Zou et al., 2023)

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). These are explored in greater detail in section 2.3.2 and 2.3.3.

Machine learning can be seen as a gamified[8] 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.

2.3.1 Performance benchmark

Dataset 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.

[8]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).

Accuracy of Model Inferences 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.com" where data and machine learning specialists may compete for the best scores on competitions. The winners of the competitions are, for many competitions, awarded prize money. This is a great way to gain experience in the machine learning field. What the target variables are for the competitions are what drives the development. According to Zou et al., these 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 brings bounding box positioning into the equation, representing how well the predicted box fits to the ground truth. A detailed explanation of what this means is given in the following paragraphs.

First we need 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 2.3.1 and 2.3.1, 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[9], the metrics usually used to evaluate it's performance is the number of true positives, false negatives and false positives.

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 8.

[9]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.

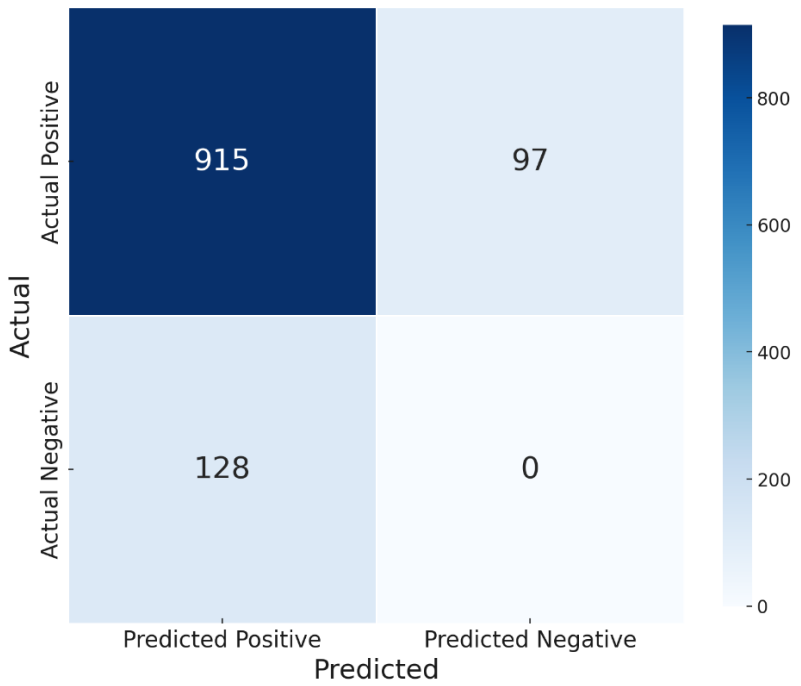


Figure 8: Example Confusion Matrix.

The confusion matrix in 8 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.

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

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.

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

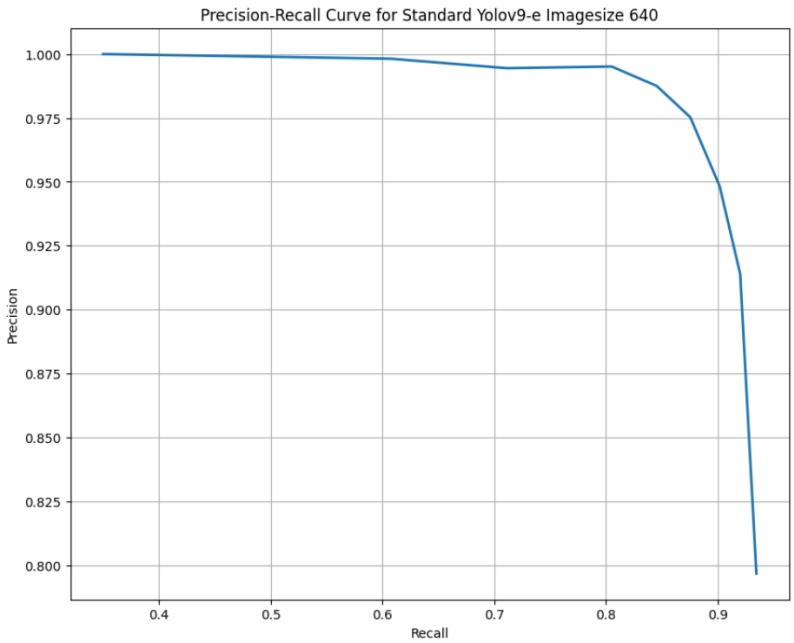


Figure 9: Example Precision-Recall Curve.

To assess a regular machine learning model's performance, the precision-recall curve is common practice (see figure 9 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 on the image, 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:

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

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 10 for an illustration. In this figure we can see that high IoU values equates to having a good fit with the ground truth bounding box. If the IoU value is over a threshold (image to the left), we define the detection to be a true positive. If we make a bounding box somewhere else (image in the middle), we get a false positive (a hallucination). If we don't have detections for a ground truth bounding box (image to the right), we have a false negative.

