Precise Detection of People Using Pre-Trained Machine Learning Models With Assisting Heat Sensor Camera

ELIAS RUDBERG & VIKTOR JOELSSON
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Precise Detection of People Using Pre-Trained Machine Learning Models With Assisting Heat Sensor Camera

Elias Rudberg
`el0032ru-s@student.lu.se`
Viktor Joelsson
`vi0842jo-s@student.lu.se`

Department of Electrical and Information Technology
Lund University

Supervisor: William Tärneberg

Examiner: Maria Kihl

June 15, 2021

Abstract

The thesis examines the possibility of developing a system that can be used to detect the number of people in a given area with high accuracy. By using a variety of hardware components, publicly available object detection models and a cloud platform a highly modern approach is examined. The aim is to develop a cost-effective and scalable system which should be able to adapt to a given area. Different modern object detection models are being evaluated based on selected metrics to optimize the outcome in the given area. The thesis is based on previous research and work in the field. The approach is based on collecting data locally using an edge device and forward relevant data to the cloud for further analysis. By enabling an interchangeable model for object detection, publicly available object detection models can be reused to evaluate its performance in a given area. External thermal data is used to validate a detection to achieve a more accurate detection, thereby extending the scope of the system. The proof of concept intends to demonstrate that the system described can be developed and evaluated with limited financial resources. Further analysis is intended to reveal further uses of the system, but since this is not the primary objective, it will be discussed in a purely abstract manner.

Popular Science Summary

An increased interest in data collection and data usage in combination with a global state, the current pandemic, sets requirements for products / services that can detect the number of people in a given area. A system that helps with the ability to comply with restrictions, recommendations or other reasons is the basis for the thesis.

By using a combination of different modern techniques and publicly available material the thesis demonstrates how a system that detects the number of people in a given area with a high accuracy could be designed with a limited financial budget.

The system is designed as follows, data is collected and handled locally in a pre-defined area. Computational power demanding tasks are delegated and handled externally. The information about the current condition in the given area can be requested, i.e. the number of people present in the given area. The delegation and information retrieval is handled through API calls, request and response.

The proof of concept evaluates different models for detection and examines potential benefits of using thermal data as an additional attribute. The thesis advocates further development through a system design that makes it possible to replace the object detection model for further evaluation and by proposing alternative uses for the system.

The system is developed through a combination of experience and research in the area. The system detects the number of people in an area with a high accuracy using a pre-trained object detection model on its own and shows how thermal data can be used to improve detection. Since the thesis is a proof of concept, optimization has had a low priority but with the intention of developing an optimized system with a purpose, to solve a presented problem.

The thesis focuses mainly on the technical aspects related to data collection, collaboration between components and the use of different detection models, but also takes into account some legal and ethical aspects the thesis includes.

Acknowledgements

Table of Contents

List of Figures

x

List of Tables

List of Acronyms

CNN	Convolutional Neural Network
mAP	mean Average Precision
FPS	Frames Per Second
R-CNN	Region Based Convolutional Neural Networks
SVM	Support Vector Machine
RoI	Region of Interest
RPN	Region Proposal Network
SSD	Single Shot MultiBox Detector
YOLO	You Only Look Once
IoU	Intersection over Union
COCO	Common Objects in Context
ACC	Accuracy
PR	Precision-Recall
SBC	Single Board Computer
NMS	Non-Maximum Suppression
GCP	Google Cloud Platform
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
VM	Virtual Machine
ARM	Advanced RISC Machines
APT	Advanced Packaging Tool
GUI	Graphical User Interface
WiFi	Wireless Fidelity

IoT   Internet of Things

GDPR   General Data Protection Regulation

EU   European Union

RGB   Red Green Blue

IR   Infrared

API   Application Programming Interface

IDE   Integrated Development Environment

ILSVRC ImageNet Large Scale Visual Recognition Challenge

FOV   Field of View

Introduction

The current information society with its high demand on effectiveness and consistent feedback leads to a more connected and updated state for the common person. What follows is a progressive development and rapidly changing environment which affects more and more people, and also slowly but surely raises the bar on what was yesterday considered revolutionary.

A modern area of interest is reading arbitrary data locally to collect and analyze it for a specific use. Obtaining and managing data comes with some technical requirements that are in line with the technical equipment and services available at present time. The availability is an essential factor but also the degree of cost-effectiveness. Not only do you want to be able to offer a solution, but it must also be cost effective. There are potential risks and / or ethical dilemmas which have a strong connection to data collection, namely anonymity and privacy. If the information collected contains personal identifiers, it may violate the privacy of individuals.

In early 2020, a pandemic was a fact and new living conditions followed. The new living conditions ranged from total lockdown to restrictions and/or recommendations regarding movement, social distancing and limitations on the number of people allowed in a given area. The necessity of keeping track of the number of people in a given area was enhanced due to the restriction requirements and by extension to maintain a minimal spread. Different techniques can be used that satisfies the need to keep track of the number of people in a given area. Differences regarding the technical solution that is being implemented, may for instance involve cost efficiency, the possibility of further development and the extent to which an accurate result can be determined.

A variety of object detection models available at the present time has shown promising results detecting people in a given area using images in combination with machine learning algorithms and datasets [1]. Libraries containing extensive datasets and open-source solutions with a wide range of usage in computer vision is available to the public [2] [3] [4]. In addition cameras are relatively cost effective and offer the opportunity for further development.

The disadvantage of camera surveillance is that it may infringe on privacy, if

the data collected is not handled properly. Requirements regarding why data is collected and how it is to be used is statutory in the European Union (EU) for companies active in the region since May, 2018 [5]. The reason is based on the opinion that the average person should have control over data that is linked to the individual and how it is potentially used.

The aim of this paper is to offer a viable method to monitor a specific area without human supervision. The system to be developed by combining hardware, software and third-party services to detect the number of people in a specific area, at a given time, is a proof of concept. The proof of concept will determine if it is feasible with a limited and reasonable financial budget to develop a system with the above requirement and to also investigate further areas of use.

1.1   Background

The request from CGI Sverige AB is a system that can calculate the number of people in a given area under the pretence that the technical requirements are feasible to meet for what exists today. The system use hardware on-site to gather data which can be forwarded to a cloud platform. The cloud platform will analyze the data and store the current state.

The purpose is to through said application be able to give information about the number of people in said area, or in extension monitor if restrictions regarding number of people in a given area are complied with.

1.1.1   Overarching Goals and Problem Statements

The goal is to be able to as accurately as possible provide information about the number of people present in a given area.

- Research different hardware, models and methods in order to find a composition which makes it possible to meet the requirements of the system.

- With information gathered from cameras that are placed at different angles, map out the open space accurately, and for example making sure to not portray a person twice.

- Combining a model to detect people present, with another method to help make more accurate detection. For example, using a heat sensor to make sure that paintings and/or mannequins that visualize a person don't get counted if it is not an actual person present in the given area.

1.1.2   System Limitations

The system will include several components which places demands on and limits the general possibilities.

The limited time frame can inhibit optimization and the extent of testing. The

time for implementation, documentation, research and purchase is limited.

Concerning the overall performance of the system overfilled spaces could lead to problems in the system. For example a train cart, if said cart is overfilled with people, the calculation becomes more difficult, hence may result in an overall limitation of the system. Furthermore obstacles in a given area could also affect the result.

The system will rely on purchased hardware. The amount and quality of the hardware will set the basic limits on what is feasible, hence the financial budget for the system will limit the system.

The system may not be able to process the environment in real time which can lead to a delay in the system and incorrect calculations may occur.

During the current pandemic extended testing may be difficult to achieve. Restrictions and remote solutions may inhibit the number of people in a given area and willingness to participate.

1.1.3 Related work

The related work section includes similar studies within the affected area. The paper is highly influenced by earlier work related to object detection models and using for instance a Single Board Computer (SBC) locally to manage and capture data from external components.

Object Detection Models

Various object detection models have been developed with high accuracy and speed [1]. The different models address the problems that occur when analyzing an image, detecting and locating objects, differently in several aspects but also similar in some aspects. The optimal object detection model is both accurate and fast, but usually one is prioritized over the other. An accurate model usually has a more complex detection, which leads to reduced speed. The default architecture of a model consists of replaceable or modifiable components which results in that one can adapt a model or choose a model for a specific purpose in a given area. TenserFlow provides pre-trained open-source object detection models to the public and by using TensorFlow Serving it is easy to implement, utilize and compare different models while maintaining one and the same system architecture [6].

Embedded systems and Cloud computing

The Raspberry Pi can be used for local embedded applications [7]. The attribute of connecting multiple components together, whether it is sensors, cameras or similar components makes it favorable in smart and small applications. The extensive number of outlets, network capabilities and Raspberry Pi OS provide functionality and ease of implementation. The Raspberry Pi is small but powerful, and through

integrated Wireless Fidelity (WiFi), for instance, communication with external devices is possible. External communication opens up a variety of options, including delegation of tasks. Since the Raspberry Pi has limited capacity, delegation can be advantageous as more capacity is required for a task.

Mentioned qualities of the Raspberry Pi makes it possible to integrate the device with a cloud platform. The delegation of tasks may include locally capturing data from an area and passing it to the cloud for further processing of data that may include, for example, storage or execution of tasks that require more computer capacity / power [8].

1.2   Thesis outline

- Chapter 1 is an introductory chapter to the paper whereas the basis for the paper is described and introduced.

- Chapter 2 includes the technical background relevant to understand and benefit from the paper.

- Chapter 3 describes the work process, from idea to developed system.

- Chapter 4 visualizes results obtained from testing the system using different models and potential impact of an external component.

- Chapter 5 processes different aspects of the system, the different models and includes the overall analysis of the paper.

- Chapter 6 describes potential further development of the system that was not included but which is made possible by the chosen approach.

- Chapter 7 presents a conclusion of the paper.

Theory

The chapter will introduce theory and concepts affecting the paper in order to facilitate understanding and reasoning.

2.1 Object detection

In this section reasoning is being made based on a review of different modern object detection models using deep learning as examined by [1].

Object detection using deep learning has undergone development in the last decade with a number of nuanced and / or innovative detection networks based on Convolutional Neural Network (CNN). Object detection is referred to as the joint result of image classification and object localization. That is determining what object, if any, is present in an image and its location. In the generic sequence of events, the outcome is a label with an associated bounding box. Object detection is a difficult task due to various reasons. For instance, objects in images can appear in different scales and ratios but also contexts. Therefore a need to systematically scan the image is necessary. Object detection using deep learning can be used to achieve an optimized systematic scan of the image, within a given time interval, depending on the object detection model used [1].