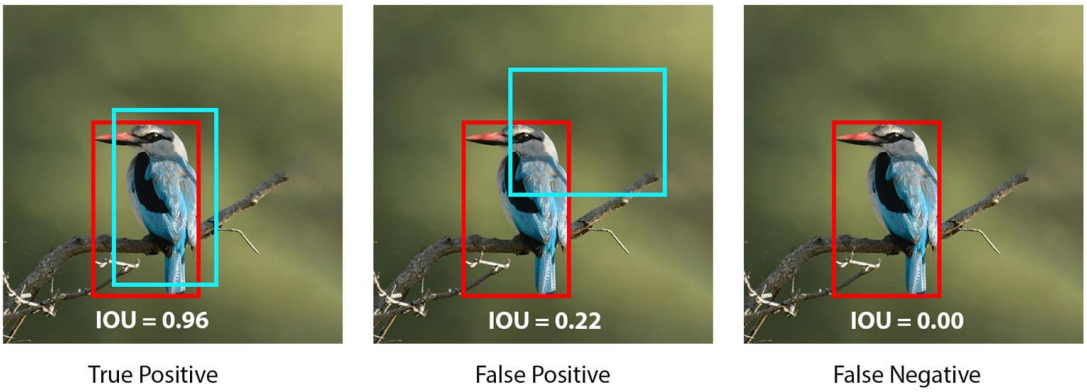


Figure 10: Intersection over Union (IoU).

The equation is simple:

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

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[10] (Zou et al., 2023).

2.3.2 Real-Time Detection Transformer

The Real Time Detection Transformer (RT-DETR) is a real-time object detection model developed by the Facebook (Meta) Research team. The model uses a transformer encoder-decoder architecture similar to a large language model. See figure 11 for an illustration of the architecture.

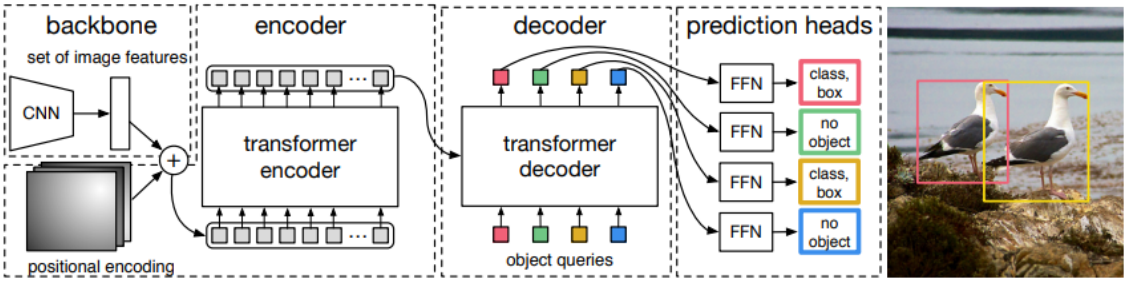


Figure 11: The architecture of the DETR model (Carion et al., 2020).

In the paper first introducing the model, Carion et al. claim it outperforms competitive baselines on panoptic segmentation[11] with a simple segmentation head trained on top of a pre-trained DETR.

The conclusions of the paper are as follows: "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

[10] 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 having the value the ability to adjust the threshold is a good idea when implementing an object detector

[11] This is a challenging pixel-level segmentation task. In segmentation tasks, an image is divided into meaningful regions. Altough different from object detection.

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, in particular 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).

2.3.3 YOLOv9

The YOLO (You Only Look Once) object detection algorithm is a popular choice for real-time object detection. YOLO processes images in a single pass, making it faster than traditional object detection algorithms that require multiple passes. YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell. The YOLOv9 is an improved version of the original YOLO algorithm, incorporating various enhancements to improve detection accuracy and speed. The YOLOv9 model is pre-trained on the COCO dataset. This is previously introduced dataset contains 80 different classes which the model is able to detect.

YOLOv9 author's comment on the DETR series

"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." (C. Wang et al., 2024).

2.3.4 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.

todo les YOLO in the Dark - Domain Adaptation Method for Merging Multiple Models

2.3.5 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.

2.4 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.

2.4.1 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.

2.4.2 GPT-4 with Vision on Object Detection

The well known large language models (LLM) have been generalized to perform more tasks, and are thus applicable to than just text. The ChatGPT-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 [12]. 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.

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 [13] 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.

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 4.4.

2.5 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 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 12 and are presented in the following paragraphs.

2.5.1 Eufy

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 12a.

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

[12] Detecting the mood of people is also much researched and could be implemented for images with high enough resolution and 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.

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

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.

2.5.2 Aqara

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 (2.2.8)), but when certain rules are triggered, such that a person has been located in a certain zone for a specified amount of time.

2.5.3 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.

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

Figure 12: Smart cameras from Aquara, Eufy and i-PRO

2.5.4 Viso

Viso.ai has applications 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).

2.5.5 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, Azure *and* Microsoft. Azure, owned by Microsoft, *and* Microsoft.. Anyways, 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" has some 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.

For outside monitoring, VMukti offer cameras that may connect through the mobile network, for monitoring outside remote locations.

VMukti delivers solutions for surveillance of vehicles, school buses, healthcare, shopping malls, smart cities, warehouses, campuses, examinations, premises, elections and banking.

2.6 Summary of Literature Review

A summarization of the current state of research, and where my thesis aims to contribute.

3 Methodology

Two cameras were deployed in a room of acquariums at "Fiskeri- og Søfartsmuseet" in Esbjerg to take images for building a specialized dataset and to evaluate the effects of developing a highly specialized detector rather than using a general.

3.1 Project

To investigate the effects of dataset quality in fine-tuning of models on the performance (primary objective 3, see section 1.2.2), a dataset was collected from the aquarium. 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". Many models were created to evaluate the effects of dataset quality. These include pre-trained but not fine-tuned "standard" models, models fine-tuned on the inconsistent partition, models fine-tuned on Consistent-1, and models fine-tuned on the external datasets PRW and CrowdHuman. All models were tested on the Consistent-2 partition. Also, multiple of the models were tested on all the consistent images (consistent-1 and 2), in order to evaluate different aspects of the models such as the influence of imgsz and number of epochs.

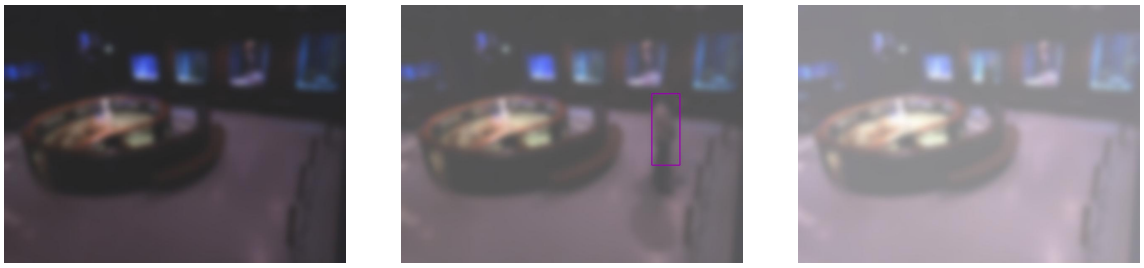
3.2 The FIMUS Dataset

This subsection provides an in-depth explanation of the dataset construction process, including the image capturing setup, image characteristics consistency, camera settings, technical challenges, image capture, and labeling process.

3.2.1 Camera Configurations

Mechanical adjustment of the aperture 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, being much shorter than any object detection application would need. This is the only camera property which is set manually. 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 13. 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 13: 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.

Consistent images The "Consistent-1" and "Consistent-2" partitions have consistent image characteritics. 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 dataset may function as training data for model fine-tuning. The settings used are found in table 2.

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	3264x2464

Table 2: Camera settings for the image capturing of 2nd and 3rd iteration of image capturing. 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 recorded data, storing each image 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 transmitting sharp, identifiable images over the internet. 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, a save quality of 90 was used for all the images.

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 1.2). 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[14] into the devices to start and stop the image capturing process.

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 21 in appendix A.

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.

1st iteration: the inconsistent images *Total number of images: 1312 (day 1), 986 (day 2) and 641 (day 3), total 2939. 1 subject.*

The first iteration of image capture was made with non-optimized camera configurations. To sufficiently brighten the images, the picamera.brightness attribute was set to 65. This is a post-processing 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 following steps in the project pipeline are described after the description of the 2nd and 3rd iterations of image captures.

Due to many technical difficulties the first few times images were being captured for the dataset, only the developer and author of this thesis is present in the images[15].

2nd iteration: the consistent-1 images *Total number of images: 295. 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).

[14](Secure Shell (SSH) is not detailed in this thesis, see section 1.2)

[15]Initially, an attempt was made to pass MQTT messages as a way to initialize image capture so multiple cameras could be deployed in several locations, thus speeding up and simplifying the image capturing process. This was discarded due to technical difficulties related to efficiently stopping the image capturing. For this single-deployment angle and area project, however, the approach with ssh-ing into the device worked fine.

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 acquarium. 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.

3rd iteration: the consistent-2 images *Total number of images: 466. 1-2 subjects.*

Similar to the 2nd iteration, but with 1-2 subjects instead of 1-4.

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. In some of the images (see figure 14), this 'seaweed-man' was presumably more likely to be a person than the human. In another instance, a person carrying a ladder was incorrectly recognized as one person carrying another.

(a) Seaweed and (presumably) a person

(b) Seaweed-man

Figure 14: "Sometimes, it feels like the seaweed is more man than I" -Anonymous Visitor

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.

3.3 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.

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.

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 1.2 for details).

3.4 Model Training

Licensing An effort has been made to keep the solution free and open to use, but some conditions may apply. The object detection algorithm 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 discussed, which could provide a solution for a company interested in keeping their solution hidden from competitors, is the DETR algorithm. This is under an Apache-2.0 license, which permits users to use and modify the code to fit their needs.