2.1.1 Brief introduction to CNN

The architecture of a CNN and the consequence of different architectures is introduced briefly as described by [9] and [10].

A CNN is made up of multiple layers. The CNN consists of convolutional, pooling and fully connected layers. The common CNN architecture starting point is convolutional layers with subsequent pooling layers in an iterative process whereas the number of layers and iterations depend on the complexity of the CNN. Finally, fully connected layers typically end a CNN [9].

The performance of the CNN, in the general case, depends on the number of layers present in the CNN. Performance refers to accuracy and speed. By increasing the number of levels a more complex network is the result. A more complex

network, in the general case, yields a more accurate CNN. However a more complex network also results in difficult training and testing, which adversely affects speed [10].

2.2 Object Detection Models

The following introductory section to object detection models is based on reasoning from [11], [12], [13], [14], [15] and [1].

Various object detection models, currently state-of-the-art, consist of a base- and detection network. In some cases the base network is completely separated from the detection network, in other cases the base and detection network is integrated. Regardless of architecture, they have the same task. The first one is a base network that works to provide the second, detection network, with high-level features which will be used for detection. The common output that the different object detection models generate is a bounding box along with classification / label. The trade-off between accuracy and speed is essential, especially when a real-time or time dependent object detection system is considered [1].

Region proposal based object detectors such as Region Based Convolutional Neural Networks (R-CNN), Fast R-CNN and Faster R-CNN consist of various sub components, including a region proposal method or network component. The various components make training difficult and lead to expensive detection even though modern end-to-end systems such as Faster R-CNN have been developed to avoid separate training and optimize the region proposal method. Progressing through different subsystems takes relatively a long time compared to competing models such as Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). Region proposal based object detectors cause a disadvantage due to the above reasons, even the modern detectors need to be modified for real-time detection [11] [12] [13] [1].

Single forward pass object detectors for instance SSD, YOLO as well as newer versions do not need the additional region proposals step. A direct mapping from pixels to bounding box and classification output by an integrated base- and detection network makes real-time detection possible [1] [14] [15].

2.2.1 Detection Networks

The following sections will describe the basis of various detector networks relevant or indirectly relevant to understand essential parts of the paper. Since training and optimizing models is not the primary area of the paper, the focus will be on the detection process, but include a certain aspect regarding training. For instance the loss function used during training is not included, but different classification functions are, thus a more high-level explanation will be presented. An important aspect to emphasize is that various components in the detector networks are interchangeable. One can simply optimize the outcome of a detector network by e.g.

replacing the standard base network with another or experiment with different thresholds for optimal outcome.

R-CNN

The following section describes the R-CNN based on [11].

R-CNN is an extended optimized object detection model with a CNN base network. Instead of convolving through the entire image, an initial step of region extraction is being conducted, hence region proposals are located. The regional proposals are generated from a regional proposal method. Various methods exist however in the case of R-CNN selective search is the standard method described in section 2.2.3. Around 2000 (default) category-independent region proposals are generated for an input image. The regions proposed are processed through image warping, resulting in a fixed size image region for each of the proposed regions. The image warping is necessary to be able to extract features using a CNN. Each of the fixed size image regions / region proposals is forwarded to the CNN. The output of the CNN is a feature vector of fixed size. The feature vector is classified and a score is calculated using binary linear SVMs, introduced in section 2.2.4. Once each proposal is classified,Non-Maximum Suppression (NMS), described in section 2.2.4, is performed. A bounding box regressor is used to minimize the localization error in order to localize where in the image a detection occurred more accurately [11].

All the regional proposals are handled individually and passed through the CNN. The CNN, SVMs and bounding box regressor is also difficult to train due to the fact that they are trained separately [12]. The result is a highly computationally expensive approach [11].

Fast R-CNN

The following section describes the Fast R-CNN based on [12].

Fast R-CNN improves some of the drawbacks of R-CNN. Instead of forwarding the region proposals to the CNN separately, Fast R-CNN processes the complete image resulting in a feature map. Further improvement was to incorporate end-to-end training. The input in Fast R-CNN consists of an image and region proposals. The region proposal method, like in R-CNN, is selective search. From the feature map region proposals are being extracted and processed through a Region of Interest (RoI) pooling layer. The RoI pooling layer generates a fixed size feature vector as output to be forwarded to the fully connected network, consisting of sequential fully connected layers. The output consists of two vectors, one related to classification and the other used for localization. The classification is based on the softmax function, described in section 2.2.4, and the output is a discrete probability distribution. The output of the softmax classification is $K + 1$ discrete probability distribution of every class K and the additional reserved slot of being an object or not. The localization regards the bounding box regressor offset. Fi-

nally NMS is performed. [12].

The improvements that followed is an end-to-end network, thereby avoiding separate training of components in the network [12]. The improved version still uses selective search as a region proposal method which imposes an overall delay in the system [13].

Faster R-CNN

The following section describes the Faster R-CNN based on [13].

Faster R-CNN addresses the problem related to the region proposal method used in both R-CNN and Fast R-CNN, selective search. Selective search introduces an approximate delay of 2 seconds per image leading to a persistent negative aspect of Fast R-CNN (and R-CNN). Faster R-CNN solves the remaining bottleneck of the Fast R-CNN system by replacing the region proposal method, selective search, with a Region Proposal Network (RPN) described in section 2.2.3. Faster R-CNN consists of a RPN and Fast R-CNN detector network which together form a unified object detection network. The RPN share the convolutional layers with a Fast R-CNN detector network which results in a minimal computation time generating regional proposals, around 10 ms per image. The detector network used in Faster R-CNN is inherited from Fast R-CNN as described in section 2.2.1. The output from the RPN is processed through a RoI pooling layer, forwarded to the fully connected network as a fixed size feature vector. The process continues with classification and bounding box regression which generate two outputs as in the case of Fast R-CNN. Last NMS is applied [13].

The accuracy of Faster R-CNN is in principle the same as Fast R-CNN. However the improvement in speed is significant. The improvement results in the network being close to real-time detection with a high accuracy [13]. Real-time detection of the Faster R-CNN can be achieved by modification [1].

SSD

The following section describes the SSD based on [14].

SSD applies a single deep neural network and does not need a separate region proposal method. SSD takes an input image and passes it through a base network. The initial feature extraction step produces multiple feature maps. By removing pooling as well as fully connected layers and adding multiple convolutional layers in reducing size, detection at different scales is made possible. The convolutional layers decrease in the terms of size as progression occurs. Each of the multiple convolutional layers outputs a feature map, hence shrinking the size of the feature map to be able to detect objects in different sizes. In an initial stage, detection of small objects is possible and the deeper one progresses the larger objects can be detected [14].

Each feature map consists of a grid of cells where the scale of the grid depends on the convolutional layer. The prediction is based on default boxes. Default boxes of different size and aspect ratio are applied at every feature map cell, resulting in a positive match if the Intersection over Union (IoU) exceeds a certain threshold (default 0.5) in relation to the ground truth. The default boxes are compared to the ground truth values during training in order to find the optimal filters. A loss algorithm is being applied with regard to positive and negative matches. The number of default boxes is arbitrary (default 4). For each default box that represents a match, confidence score and bounding box offset is predicted. Each prediction consists of centre coordinates of the bounding box, width, height and c confidence scores for every class in relation to the default box. Background is treated as a class in SSD. A Softmax classifier is used for predicting the classes, based on the softmax function. Finally NMS is applied [14].

The SSD512 and SSD300 differs only in image input sizes, 300 x 300 and 512 x 512, whereas SSD512 outperforms Faster R-CNN in terms of both accuracy and speed. SSD300 prioritizes speed but still maintains a high accuracy. Concerning default boxes, similarities with anchor boxes in Faster R-CNN exist, but with the difference that the set of default boxes can vary depending on different feature map resolutions, resulting in a more efficient method. SSD could be used as a real-time object detection model. The architecture result in ease of training and integration [14].

The drawbacks of SSD is the difficulty of handling small objects, however by replacing the base network with a modern version the problem can be solved to varying degrees [1].

YOLO-v3

The following section describes the YOLO-v3 based on [16].

YOLO-v3 applies a single CNN with high performance regarding accuracy and speed. YOLO-v3 base network is Darknet53, described in section 2.2.2. The base network used provides a more accurate detection but at the expense of speed compared to earlier versions [16].

The network resizes the input images to network size, hence images do not need to be resized. Detection occurs at 3 scales. A grid of cells is generated of different sizes depending on the scale. The 3 different scales are intended to detect objects of different sizes, large, medium and small. The output feature maps of the 3 different scales predict 3 bounding boxes for every cell in the output feature maps. The prediction is based on anchor boxes / priors. For each cell, in each feature map, of the 3 different scales, 3 anchor boxes are applied. The anchor boxes are determined by k-means clustering. Each prediction consists of centre coordinates of the bounding box, width, height, objectness score and c confidence scores for every class in relation to the anchor box. Where the center point of the object is located determines which cell is responsible for predicting the object [16].

The predicted bounding box offset to the anchor box is calculated by using a logistic function. The objectness score is the probability of a cell containing an object, whereas the center cell should have the objectness score 1. The objectness score is a result of the probability of being an object and the IoU (default threshold 0.5) of the predicted bounding box and the ground truth bounding box. The result is predicted by logistic regression. A independent logistic classifier are used for predicting the classes, a logistic function is described in section 2.2.4. Finally NMS is applied [16].

The YOLO-v3 approach is a strong competitor to other object detection models with properties that generate high accuracy and speed. By varying the resolution, high or low, the accuracy and speed are affected. A low resolution gives priority to speed while a high resolution prioritizes accuracy. Also by using a independent logistic classifier loss while training provides the ability to use various datasets, for instance Imagenet, described in section 2.2.5, because the approach promotes multi-labels as opposed to softmax [16].

In addition, the approach may encounter difficulties inherited from YOLO. Deviating objects regarding size and aspect ratio or small objects that are grouped together may lead to difficulties [1].

2.2.2 Base Networks

Darknet-53

Darknet is an open source neural network, where Darknet-53 is the successor to the Darknet-19 network, Darknet-19 being the network used in YOLOv2. Darknet-19 is constructed with 19 convolutional layers with pooling layers spread throughout. And while Darknet-19 has good performance, competing with the current state of the art networks at the time, YOLOv3 came out with a new version of Darknet, Darknet-53, which as the authors say "*is a hybrid approach between the network used in YOLOv2, Darknet-19, and that newfangled residual network stuff*"[16].

The additions made leave Darknet-53 a little slower than it's predecessor as a consequence of the added layers (19 → 53), the new shortcut connections, and the extra complexity that follows. But while the additions of the extra layers, and the "*newfangled residual network stuff*", might slow it down a bit, it boosts the accuracy of Darknet-53 compared to Darknet-19 [16].

MobileNetV2

MobileNetV2 is an open source model, the successor to the MobileNetV1, and works to improve the performance of mobile models of different sizes. Just as the first version of MobileNet, it uses depthwise convolution. Where it differs from its predecessor is that the V2 changes the architecture so the final 1x1 convolution layer in the block decreases the channels and the data flow. Keeping a low dimensional tensor is a key aspect in order to be able to reduce the amount of

computations done. Too small however, and it would not yield a good result. Therefore, to be able to take a low dimensional tensor as the block input, the V2 architecture also adds an expanding layer at the start of the block. That first expanding layer increases channels which are then run through the 3x3 depthwise convolution, and then finally projected down again to fewer channels at the final 1x1 layer [17].

V2 also adds an optional residual connection, a "shortcut", between the bottlenecks in the blocks. This connection helps gradients propagate through the network without having to pass through the network blocks and it's internal layers. Passing the gradient through non-linear activation functions can be harmful to it, depending on the weights. Letting the gradient propagate through the network by "skipping" blocks through this connection simplifies the network, and optimizes the learning process [17].

The changes in these blocks, as well as the performance boost made by the improvements done on V2 can be seen in the Figure 2.1 and Figure 2.1.
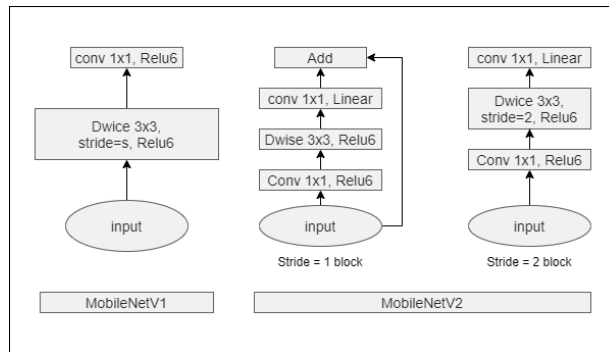
Figure 2.1: MobileNetV2 Architecture [17].

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	**3.4M**	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NesNet-A	74.0	5.3M	564M	183ms
MobileNetV2	**72.0**	**3.4M**	**300M**	**75ms**
MobileNetV2 (1.4)	**74.7**	6.9M	585M	**143ms**

Table 2.1: MobileNetV2 performance.[17]

Resnet

When it comes to object classification, and object detection, a pattern is that
often networks get deeper and deeper to be able to offer more accuracy. One trade
off here is that the extra depth worsens both size and speed, and also that the
training accuracy can saturate and quickly degrade [18]. Resnet was introduced in
2015, and proposed a design to combat these problems, with a goal of being able
to optimize training for deeper networks, allowing more options when designing
it's architecture, and with the extra depth hopefully leading to and increase in
accuracy [19].

The solution that Resnet introduced was to add an identity mapping as a residual
connection within blocks to allow gradients to propagate without vanishing due
to the functions within the block. As is visualized in Figure 2.2, that connection
simply adds the input onto the output. If the dimensions of the output are dif-
ferent than the input, a weight is simply added to the identity mapping to match
the change.
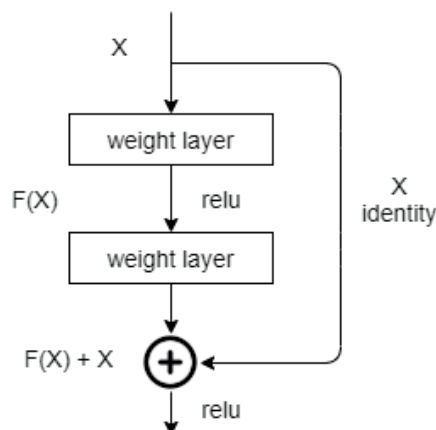
Figure 2.2: Residual learning: a building block [19].

2.2.3   Region Proposal

Selective Search

Selective search utilizes segmentation and exhaustive search to select regions of in-
terest, i.e. potential object locations. Selective search brings forth various regions
by segmentation. Each region is related to one object. A recursive method is used
to merge regions based on parables as colour, texture, intensity etc. Through the
merging of regions the amount of clusters will decline and reveal potential object
locations [20].

Region Proposal Network

The RPN is described according to [13].

Region proposals are generated by sliding over the output feature map generated from the most recent shared convolutional layer. At each location, k proposals are predicted. Each prediction is associated with an anchor box. Anchor boxes, which are predefined and of different size and aspect ratios are applied in every location. The number of anchor boxes is arbitrary, however the default number is 9, 3 x 3 different sizes and aspect ratios. During training an anchor box is regarded as a positive match, with reference to being an object, by maximum IoU or a certain threshold in relation to the ground truth boxes is achieved (default 0.7). A negative match occurs if the IoU in relation to the ground truth boxes is less than a certain threshold (default 0.3). The anchor boxes which do not belong to either category are ignored during training. A loss algorithm is being applied with regard to positive and negative matches. The output of the RPN is a classification, objectness score, of being an object or not and a bounding box regression offset for every proposal. NMS is typically used to reduce redundancy [13].

2.2.4   Classification algorithms and general concepts

Non-Maximum Suppression

NMS addresses the multiple boxing issue i.e. one object or region is represented by multiple boxes. To determine if multiple boxes occur, IoU is calculated for every detection in relation to all other detection's and if it exceeds a certain threshold value, it should be considered as a multiple box otherwise it should be discarded. IoU is a value that reflects the degree of overlapping regions. If multiple boxes representing the same object or region are present, to avoid redundant detections, the confidence score will then decide which is the best option [21].

Support Vector Machine

The following section describes SVMs based on [22] and [23].

SVMs is used to classify data points and is a linear binary classifier. SVMs take advantage of support vectors to find an optimal hyperplane, decision boundary, to classify data points. Support vectors are constructed by data points not easily separated / classified, i.e. closest to the decision boundary [22].

The data points are made up of a tuple (x_i, y_i) where i indicate the specific data point. The x_i is a vector and y_i is a class indication 1 or -1. The aim is to find the maximum margin hyperplane separating class 1 from -1. The output of multiclass SVM is visualized in Equation 2.1, where k represents the class specific SVM for a set of data points x. The w is a normal vector. In Figure 2.3 an illustration is presented which visualizes the use [23].

$$a_k = w^T x \qquad\qquad (2.1)$$
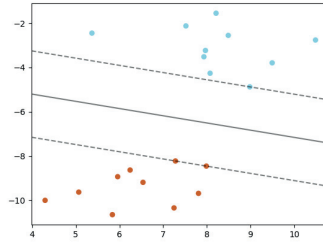
Figure 2.3: Maximum margin SVM [22].

Without the use of support vectors, the decision boundary can simply be any decision boundary that separates data points. Figure 2.4 illustrates decision boundaries without the use of support vectors. Issues occur when further data points are to be classified, hence the data points can easily be misclassified [22].
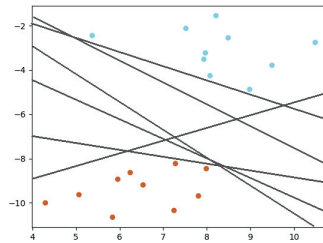
Figure 2.4: Possible decision boundaries [22].

Softmax Function

The Softmax function aligns with the amount of classes and is a multi-class classification algorithm presenting the discrete probability of each class appearing in the image. The probability distribution adds up to one and is visualized in Equation 2.2 for every class $i = 1...n$, where n is the number of classes [23].

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad\qquad (2.2)$$

Logistic Function

The independent logistic classifier is based on the sigmoid function which assumes independence between classes [24]. In Equation 2.3 an illustration of the sigmoid function is visualized, where X is the input feature and w is a weight parameter updated for every iteration in order to establish a relationship between the input features and label [25].

$$P(Y = 1|X) = \frac{1}{1 + e^{w_0 + \sum_{i=1}^{n} w_i X_i}} \qquad (2.3)$$

2.2.5 Dataset

Imagenet

Imagenet is a project working to create a database similar to that of wordnet, but for images. This means labeling and categorizing images and organizing them hierarchically, examples of this hierarchy can be seen below. The project was first introduced in 2009, and had at the time gone through 3.2 million images in total [3]. Since then, the project has kept going, and has as of today gone through over 14 million images [26].

mammal $\rightarrow$ placental $\rightarrow$ carnivore $\rightarrow$ canine $\rightarrow$ dog $\rightarrow$ working dog $\rightarrow$ husky

vehicle $\rightarrow$ craft $\rightarrow$ watercraft $\rightarrow$ sailing vessel $\rightarrow$ sailboat $\rightarrow$ trimaran

Usually when speaking of Imagenet in the context of computer vision and what a model has been trained on, the dataset is subset of the much larger Imagenet. The most common subset is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 image classification and localization dataset. This subset of Imagenet has over 1.2 million images, and classifies 1000 object for detection[27].

COCO

Common Objects in Context (COCO) is a dataset first introduced in 2014, consisting of 2.5 million labeled instances in 328 thousand pictures, and aims at addressing problems in computer vision. A major part of computer vision is to understand the entire scene that is visible. However, it requires not only detecting objects within a scene, but also recognizing the relationship between the objects, here COCO is

focusing a lot on segmentation of entire scenes rather than just dominant objects. And while other datasets at the time provide what's needed to train a system to recognize objects in iconic views, where the object is square in the picture with good focus and no obstructions, they "*struggle to recognize objects otherwise – in the background, partially occluded, amid clutter – reflecting the composition of actual everyday scenes.*"[2].

The COCO dataset includes images consisting of objects in their natural use [2]. Objects from different angles and usage are included in order to be able to detect and/or recognize objects in different contexts.

2.3   Classification measurement and assessment

The following section describes various classification metrics based on [28].

Classification measurement values area of use is widespread and various metrics exist to evaluate different attributes. The metrics can be traced to areas and a variety of sub-areas that include classification algorithms and are used to assess the output.