3.4.1 Transfer Learning

In this thesis, we are transferring the knowledge of a model that is pre-trained on a general dataset, to a model that is optimized for an aquarium environment. The general dataset in this case is the COCO dataset, a large dataset of 118 000 images and 80 different classes.

3.4.2 Hyperparameter Tuning

todo finskriv... Not really optimization.. More like finding... Since we're doing a cheeky approach to this. Done with autogluon, follow this guide for installation: AutoGluon guide. I had to run pip install autogluon twice for the imports to see autogluon.

This guide could be used to fine tune the hyperparameters of the model. A simpler guide was implemented to find the hyperparameters. This was to save time, and since our models require an okay level of hyperparameters. However, this choice to not give every dataset the same "fighting chance" with their optimal hyperparameters might have led to a lower validity of this experiments results.

Experiment 1: Do we need to run inference on the whole 2939 images to evaluate our model performance or will the results of evaluation on a subset be relative to the results of evaluation on the whole?

The standard yolov3 and yolov9 models were tested on the 1st and 2nd iteration images to see their out-of the box performance. Then, the yolov9 (which performed slightly better) was ran again but with a higher imgsz which typically increases the accuracy of the model. The three inferences: 1) yolov3 with imgsz 1280, 2) yolov9 with imgsz 1280 and 3) yolov9 with imgsz 1280 were then evaluated on the 1st iteration images and 2nd iteration images.

As expected, the inferences using a hightened imgsz was better. Another key takeaway was that the scores were relative, meaning evaluating on the full dataset was unnecessary. Moving forward, the 2nd iteration images from the acquarium were used to evaluate the models, while the 1st iteration images were used for training (and validation). This decision is also motivated by the fact that the 2nd iteration images are the most similar to the real-world application of the models.

We'd like the hyperparameter tuning process to focus on finding the hyperparameters that will

best infer on images from the 2nd iteration. Therefore, an amateur would plan to tune the hyperparameters for each of the models using their respective dataset's training data and the 2nd iteration FIMUS dataset for validation data. This means the yolov9 CrowdHuman Smodel would use the same data for validation and testing, but only to find the right hyperparameters. Then, after using the testing data for tuning the hyperparameters, we replace the FIMUS 2nd iteration images for the datasets validation images and train the models using the optimal hyperparameters. This as a method to find the optimal hyperparameters may seem intuitive, as it has the strength of focusing on the right data when doing transfer learning with data other than the specialized task the model is specializing for. This means that the models would, during training, train to infer on data different from what it sees during training, which is what it will be doing after deployment.

The aforementioned approach is completely ignorant to how machine learning works, however. In supervision machine learning, the validation set is used to tune the hyperparameters, and the test set is used to evaluate the model's performance. The test set should be representative of the data the model will see after deployment, and the validation set should be representative of the data the model will see during training. The test set should not be used to tune the hyperparameters, as this would lead to overfitting the model to the test data.

Then, the . Tuning the hyperparameters for the test data means that the validation data used for training later will not at all be representative of the situation earlier, meaning the optimal hyperparameters will no longer be valid. Such an approach would not

What makes this project interesting is not the varying nature of datasets. Usually, one wants the best, most similar, and most data possible, and it's therefore irrelevant to see what using a less relevant dataset would do to the model. In this project the focus is more about how much of the high quality data is needed to make an impact. If 100 labeled images does the same work as 1000, then much work can be saved by only labeling 100 images. If only 100 images is needed then setting up a proper labeling tool might not even be necessary, as it may be done faster using a sub-optimal but fast-to-employ labeling tool. The insights may also be useful when

3.5 Model Evaluation

3.6 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.

3.6.1 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.

3.6.2 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.

3.6.3 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.

3.7 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 15.

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 15: Heatmap Development Drafts

3.7.1 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 and may be seen in figure 16. This solution supports generating heatmaps from data in a pandas dataframe, allowing for filtering the dataframe to generate the preferred heatmaps based on for example the relevant times of the day or for the correct time interval.

Notes

Tried to download/use model from Roboflow, but either image has to be sent to an API which would not retain privacy, or the device has to host an API itself to run the inference... Seems unlikely to be the most preferable solution, as the device would have to set up the service and run it locally.

Possibly an interesting solution would be to do this with multiple devices. This supports the master-slave pattern of having multiple weaker computers and have them send to the stronger unit. Setting up private TCP connection between the weaker units and the strong unit and have the images sent to the stronger, so it can detect on them and send information etc. How many weak units do we need in order to make it profitable to have a strong GPU unit to do the processing? This whole systems sounds to be a complicating setuo process, not making the product modular and easy-to-use. Includes a lot of connection/networking to make the weaker units find and connect to strong, physically close device.

This task would mean setting up a strong device to host a network to which the weak units might connect to, and send images to. The issue is whenever images are sent, a lot of transmission is used... But the model takes image input size of 416x416. Would it be similar to just downscale the image before sending, or would this give the model less detail to work with?

Will now run several models on datasets from the web, i.e. the CrowdHuman dataset to see their accuracies. Will then deploy the models to device in acquarium to see if the best-performing model is an option in terms of size and inference speed. If it is preferable, I will attempt to increase it's accuracy by accumulating and annotating a specialized dataset for that setting, and training the final layers on the data. Can this be done with a

4 Results

This chapter will present the results of the human detection and tracking system, including the system's performance in the museum environment, the effects of adding labeled images from the museum environment to the training dataset, and the system's ability to detect and track humans in real-time.

The FIMUS 2nd iteration images were used as the test set for all of the evaluations. This dataset consists of 295 images of similar light condition and image quality. They are the closest representation of the images the device will be capturing in the experimental setting. All images are of 3264x2464 resolution (which is the maximum for the hardware).

4.1 Machine Learning Models of the Project

The models in this project may be seen in table 3. All models were pre-trained with the COCO 2017 training/validation datasets. Unless stated otherwise, they all have image input sizes of 640.

4.1.1 YOLOv9

The first 7 models are YOLOv9 models.

The first three are fine-tuned with the FIMUS inconsistent dataset for respectively 5, 15 and 50 epochs while freezing the backbone[16].

The fourth model is fine-tuned for 5 epochs on the CrowdHuman dataset, also with a frozen backbone.

Input image size The fifth, sixth and seventh modelsare not fine-tuned. The dependent variable in this sub-experiment is the image size, which is the size of the input image when the model is performing inference. What input image size is optimal depends on the dataset and use case, and should be tested for a given sceneario. According to some, increasing the size may elevate the accuracy of a model (see the quote below).

> **Increasing the image size may elevate 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).

In the experiment of this thesis, increasing the image size rendered worse results for the YOLOv9 model when considering the location accuracy of the detections as well. This discrepancy may be due to **<empty citation>** only considering AP50, or it could be due to the difference in the datasets.

The next model is a standard YOLOv3 model from Ultralytics, also pre-trained on COCO. four models are DETR models, which are transformer-based models. The first two are ResNet50 and ResNet101, while the last three are the ResNet50 model with different confidence thresholds. The DETR models are not fine-tuned, and the confidence threshold is set to 0.5 for the first model, and then increased to 0.9, 0.95, 0.97 and 0.99 for the last four models.

[16] Freezing the backbone of a machine learning model is done to avoid unlearning basic shapes and features of the model. In transfer learning, this is a common practice when specializing a model to learn a new task or to better perform in a given sceneario where additional, specialized data may be limited.

4.2 Model Evaluation

The full model evaluation jupyter notebook can be seen in appendix todo insert model evaluation ipynb.

The confidence threshold for all detectors were set at 0.1, allowing for the calculation of an AP for confidence levels from 0.1 to 0.95.

The FIMUS 5, 15 and 50 epochs are models pretuned on the COCO dataset, and fine-tuned with the FIMUS 1st iteration dataset.

These were all trained for and detected at imgsz 640. The CrowdHuman 5 Epochs model was fine-tuned with 5 epocs on the training/validation datasets of CrowdHuman. The rest of the models were not fine-tuned.

Model	**AP50**	**AP75**	**AP90**	**mAP50-95**
FIMUS Fine-Tuned 5 Epochs	0.97	0.89	0.67	0.88
FIMUS Fine-Tuned 15 Epochs	0.94	0.67	0.44	0.71
FIMUS Fine-Tuned 50 Epochs	0.94	0.75	0.51	0.77
CrowdHuman Fine-Tuned for 5 Epochs	0.96	0.93	0.77	0.91
YOLOv9 Imagesize 320	0.92	0.85	0.62	0.84
YOLOv9 Imagesize 640	0.96	0.93	0.90	0.94
YOLOv9 Imagesize 1280	0.97	0.93	0.79	0.92
Yolov9 Confidence .50	**0.99**	**0.97**	**0.95**	**0.98**
YOLOv3	0.95	0.88	0.63	0.87
DETR ResNet50	0.32	0.29	0.17	0.28
DETR ResNet101	0.63	0.56	0.30	0.53
DETR Confidence .90	0.33	0.28	0.13	0.27
DETR Confidence .95	0.98	0.87	0.41	0.83
DETR Confidence .97	**0.99**	0.89	0.43	0.85
DETR Confidence .99	**0.99**	0.92	0.48	0.88

Table 3: Performance Metrics of Object Detection Models on 294 images.

Larger Test-set As seen in table 4, adding in the images collected during the 3rd iteration reduced the scores for YOLOv9 while increasing the scores of the YOLOv3. The fact the scores change when adding more test data illustrates illustrate that we have not yet reached a level of test data where the scores converge. More test data should be added to further increase validity of the results.