The Accuracy (ACC) is calculated as the sum of True Positives and Negatives samples, that is samples that yield a correct classification, divided by the total number of samples as visualized in Equation 2.4. The ratio shows the proportion of correctly classified samples one can expect from a set of samples, neither more nor less. Other factors of interest such as the size of the error and whether it was due to overestimation or underestimation are not included [28].

$$\frac{True\ Positive\ +\ True\ Negative}{True\ Positive\ +\ True\ Negative\ +\ False\ Positive\ +\ False\ Negative} \quad (2.4)$$

The precision ratio results in a probability that indicates if a positive classification occurs, what is the probability that it is in fact a true positive. For example if the precision rate is 70 percent and a person is classified as a person, there is a 30 percent risk that this is not in fact a person, i.e a false positive. The precision equation is visualized in Equation 2.5 [28].

$$\frac{True\ Positive}{True\ Positive\ +\ False\ Positive} \quad (2.5)$$

The recall ratio results in a probability that indicates if a negative classification occurs, what is the probability that it is in fact a true positive. For example if the

recall rate is 70 percent and a given area is filled with people, 30 percent of the people will not be classified as persons, i.e. false negative/negatives. The recall equation is visualized in Equation 2.6 [28].

$$\frac{True\ Positive}{True\ Positive\ +\ False\ Negative} \tag{2.6}$$

The F1-score address the trade-off between precision and recall. The optimal outcome is high precision and recall but the two metrics often affect each other. By manipulating classification thresholds, precision or recall can be prioritized and hence support the intended area of use. The F1-score equation is visualized in Equation 2.7 and represent the harmonic mean of precision and recall [28].

$$2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{2.7}$$

The trade-off between precision and recall could also be visualized by a Precision-Recall (PR) curve which consists of x, y - axis, where the x-axis represents recall, and the y-axis precision [28].

2.4 Cloud platform / API

Google Cloud Platform (GCP) offers a large selection of services, and for our use case, a serverless option seems good since we will only use it for computations upon making requests. With that in mind, we have some options to choose between.

2.4.1 Cloud Functions

Cloud functions is a server-less, event-driven option, which takes code packaged in a function. After deploying your code to GCP you can then configure it to listen for Hypertext Transfer Protocol (HTTP) request. You can also configure it to run your function when different events happen in your GCP project, for example, when a picture is uploaded to GCP's cloud storage. This approach does however come with some constraints, the biggest one being of course that you need to package all code in a function, but also that it only supports a handful of languages [29].

2.4.2 Cloud Run

Cloud run is a service that offers the possibility for GCP to run and host your containers. So if you build a container that exposes a service to a given port, you can use GCP's Cloud Run to now run and host said container and expose it on a Hypertext Transfer Protocol Secure (HTTPS) endpoint for incoming requests. And deploying containers with cloud run, you can also choose for GCP to

completely manage the deployment, and if wanted, GCP will also auto-scale your deployment to handle an increase in demand, or shut it down completely to save cost if no traffic is incoming [30].

2.4.3   App Engine

App Engine will give you a platform to run your code on. To use App Engine you can simply write your code, and deploy it. App Engine can then stand for the management. If you only want to deploy code to App Engine, you are limited to Node.js, Ruby, C#, Go, Python, or PHP. But App Engine also lets you provide your own language runtime by supplying a Docker container that runs your application, and similarly to Cloud Run, App Engine can then use that to deploy and manage your application. App engine, just like cloud run, can also manage auto-scaling, to make sure that any increases or drops in incoming traffic is handled [31].

2.4.4   Compute Engine

Another usable service provided by GCP is the compute engine, which will provide you with infrastructure to run your code in the form of a Virtual Machine (VM) that is then running in the cloud. GCP will give you full access to this VM, so you decide yourself what to install on it, which means you also have to manage it. You may also decide for yourself how you want your VM to look, deciding things like memory, processing power, OS, and storage. Pricing on a computer engine depends on all of these choices, giving you a lot of control [32].

Google also offers a free pricing tier for most of their GCP tools, where it won't cost you anything if you keep under certain quotas when it comes to the different priced resources.

2.5   Raspberry Pi

Raspberry Pi is a SBC . With the support of some additional hardware components, a power supply and a micro SD card with a supported operating system as Raspberry Pi OS (or any other supported distribution) a cost effective, power efficient, small and portable fully working computer is the result [7].

Raspberry Pi OS is based on Debian and is supported with a large number of packages [33]. The possibility to integrate customized software but also existing is one of the advantages with the Raspberry Pi OS. The operating system is installed with a set of programs that can help with interaction, but additional packages that may be needed can be installed semi-manually using the Advanced Packaging Tool (APT) or manually as long as it is compatible with the Advanced RISC Machines (ARM) architecture [34].

Additional hardware as mentioned a power cable and micro SD card is needed

in order to supply power and install Raspberry Pi OS respectively. Opportunities to purchase a pre-installed Raspberry Pi OS micro SD card are available. Another option is to purchase a micro SD card and install the Raspberry Pi OS single-handed [35]. When the Raspberry Pi OS is utilized, interaction with the Raspberry Pi demands either an initialized network connection with pre-configured connectivity options which enable SSH for remote connection or a combination of monitor and keyboard and/or mouse. A Graphical User Interface (GUI) with the most common configuration changes can be accessible from the command raspi-config, where for example WiFi can be made available [36].

Raspberry Pi satisfies several ports, outlets and I/O pins depending on the version. The Raspberry Pi 4 entails one USB-c power supply, two micro HDMI ports, two USB 2 ports, two USB 3 ports, Gigabit ethernet outlet, micro SD card outlet, 40 I/O pins with several different uses and more [37].

The qualities of the Raspberry Pi combined with the Raspberry Pi OS makes it an attractive component within embedded systems and/or as an Internet of Things (IoT) device [38]. Since Raspberry Pi can be used in embedded systems a vast majority of additional hardware support the use of the associated operating system. Drawbacks of a fully integrated operating system is the delay it may introduce in comparison with other IoT devices.

2.6   General Data Protection Regulation

Camera surveillance has statutory consequences if the General Data Protection Regulation (GDPR) is not complied with [39].The ethical implications of a product / service a developer or a company should consider is reinforced by a designed regulatory framework. GDPR was formed to maintain privacy and is therefore enforced by law. The main concern is that collecting data provides the opportunity for secondary usage.

Because of the advancement in technology an image of a person could be enough to fully identify a person. Hence, one must consider the law related to the collection of images that may be enough to identify a specific person. Hefty fines await companies that do not comply with the rules related to the GDPR [40]. GDPR only apply in the EU and because several companies in the digital sector operate globally negative implications may arise. Direct consequences such as development and data management taking place outside the EU to circumvent the regulations may occur. The transfer of data from the EEU is protected by GDPR, however further implications may involve that certain products and services are made inaccessible to and within the EU [41]. The positive consequences may be that the global platform is forced to comply with the GDPR and thereby maintain integrity in a digital era.

Approach

The chapter describes the approach, from idea to execution. The paper consists of two phases, system development and object detection model evaluation. The system development was performed and designed to promote an interchangeable object detection model. The sections 3.1 and 3.2 emphasize the overall system architecture and software design. Succeeding section 3.3 introduce the three different models selected for evaluation. Finally the three remaining sections 3.4, 3.5 and 3.6 present the joint calibration of cameras, environmental setup where the testing was performed and a descriptive approach regarding the testing in different phases.

3.1 System Architecture

The basis of the system will rely on technical equipment acquired, thus, it is essential in several aspects. The technical selection entails limitations but also opportunities and creative thinking. The system will be made up of various components, each component intended for one or more specific purposes.

The high level architecture, see Figure 3.1 visualizes the initial system architecture idea and operation in an arbitrary environment. The two cameras, where one camera produces RGB images and the second camera produces thermal images, forms the basis of the information flow. The RGB image is used to detect the number of people using the object detection models. The thermal data will validate if a detection is reasonable, by observing the temperature at a given position obtained from the RGB image. To establish a communication pipeline between the cloud and the cameras a Raspberry Pi is to be used. The Raspberry Pi manages the data captured by the two cameras and sends data that requires more computational power to the cloud. A further analysis, using an object detection model, is performed in the cloud in order to determine the number of people located in a given area. Once the analysis is performed the information is stored. An application can be used to access the stored information, the current state, and present it to the end user. The end user will be able through the presented result in the application to observe the number of people in a given area.
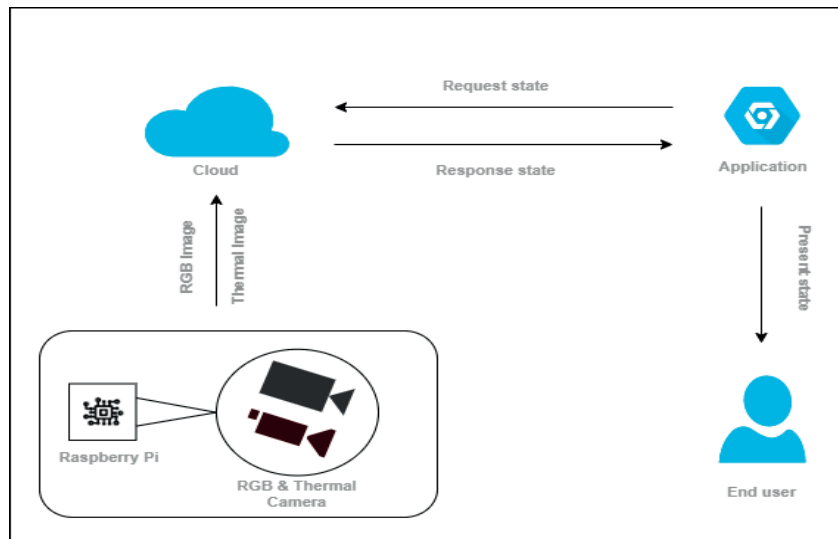
Figure 3.1: High level architecture

3.1.1   Hardware List

Raspberry Pi module and associated components

- Raspberry Pi 4: Module

- Raspberry Pi 4: Power supply 5.1V/3A

- Micro HDMI cable

- Micro SD: 16GB with Raspberry Pi OS

- Protective case: Raspberry Pi 4 module with fan

- Various connection cables

Cameras and Others

- RealSense camera: Depth D435i

- FLIR Lepton camera: Breakout board v2.0

- Arbitrary computer

3.1.2   Cameras

The system uses two cameras, one captures the RGB video feed and the other thermal. The data will be used to detect the number of people by applying an object detection model that processes the RGB data and validate a detection using thermal data.

3.1.3   Raspberry Pi

The cameras will provide a stream of data / images, the management of the data stream at the edge of the technology will be obtained by using a Raspberry Pi, more specifically Raspberry Pi 4, described in section 2.5. Simpler tasks will be performed and more resource-intensive, computational power demanding tasks will be delegated to the GCP. Attributes that Raspberry Pi 4 possesses and that are essential are a variety of sockets, WiFi, the ability to perform simple calculations and decide which data to send to the GCP and more importantly when to send data. A constant flow of data to the GCP is not desirable, since unnecessary calculations would be carried out which increases cost. An analysis in the GCP should occur if the environment has changed, but observing if an environment has changed can be made closer to the environment. Different solutions of how a potential change in environment can be observed by using the processor in the Raspberry Pi alone, hence with minimal effort, are devised and feasible. Examples of solutions could be change in depth of the RGB images or change of heat in the thermal images / data.