Model	**AP50**	**AP75**	**AP90**	**mAP50-95**
YOLOv3 (n=294)	0.95	0.88	0.63	0.87
YOLOv3 (n=759)	0.97	0.91	0.67	0.89
Yolov9 (n=294)	0.99	0.97	0.95	0.98
YOLOv9 (n=759)	0.98	0.96	0.93	0.96

Table 4: Performance Metrics of Object Detection Models on 294 images vs 759 images

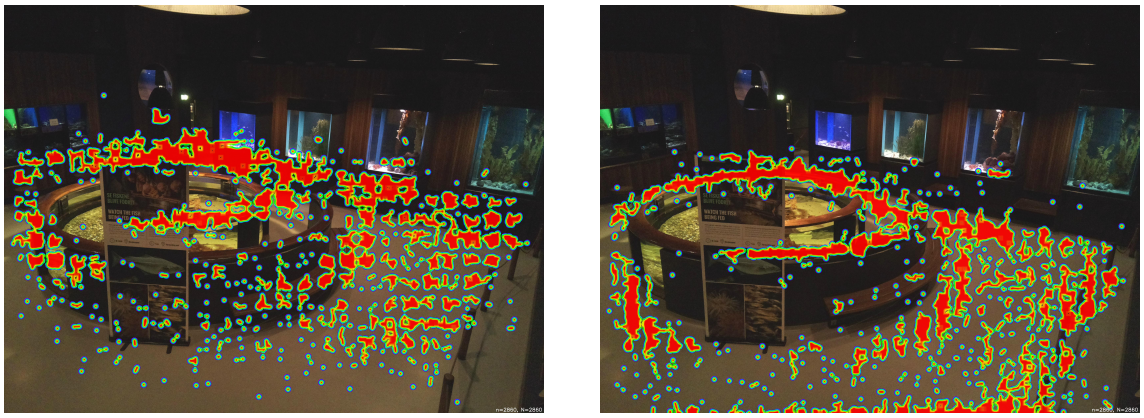
The pre-trained weights are available for multiple of the available versions of the Yolov9 model. The largest weights-file, called 'yolov9-e', is what has been used for this project. These weights are available for download on the Yolov9 Github repository.

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 3.7, 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 16. These heatmap generation code for the two heatmaps are identical, apart from one variable: the position where detections are mapped to. In 16a 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 16) the railing of the fish tank in the center.



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

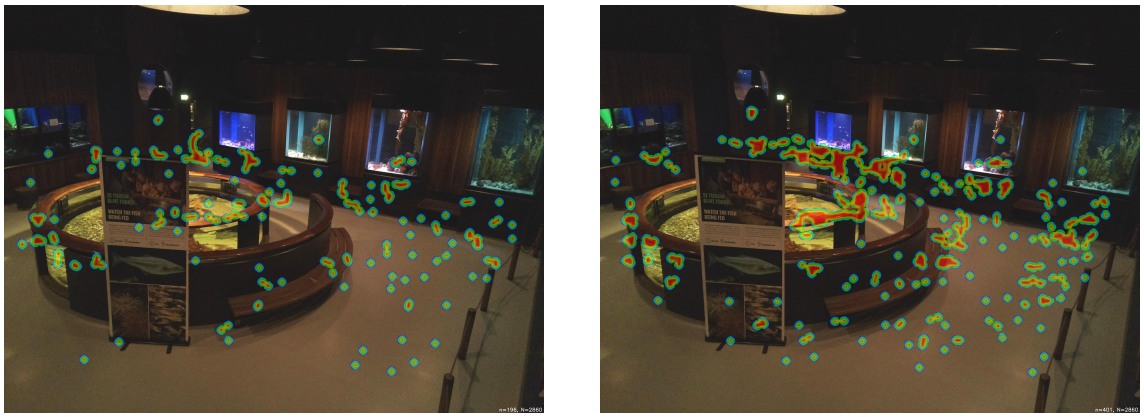
(b) Final Heatmap: Supervision, sampled from the *bottom* of detections

Figure 16: Final Heatmaps

There's another, more important take away from the small modification. While seemingly similar, the heatmap sampling detections from the middle of the bounding boxes (16a) 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 (16b) 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[17], these heatmaps may provide valuable 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 17.

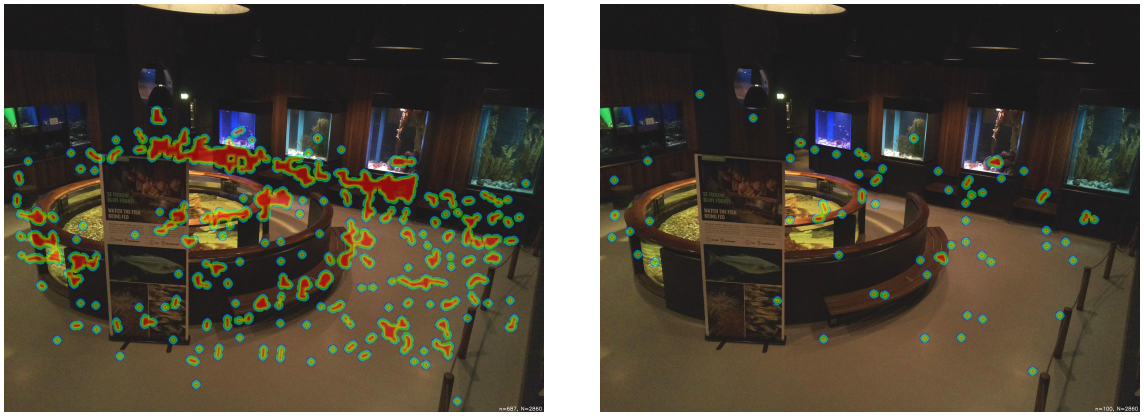
[17]These weaknesses in our detector models could be revealed by looking at the annotated images. However, looking at the annotated images is not possible for a on-device processing image-deleting device. In this case, one would need to display/plot the detections onto a base-layer image (heat map), or make use of obfuscation discussed in section 2.2.9 to illustrate and reveal model weaknesses.



(a) Heatmap Wednesday 8th of May, 2024 (b) Heatmap Saturday 11th of May, 2024

Figure 17: 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 18.



(a) Heatmap 13:00-14:00 (b) Heatmap 16:00-17:00

Figure 18: Hourly Heatmap

The heatmaps in figure 18 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 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.

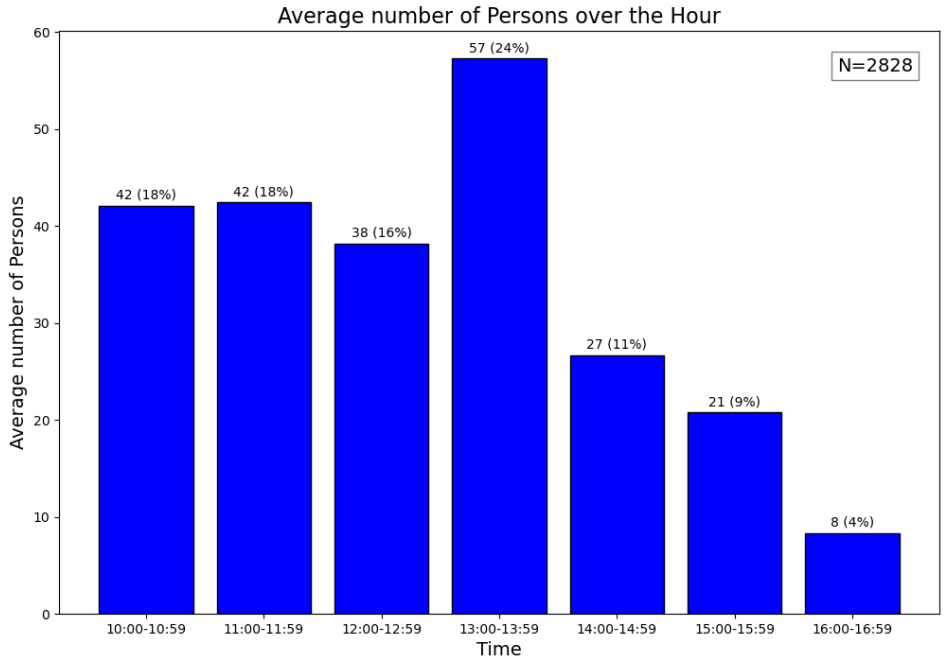


Figure 19: Peak Hours Analysis

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[18], 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 whole facility more favorable for a visit regardless of seasonal factors.

4.4 Third-party Services

As we've seen in section 2.4, 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.

[18]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.

4.4.1 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 Conclusion

Summarization of the thesis and its contributions to the field.

5.1 Disclaimers

5.1.1 Plagiarism

Substantial parts of some of the sections in the literature have been copy-pasted from a pre-thesis project of the same author as this thesis on the same subject from the preceding semester. The text has been partially rewritten and improved, but still share a high degree of similarity to the original content. This includes much of the text in sections 2.2, 2.3, 2.5

5.1.2 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 20 we see an example of one of these distracting suggestions, where it may seem AI thinks very strongly we need more reasearch on who are not wearing masks and are not vaccinated... The somewhat humurous suggestion serves only as a distraction.