3.1.4   Google Cloud Platform

The GCP is the cloud platform used, introduced in section 2.4. Given that the Raspberry Pi has limited resources, the GCP holds a large area of responsibility in the system having to perform the heavy computations, e.g. running an object detection machine learning model. Running in the GCP is an open Application Programming Interface (API) to call with a RGB image and associated thermal data packaged in a request. The application running on GCP will extract the data, make a request to an additional API on GCP that hosts the machine learning model, store the current state and respond with success or failure.

3.1.5   Simple Application

A simple application that will request and update the current state.

3.2   Implementation

The software challenges which can be the result when several technical components are to work together is for instance compatibility, how to deal with interchangeable platforms and the physical limitations the hardware may impose. In addition information that is extracted will determine the conditions for the object detection. An initial design of the system is the basis for how the implementation is to be carried out, see Figure 3.2. The different components and areas of responsibility are isolated to individual classes in order to facilitate changes and also minimize the coupling between subsystems.
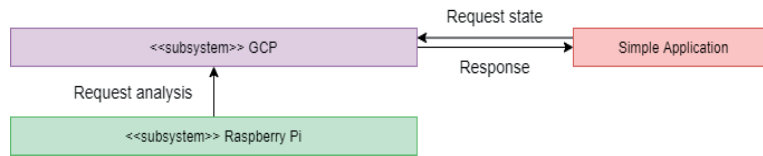
Figure 3.2: Software overview

3.2.1 Raspberry Pi Subsystem

The subsystem Raspberry Pi uses the programming language Python, locally in the area of use, i.e. an edge device. The subsystem is visualized in Figure 3.3.
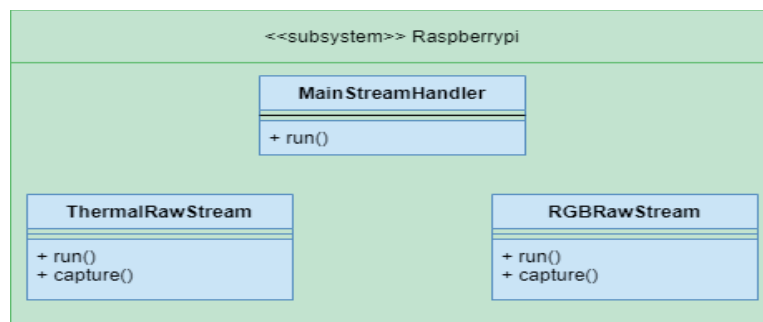

Figure 3.3: Raspberry Pi subsystem

RGB camera

The RGB camera offers the ability to connect via USB. the manufacturer provides a library of Python-based code that can ease implementation. An open-source computer vision library is used to manipulate the stream output which includes useful functionality. Different methods can be used to extract the x,y,z coordinates for a given coordinate which is feasible because the RGB camera includes a depth sensor. The object detection algorithms output a bounding box region in the image which includes a detection, i.e. a person. From the bounding box specific coordinates can be extracted or alternatively an area which can be used for further validation with the thermal camera.

Thermal Camera

The thermal camera is connected to a breakout-board. The breakout-board is connected to the Raspberry Pi via I/O pins. Python-based code is used to extract

the thermal data from the pins. Documentation and guidelines are made available by the manufacturer.

Raspberry Pi

Raspberry Pi is set up with a simple Integrated Development Environment (IDE) based on Python and as a powerful programming language the potential is limited mainly through what is feasible for the processor. Simple tasks are implemented locally in the Raspberry Pi OS. The simple tasks include software that set up a stream from the two cameras and monitors the output. The output from the RGB camera, the data, is not directly relevant information locally on the Raspberry Pi. Thermal data that originate from the thermal camera stream are highly relevant. From the thermal data a summation of the number of hot spots can be made and if it differs more than a specified threshold it triggers an action. The action which is triggered when a change occurs in the present environment forward the most recent RGB image and associated thermal data to the cloud.

Classes

This section describes how the software running on the Raspberry Pi has been split up, and what the different classes are used for. This can be seen in 3.3.

The MainStreamHandler class initiates the main loop of the subsystem Raspberry Pi with functionality such as retrieving information from the ThermalRawStream and RGBRawStream classes and making a POST request to the cloud API. By continuously retrieving the summarized temperature values from the Thermal-RawStream a decision can be made about potential environmental changes. If an environmental change has been taking place the most recent image from the RG-BRawStream and the most recent data from the ThermalRawStream is retrieved and a POST request is sent to the cloud for further analysis.

The ThermalRawStream class responsibility is to establish a connection to the thermal camera and retrieve the data. The data consist of a matrix containing values representing the temperature in a specific area of the image. The class possesses functionality to summarize all the values to recognize if a change in the environment has occurred and notify the MainStreamHandler class if it has changed. Furthermore the class can satisfy the need to retrieve the most recent data.

The RGBRawStream class responsibility is to establish a connection to the RGB camera and on occasion capture the most recent image. The class uses the manufacturer's software library to enable and establish a connection. To handle the data an open-source computer vision library is used.

3.2.2 Google Cloud Platform Subsystem

The software running in GCP is set up using their Cloud Run service, two different docker containers are given to the service, which hosts and exposes HTTP

endpoints. The first one is a Flask application that presents the user with a simple web page to upload pictures stored locally. When uploaded, it takes the picture data, performs some necessary preprocessing on the image, and sends it onward to the second endpoint.

The second endpoint is a docker container running TensorFlow serving docker containers. Meaning it takes a model saved in TensorFlow's 'save model' format, and exposes it on a port. Pushing this container to GCP's Cloud Run service and expose it on a public HTTP endpoint for easy access. When it is serving a model, it takes image data as input, and will output a list with detections made, their score, and their locations.

3.2.3   Simple Application

The arbitrary application simply request the current state from the cloud and present it in text format to the user.

3.3   Object Detection Models

The different detection networks selected for comparison are SSD, Faster R-CNN and YOLO-v3 which general concepts are described in section 2.2.1. The base networks and datasets are presented in 2.2.2 and 2.2.5 respectively.

3.3.1   SSD Model

The SSD model consists of the base network mobilenet v2 and is trained using the dataset COCO 2017.

3.3.2   Faster R-CNN Model

The Faster R-CNN model consists of the base network resnet50 v1 and is trained using the dataset COCO 2017.

3.3.3   YOLO-v3 Model

The YOLO-v3 model consists of the base network DarkNet53 and is trained using the dataset Imagenet.

3.4   Joint Calibration of Cameras

To achieve joint coordinates through conversion, fixed and unalterable mounting of cameras is assumed. The assumption generates an inflexible solution with regard to that more cameras are added to the system, however if more cameras are added to the system more issues will follow. For instance multiple representation, as mentioned in section 1.1.1. The result is visualized in Figure 3.4. A small deviation occurs between the images but it is negligible.
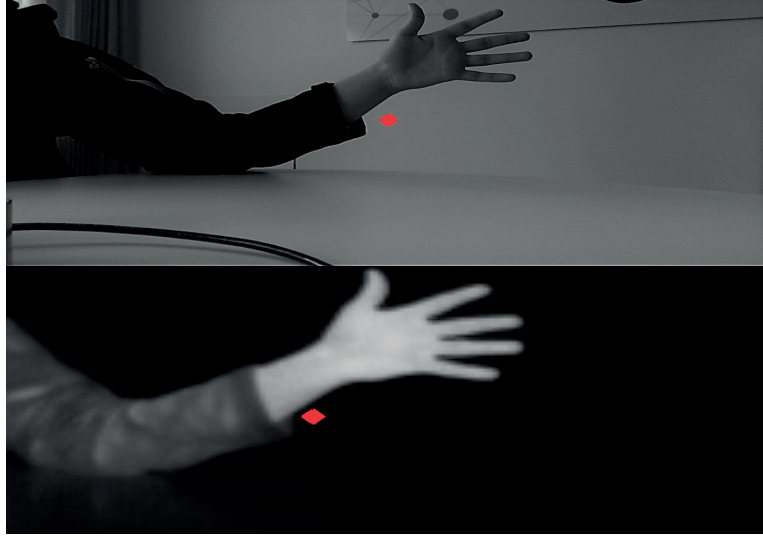
Figure 3.4: Joint calibration images: Top RGB & Bottom thermal

The mathematical formula is visualized in Equation 3.1 and Equation 3.2. The conversion procedure is as follows:

$$Resolution\ conversion\ width\ (RCW) = \frac{Resolution\ 1\ width}{Resolution\ 2\ width}$$

$$Resolution\ conversion\ height\ (RCH) = \frac{Resolution\ 1\ height}{Resolution\ 2\ height}$$

$$x' = (\ x \pm DX\ ) \times\ RCW\ , where\ DX = Deviation\ in\ the\ X-axis \qquad (3.1)$$

$$y' = (\ y \pm DY\ ) \times\ RCH\ , where\ DY = Deviation\ in\ the\ Y-axis \qquad (3.2)$$

3.4.1   Thermal Data

The primary use of the thermal data is to distinguish a true positive detection from a false positive. Representations in human form, whether it is a painting or

a mannequin, should not be taken into account. With the help of joint calibration, an area related to a detection can be located and thermal analysis can be performed. The idea is based on using a threshold value, heat in a given point or area, to determine if it is a true positive or false positive.

Secondary use of the thermal data is to summarize the total amount of heat values in order to detect a change in the environment and thereby trigger an API call that analyzes the area and updates the current state.

3.5   Environmental Setup

This section focuses on the experimental area where the testing is conducted. What can potentially be considered to limit the outcome is presented. Also the different alternatives that is possible and why a certain approach was prioritized in the environmental setup is described.

The testing area is a conference room with the measurements 500 cm x 550 cm. Having only one camera angle available makes full surface coverage difficult and with components available not reasonable.

The area is rectangular and different angles, A, B, C and D, reflect different areas of interest visualized in Figure 3.5. Desirable attributes is an overview of the conference table, presentation/white board and minimize external information. All the angles meets an overview of the conference table. A and D does not include the presentation/white board and is therefore excluded. B include windows facing the outdoors and C include corridor windows facing indoors. C will have a higher risk of including external information, hence B is the best option for the given criteria.

A wide-angle lens is used in both the thermal and RGB camera to allow for as big of a Field of View (FOV) as possible, see Figure 3.6. Blind spots are difficult to avoid with the equipment used and in some cases a person will only be partially visible. Desired is a model that is able to detect a person even though, for example, the head is not visible.

3.6   Testing

The testing section present the approach of the testing, while the results and additional materials is available in the Chapter Result and Appendix.

The project is designed and constructed without a frame of reference but with a basic idea in mind. A detailed testing is therefore necessary and helpful to ensure that all parts are functional and behave as intended. The section is divided according to the chronological order in which the various components is implemented and tested. Component and subsystem testing is the first step followed by a complete system test. Finally, different object detection models is tested, with
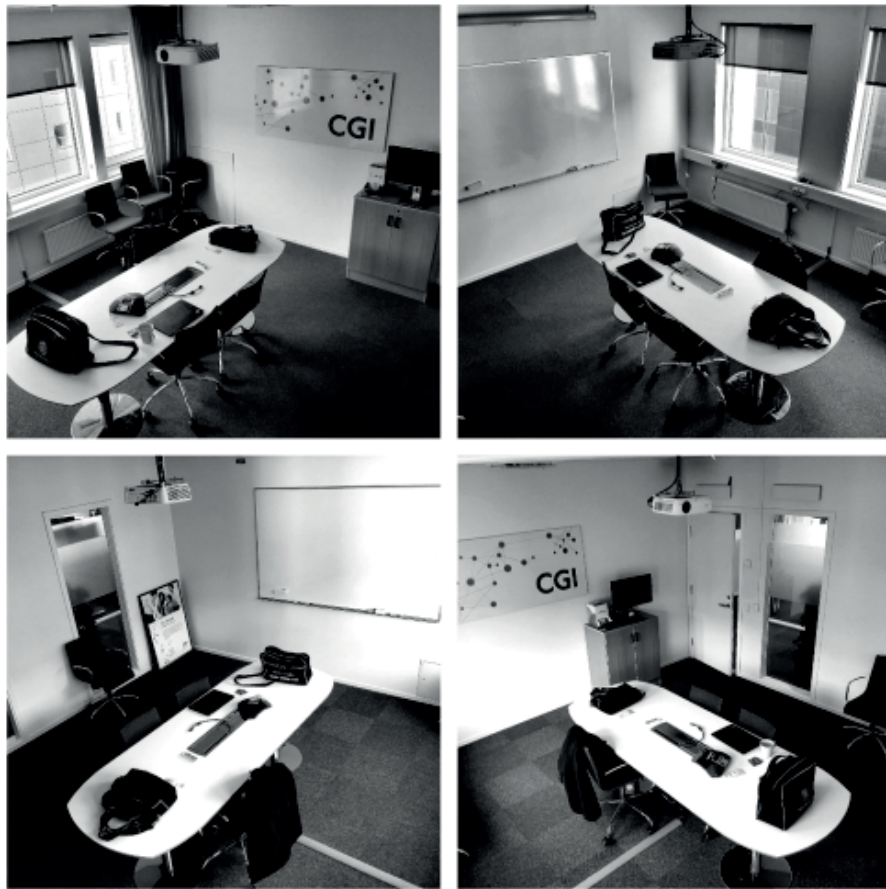
Figure 3.5: Environment angles: A) top-left, B) top-right, C) bottom-left & D) bottom-right.

Figure 3.6: RGB camera view

respect to classification metrics introduced in section 2.3. The object detection models is tested with and without an additional thermal component.

Testing and its scope are significantly limited by the current state of the world, that is, the current pandemic. The number of people in a given area are limited and must be complied with for everyone's safety and statutory provisions.

3.6.1 Components and subsystem testing

The test cases start from the basic level, it refers to whether a component is functional and what is required to extract data if necessary. The external test components are thus not interconnected but are tested separately. When achieved results are satisfactory, they are integrated in the system.

3.6.2 System testing

A full-scale test where the system functionality is tested in the presented environment, section 3.5. The different components must be adapted to the environment, in regard to the physical location and angles of the cameras for optimal results. Previous tests have confirmed that a functional flow is achieved and data is received. The system test make sure that all the components work together to achieve satisfactory results.

3.6.3 Object detection

This section will describe the test phase of the different object detection models and the outcome will be presented in the section 4, that is how well they perform in a limited area with people in circulation. The testing will visualize how well the model detects people without the support of an external component, but also include separate test data using an external component. The external component will provide heat data which can be used to validate whether a detection is reasonable or not.

Test data

The generation of samples transpire during a time interval where several people is circulating in the given room described in 3.5. The number of people during the time period varies. The maximum number of people is 4, the minimum 0, as to follow the pandemic restrictions. The outcome of the process is 80 samples. A sample includes a RGB image with a resolution of 1280x720 and associated data from the thermal camera.

Object Detection Models

The test data is used by processing the collection of samples for each model and observing the proposed number of people from the detection model compared to the actual value in the given samples. Different metrics, presented in section 2.3, are used to evaluate the models. Initially, testing is to be performed with only the

object detection models to see how well the model performs without an additional component and thereby enable a comparison on a model basis.

Object Detection Models with external component

The object detection models with the support of external thermal data provides insight into the potential for improvement. Additionally various thresholds can be weighed to obtain an optimal outcome. Incorrect detections presented in section 3.4.1 should not be included, but are difficult to filter out in object detection models without affecting its general detection capability. Further uses such as confirming a change in the area can be investigated to minimize the number of requests to the cloud platform.

3.7 GDPR

The system is designed and implemented in a way that images will never be saved in any kind of database. Storing images may contravenes legislation, GDPR, as stated in section 2.6. When an image is taken by the camera and loaded into the software running on the edge device, it will send the image data to the application running on the cloud platform, after sending the data, the image will be removed locally. On the cloud, the image is read, preprocessed, and forwarded to the model API.

When the detection model API, the one serving the TensorFlow model, receives the image data it will preprocess the image, run it through the model, and output information regarding detections. Nowhere in the process will the image be stored. During development and testing of the system, images is stored to easier evaluate the results, it is done with the knowledge and consent of people involved, but does not reflect how the system would operate in an actual deployment.

Result

The result section mainly presents overall experimental data and measurement values for the different models with and without the use of external thermal data. Additional in-depth data that can be used to derive incorrect classifications will also be visualized. The same developed core system is used for the different object detection models and a short introductory part introduce the functionality of the system.

4.1 System

The system is highly functional and the system design allows replacement of model. The result of the components and overall system functionality is presented in the Appendix Table A.1. The system introduces time delay because it involves external API calls. Priority of a functional system that meets the specified requirements results in that other potential shortcomings have not been reviewed.

4.2 Object Detection Models

As stated in section 2.2.1 the standard implementation of a detection network may differ significantly from an optimized version as the possibility to replace the base network, modify parameters, etc. is possible. The following object detection models are evaluated in the paper:

- SSD MobileNet v2 dataset COCO 2017 referred to as SSD

- Faster R-CNN Resnet50 v1 dataset COCO 2017 referred to as Faster R-CNN

- YOLO-v3 Darknet-53 dataset Imagenet referred to as YOLO-v3

Visualized in Figure 4.1 and Table 4.1 is the correct versus incorrect classified samples of the different object detection models in the given environment.

The dataset used for testing consist of 80 samples. More detailed information about the samples is described in section 3.6.3.
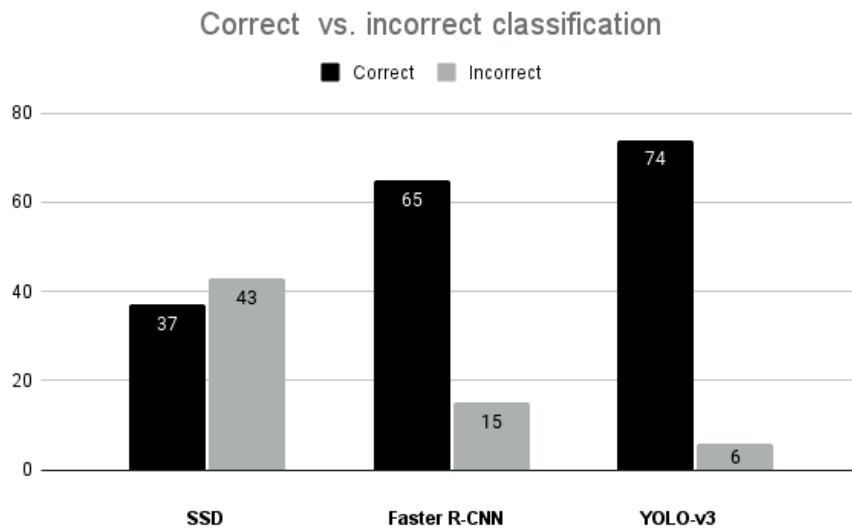
Figure 4.1: Correct vs incorrect classification without external thermal data

Object Detection Model	Correct	Incorrect	Total samples
SSD	37	43	80
Faster R-CNN	65	15	80
YOLO-v3	74	6	80

Table 4.1: Correct vs incorrect classification without external thermal data

4.2.1   Reason of error

The reason for the errors is a result of manually reviewing each image that contains an error for each separate model and drawing a conclusion as to why an incorrect classification occurred. Since the analysis is performed by a human, a certain margin of error may exist The different categories are camera coverage, lightning condition, multi boxing and error. Camera coverage occurs when the person is only partially visible in the image, the positioning of the person is not favorable and potential blind spots. Lighting conditions is the impact of, for instance, reflection and brightness. Multi boxing refers to when a person is detected multiple times. The simple "Error" occurs when a derivation to the above categories is not possible, hence it is to be considered the most critical reason. Because it is unknown what lead to the incorrect classification.

The distribution of errors and the reason thereof is essential, but to avoid misleading statistics, the data on which the distribution is based on is presented as a supplement for each model.

SSD

In Figure 4.2 the diagram shows that 59.6 percent, i.e. a majority, of the incorrect classifications could be derived to camera coverage. Followed by lighting conditions 27.7, error 8.5 and multi boxing 4.3 percent. In Table 4.2 the data is presented in which the distribution is based on.
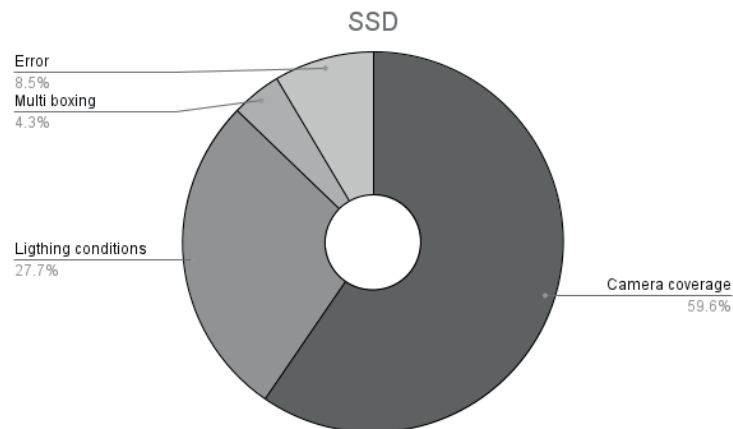
Figure 4.2: Incorrect classification distribution SSD

Reason of error	Quantity
Camera coverage	28
Lighting condition	13
Multi boxing	2
Error	4

Table 4.2: SSD reason of error, quantity.