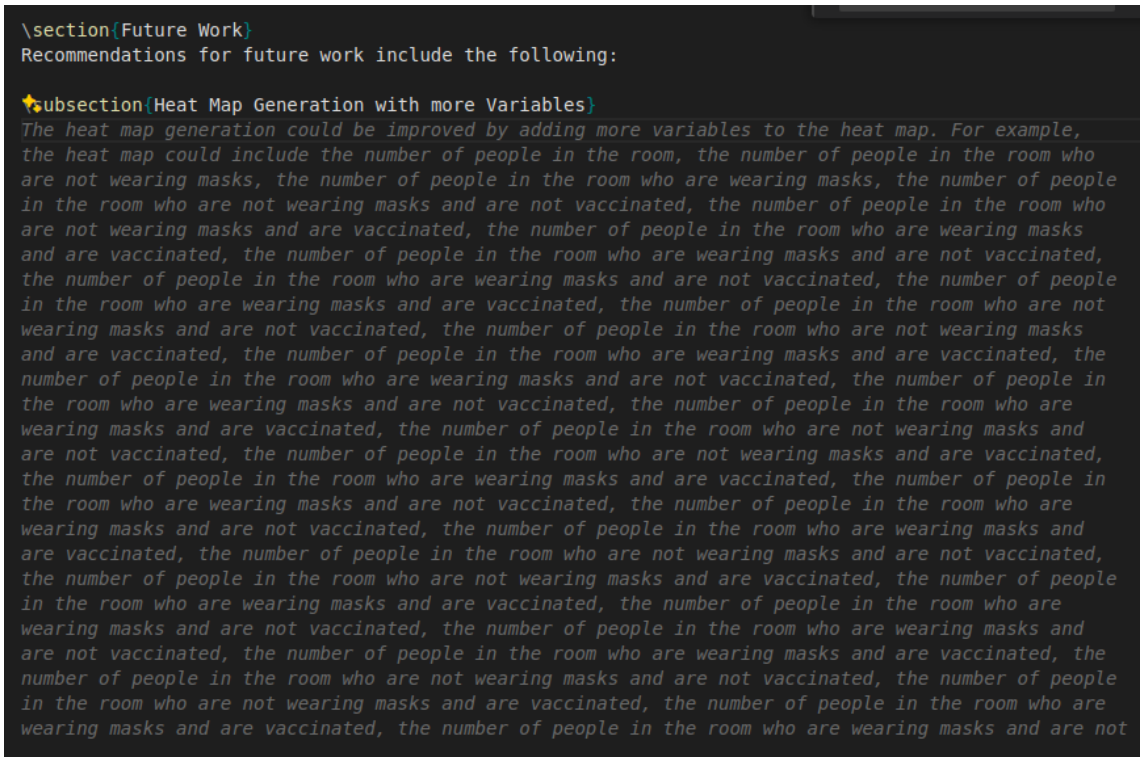


Figure 20: The Co-Pilot is mostly annoying for Latex.

However, some boiler-plate Latex-code and some of the sections have been fed to an AI to verify it's quality and to get suggestions on how to enhance readability and flow. The tools used have been Chat-GPT4 by OpenAI and Github Co-Pilot.

5.1.3 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.

6 Future Work

Recommendations for future work include the following:

6.1 Heat Map Generation with more Variables

Bibliography

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. 10, 11).

Blum, A., Ligett, K., & Roth, A. (2011). A learning theory approach to non-interactive database privacy. (Cit. on p. 11).

Boesch, G. (2023). *Yolov7: The most powerful object detection algorithm (2024 guide)*. https://viso.ai/deep-learning/yolov7-guide/ (cit. on p. 26).

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. 6).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. (Cit. on pp. 21, 22).

Chhabria, P. (2022). *Expert a 1m53s 16x9*. https://www.youtube.com/watch?v=w78U7w33NTI&t=5s (cit. on p. 25).

Dictionary, O. E. (2023, July). Gamification (n.) https://doi.org/10.1093/OED/7320229446 (cit. on p. 17).

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. 11).

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. 11).

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. 14, 15).

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. 13, 14).

Eufy. (2022). *What makes eufycam 3c stand out?* https://eu.eufy.com/pages/security-eufycam3c (cit. on p. 24).

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: date-month-year]. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022L2555 (cit. on p. 7).

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. 17).

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. 6).

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. 10).

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. 8, 11).

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. 12).

James Gallagher, P. S. (2024). How to train yolov9 on a custom dataset [Accessed: 2024-05-21]. (Cit. on p. 36).

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. 5).

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. 16).

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. 17, 21).

Luccioni, S. (2023). Ai is dangerous, but not for the reasons you think author [Accessed: May 17, 2024]. https://www.youtube.com/watch?v=eXdVDhOGqoE&ab_channel=TED (cit. on p. 18).

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. 14).

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. 49).

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 p. 49).

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. 22).

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. 5).

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 p. 49).

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. 9).

Redmon, J. (2020a, February). *Joseph redmon's twitter* (Twitter, Ed.). https://twitter.com/pjreddie/status/1230524770350817280?lang=en (cit. on p. 16).

Redmon, J. (2020b, June). *Joseph redmon's twitter* (Twitter, Ed.). https://twitter.com/pjreddie/status/1230524770350817280?lang=en (cit. on p. 16).

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. 7).

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. 17).

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. 11).

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. 8).

Termly. (2023). Natural person [Accessed: 2024-05-15]. https://termly.io/legal-dictionary/natural-person/ (cit. on p. 6).

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. 6).

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. 16).

Wang, C., Yeh, I.-H., & Liao, H.-Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information. (Cit. on p. 22).

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. 8, 13).

Westin, A. (1967). *Privacy and freedom*. Atheneum. (Cit. on p. 6).

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. 11).

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 pp. 9, 10).

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. 13, 17, 18, 21).

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 21: 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

TinyML is when machine learning models are aimed at deployment to heavily resource constrained environments, like what are called frugal devices. These are, as previously mentioned in the introduction, 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 2.2.4 about federated learning as a way of implementing a collective learning network for the edge devices.