Faster R-CNN

In Figure 4.3 the diagram shows that it is an even distribution of camera coverage, multi boxing and lighting conditions. In Table 4.3 the data is presented in which the distribution is based on.
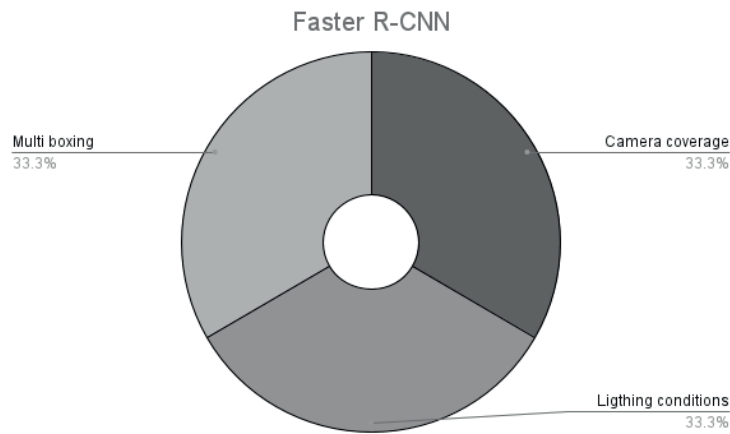
Figure 4.3: Incorrect classification distribution Faster R-CNN

Reason of error	Quantity
Camera coverage	5
Lighting condition	5
Multi boxing	5

Table 4.3: Faster R-CNN reason of error, quantity.

YOLO-v3

In Figure 4.4 it is visualized that the most common error, almost exclusively, is due to the lack of camera coverage. In Table 4.4 the data is presented in which

the distribution is based on.
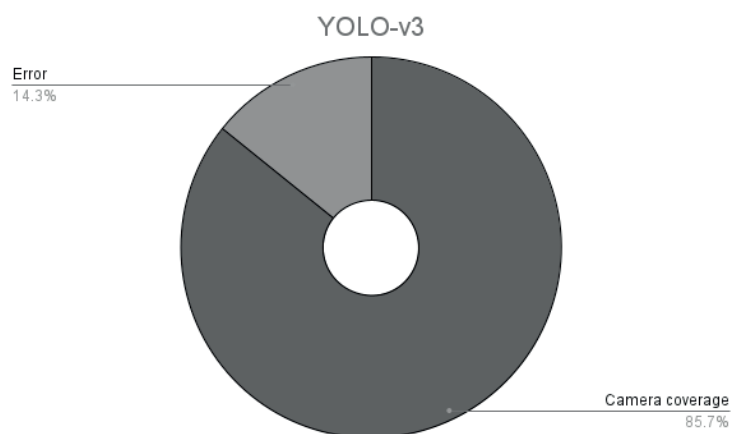
Figure 4.4: Incorrect classification distribution YOLO-v3

Reason of error	Quantity
Camera coverage	6
Error	1

Table 4.4: YOLO-v3 reason of error, quantity.

4.2.2 Overall performance

The overall performance of the different models are evaluated by metrics shown in Table 4.5. The processing speed consists not only of model analysis, but a complete system cycle. The course of events begins when an image is sent to the cloud, it is then analyzed by the model and finally an estimated value is returned. As the course of events does not differ from the different models more than the model itself, the choice was made to use the entire cycle.

In Figure 4.5 the PR curve is visualized.
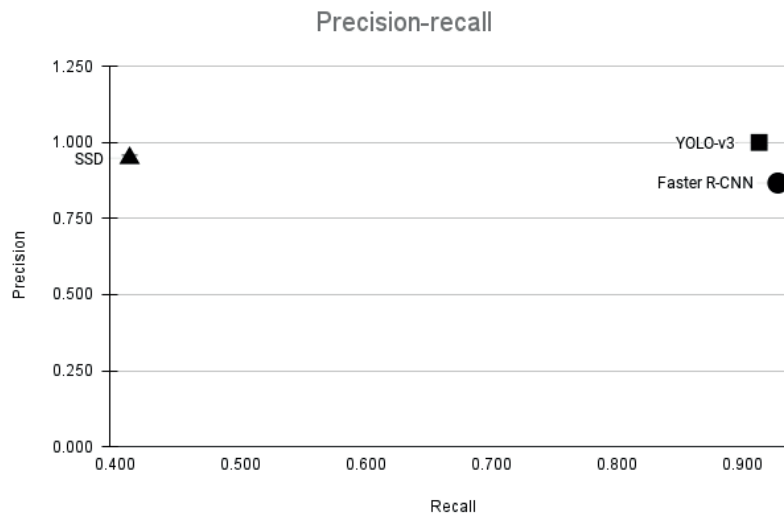
Figure 4.5: Precision-recall

Object Detection Model	ACC	Recall	Precision	F1-score	Sec/image
SSD	0.463	0.411	0.949	0.574	1.382
Faster R-CNN	0.813	0.929	0.867	0.897	3.761
YOLO-v3	0.925	0.914	1.000	0.955	1.805

Table 4.5: Overall measurements

4.2.3 Improvement Using External Thermal Data

The only aspect whereas the external thermal data can be used is related to reflection. Incorrect detections do not occur in other forms on the test data used.

In Figure 4.6, Table 4.6 and Table 4.7 the overall improvement of Faster R-CNN is visualized. Concerning the other models, SSD and YOLO-v3, reflection is not detected. The improvement of Faster R-CNN

Figure 4.6: Correct vs incorrect classification using external thermal data

Object Detection Model	Correct	Incorrect	Total samples
SSD	37	43	80
Faster R-CNN	70	10	80
YOLO-v3	74	6	80

Table 4.6: Correct vs incorrect classification using external thermal data

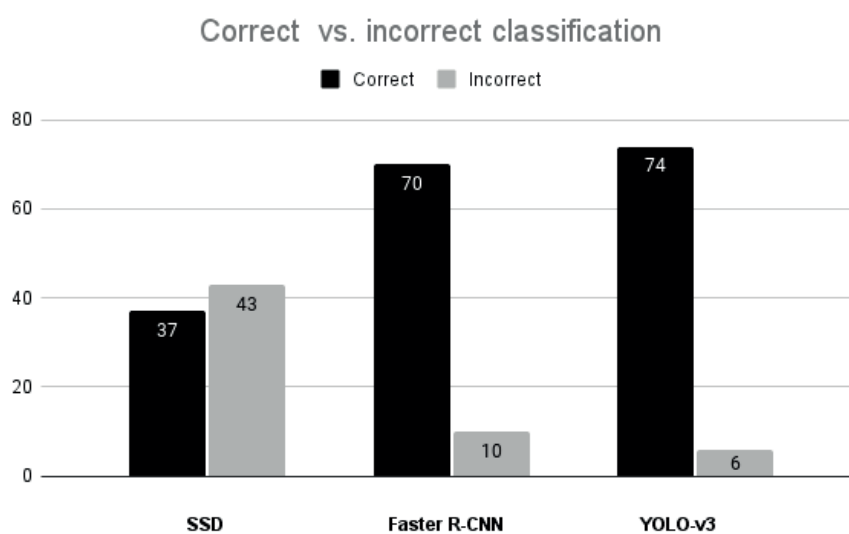| Object Detection Model	ACC	Recall	Precision	F1-score	Sec/image
SSD	0.463	0.411	0.949	0.574	1.382
Faster R-CNN	0.875	0.929	0.933	0.931	3.761
YOLO-v3	0.925	0.914	1.000	0.955	1.805

Table 4.7: Overall measurements

4.3 External Thermal Data

The thermal data is used as an additional measure to filter False Positives. Paintings, mannequins and reflections are examples that can trigger a detection, but which in reality should not be considered as one.

From testing it follows that thermal data can be used to validate a detection by calculating joint coordinates and setting reasonable thresholds values.

In Table 4.8 the threshold values given by the testing are presented. A True Positive can be validated by checking if the temperature at a certain point or range is equal to or exceeds 25 degrees, a steady value that humans were always observed to be above. A False Positive is assumed if the temperature at a certain point or range is equal to or less than 22 degrees. The interval between 22-25 degrees did not occur during testing, but is to be considered as a False Positive. The following thresholds are based on an ambient temperature of 22 degrees and may therefore vary depending on the environment.

Detection	Heat Values
True positive	$x \geq 25$
False positive	$22 \geq x$

Table 4.8: Thermal thresholds

A certain quantity of the thermal data contains inaccuracies. Regardless of errors, the structure and amount of values does not change. The number of inaccuracies is presented in Table 4.9.

Total samples	Inaccuracies
80	31

Table 4.9: Thermal inaccuracies

Even though the number of inaccuracies is relatively large, data can be used in most cases if a detection does not include the affected area.

Discussion

5.1 System

5.1.1 System architecture

When deciding upon the architecture of the system as a whole, there are quite a few factors to take into account. A few were chosen to focus more on than others, mainly to keep the cost down since the end goal is developing a simple proof of concept, with the ability to scale. High processing capability is also of importance to be able to run the models mentioned in section 2.2. While speed might not be essential to the system, since we're not aiming at running the detection in real-time, the Raspberry Pi should still be free to use its resources to interpret the data from the thermal camera. Using the cloud for running the models then works well.

The request to an external API adversely affects speed. In section 4, the seconds per image is presented and it is far from optimal. Even though the speed is not the primary subject of the paper it is a drawback of the system design which contributes to the fact that real-time detection is not possible, hence limiting the area of use.

Using the cloud for the major processing tasks contributes to the scalability of the system, making it easier to manage multiple deployments, e.g. changing your object detection model would only have to be done once on the cloud platform, and not at each deployment.

The inaccuracies, discussed in section 4.3, means that thermal data can not always be used. The thermal data can be used to detect a change in the environment if inaccuracies are not present, thereby reducing the number of requests to the cloud. However due to inaccuracies combined with the fact that the sum of heat values can vary depending on the distance to the device and not only on the number of people, it is not an optimal strategy.

5.1.2 Object detection

The area object detection emphasizes accuracy and speed. The accuracy is not affected by the system design, however the speed is as introduced in section 5.1.1.

An option to overcome the issue, is to try and adopt one of several models that aim to be more lightweight, to run it on site in the edge device. This would remove the considerate amount of time it takes to send the request. Another option is to host your own server in another device on site, but it would come at the cost of scalability.

5.2   Object Detection Models

Results presented in section 4, show that YOLO-v3 is the best-performing model for the generic area of use. Even though Faster R-CNN is relatively close if one ignores the processing speed. SSD however is the fastest model but deficient in the terms of ACC, recall and by extension the F1-score.

In section 2.2.1, the different detection networks are introduced and the attributes described. The object detection models used for testing prioritizes speed and accuracy differently as introduced in section 3.3.

The difference between the three base networks used for the different models may impact the ACC. As mentioned in section 2.1, the degree of complexity in the CNN network will increase the ACC in the general case. Regarding speed, Faster R-CNN is the time consuming option compared to SSD and YOLO-v3. In section 4, the speed, seconds per image, is presented. The speed is not isolated to the model alone, as mentioned in section 5.1, but the complete system running the test data divided by the total number of samples. Regardless, the model is the only component that differs and therefore a comparison can be made.

The SSD base network is MobileNet V2 which is a lightweight base network with few layers compared to Resnet50 and Darknet-53 as presented in section 2.2.2. Since the environment does not involve a lot of traffic or undergoes rapid changes the SSD suffers disadvantage in the area of use. As presented in section 2.2.1, SSD should at least perform as good as Faster R-CNN. The major factor that distinguishes them is the use of different base networks. It is difficult to conclude that with similar base networks, SSD would perform better than competing detector networks. But using a lightweight base network means that SSD cannot compete with detector networks that use complex base networks with regards to accuracy.

The Resnet50 is a less complex network than Darknet-53, however by using a more complex network the speed of the Faster R-CNN will be adversely affected, and since Faster R-CNN is already significantly slower than competing detection networks, a change of base network is considered unreasonable.

The datasets used, COCO2017 and Imagenet can also be a decisive factor. Both datasets are used frequently and no conclusion can be drawn from which is preferable in the test environment. The choice of dataset on which the different models train is therefore not considered to be a decisive factor. Extended training is a factor that has not been tested and that can affect the overall outcome of the

different networks.

The Faster R-CNN network suffer because of multi-boxing, mentioned in section 4.3, which could be optimized by tuning different parameters, however there is a trade-off between recall and precision, as presented in section 2.3, resulting in that it should not perform better than YOLO-v3 overall. The overwhelming speed disadvantage of Faster R-CNN should not be neglected and with that in mind, it can not compete with YOLO-v3. Even though Faster R-CNN is an end-to-end system its architecture negatively affects the speed compared to single forward pass object detectors as YOLO-v3 and SSD.

5.2.1 Evaluation Metrics

The area of use places different demands on the object detection model. The critical metrics vary depending on what is prioritized. Different requirements for example related to speed, potential restrictions and the ability to be accurate or a combination thereof can provide a reference to which model to use. The system is designed according to the principle that the model should be interchangeable.

Recall should be considered the most critical metric when restrictions are applicable to the given area. Restrictions refer to the maximum number of people allowed. As described in section 2.3, if 3 persons are present in the given space but the model only detects 2. A critical problem may be that there is an area restriction of 3 people. Only detecting 2 people and presenting it to the user invites the user to enter the area and thereby violate restrictions, which can lead to fines or, in the worst case, the spread of infection/disease. The precision is not the critical metric in this case, but it is important, especially if one considers the trade-off that exists between precision and recall. However, detecting more people in the area does not signal to the user that it is OK to enter the area, hence the precision is not critical. The F1-score, i.e. the harmonic mean between precision and recall, address the trade-off between the two metrics.

The ACC is not associated with truly critical factors such as recall. ACC may be low, but that does not necessarily mean it is a critical metric. The cause of the error may be due to overestimation or underestimation, which the ACC does not consider. Low ACC results in low reliability and potentially dissatisfied users. Hence, if the business aspect is taken into account, ACC represents reliability and optimal use of resources. From a business perspective, ACC is important.

The processing speed could be a critical factor if the environment in which the system is intended to be used is changeable and time-dependent.

5.3 Thermal data

The impact of the external thermal data is visualized in section 4.2.3. In the environment where the testing was performed, there were only reflections that could be handled using thermal data. The reflection does however show another scenario

where the thermal data is useful. Reflections can occur in almost all areas which signify the benefit of incorporating a thermal camera into a detection solution. As presented in section 4.2.3, there is only one model that detect reflections. The extent of the problem related to reflection has not been investigated and it may be possible that detection of reflections will not occur, if fine adjustment of threshold values is performed.

In alternative environments such as shops, exhibitions, etc., the external data can help to distinguish real people from illustrations of people. The strive of a generic solution made us consider these areas relevant and thermal data can help overcome difficulties that these areas impose. The use of thermal data thus leads in some cases to better accuracy.

The use of thermal data to detect a change in the environment is also helpful in the developed system described in the section 5.1.

The inaccuracies presented in 4.3 causes problems when using thermal data. In our environment, it did however prove easy to spot such inaccuracies, since the values in certain ranges are clearly values that would not be observed in the environment. When inaccuracies are observed by the software, the data in certain ranges cannot be relied upon, and it will have to trust the prediction made by the detection model. The interval consisting of incorrect data is relatively small and there is a low risk that a detection occurs within that interval. The potential environmental changes suffer the most because of the inaccuracies, since the sum of all the heat values will be incorrect which does not allow for the use of the thermal data for these evaluations.

5.4   Cloud Platform

To begin with, there are several cloud platforms to choose from that have the services that are used in the system, Google, Microsoft, IBM and Amazon all offer different platforms. But with Google's cloud platform being recommended, the platform was favored.

As presented in the section 2.4, Google Cloud Platform (GCP) offers several services that can be used to complete the task at hand. With all of the services offering different levels of customization and management options, they all work well, but focus on different scenarios. Between the options that were looked into, the cloud run, and cloud functions seemed like the two services best suited for the task.

And while cloud functions seemed like a perfect fit for the task, with it's simple deployment, and being meant for simpler tasks that can fit within a function, like in our case where the first API is preprocessing an image and forwarding it to another API call. We ended up going with a cloud run.

With cloud run, you only have to build a docker container that hosts a simple flask front end application. With this containerized solution, it makes it easy to adapt the application. This came in handy when starting to test the final implementation, making it easy to visually see the results by presenting it in our application, add and remove small features quickly, since you can easily test that application locally first before deploying it up in the cloud.

Another important thing to take into consideration when picking the right service for your project is the pricing of the different alternatives. Deploying the system in a real world scenario would exceed the quotas GCP sets up for their "free tier". In the end the different services all offer different possibilities and pricing to follow. E.g. Compute engines offer a lot of ability to customize different parameters, which lets you have more control over the pricing as well. During the development and testing of our system we never did exceed said quotas, and going into any further discussion on the pricing of different options becomes a bit superfluous, since it will depend a lot on how the final deployment will look.

5.5   Joint calibration of cameras

Initially, the idea was to map the given area by using several cameras. By achieving a feasible mapping of the given space an arbitrary placement of the thermal camera is possible. The placement of the thermal camera thus becomes flexible and could be placed with an overview of the room.

The orientation of the different camera axes will differ, hence one must perform a conversion so that joint representation is obtained. Different angles and resolution must be taken into account but also 3D and 2D conversion. The conversion is difficult especially if it is to be scalable, however the goal of the project is a proof of concept. A complete generic solution is desirable though unreasonable. The idea was ambitious, and because several cameras were needed, costly.

By limiting the initial idea, the mapping of the given area did not become a major part of the paper. A joint calibration of the RGB and Thermal camera was performed, assuming fixed position. The relatively simple process is described in section 3.4. The consequence of using a limited number of cameras is that further development is required, where larger areas are involved and total coverage of the given area is considered. Despite the aforementioned negative consequences of a small number of cameras, a number of assumptions and statements can be made. Limiting the scope of the paper also makes it more manageable.

5.6   Cameras

Initially an all in one solution was considered regarding the camera alternatives, i.e. a dual-camera. The benefits with a dual-camera which includes both RGB and Thermal data in the common x, y - plane / x, y, z - room. The approach is based on detecting and validating using the RGB images and thermal images

respectively, hence why joint coordinates would be beneficial. Options exist that offer an all in one solution but the cost is significantly higher and the ability to customize the solution is limited. Since the choice of hardware, the economic factor, its area of use, and possibilities that follow are the basis of the paper, a decision was made to investigate separate parts.

The approach is based on detection at a position (x, y / x, y, z) using the RGB images. Multiple images and data from different angles would enable the mapping of the plane / room to represent the given space monitored. However the idea based on multiple cameras and thus angles was suppressed mainly by the impact on the economic factor but also the limited time for implementation. To represent three axes, an attribute which measures the depth is required.

Even though multiple cameras were no longer a viable option, a depth sensor adds further development opportunities. The Realsense technology meets the requirements by using an Infrared (IR) depth sensor. The resolution of the RGB camera is also of great importance as object detection is the technology to be used to detect people. In addition, desirable attributes such as extended scope and a variety of sockets are beneficial. Extended scope results in greater surface coverage, and less demands on other hardware is fulfilled by a variety of sockets. The Realsense technology provides RGB cameras with all the above attributes, depth sensor, high resolution, wide angle lens and a variety of sockets.

Concerning the resolution of the thermal camera it is not as important. By observing the heat at a given position indicated by the detection a true positive or false positive is derived. Hence, the essential attribute is the possibility to distinguish heat in different areas, thus the representation of a position in the detected area may contain a larger area of pixels. The thermal cameras with a low cost factor and thus favorable alternatives resulted in one large heat signal whereas a high cost factor led to unreasonable alternatives. A middle ground alternative was discovered, the FLIR Lepton camera. Attributes as a wide angle lens and a variety of sockets is, as in the RGB case, important. For the thermal camera a wide angle lens is even more important since they are to be used only for validation and an extensive surface coverage provides a low cost factor. All requirements are met by the camera concerned, the FLIR camera.

5.7   Environment

The impact of the environment in which the test was performed should not be diminished. Testing the solution in a conference room is beneficial. A conference room is relatively simply decorated, which provides fewer distractions and obstacles and is often intended to gather people in the center of the room. Based on a conference room, the requirement for speed decreases due to the fact that the area of use rarely generates much traffic. The temperature in the room is relatively constant on a daily basis, however, heat sources, such as a coffee cup, can affect the outcome of the heat point summation.

5.8   GDPR

As mentioned in section 3.7 GDPR is taken into account. The images used to detect people are not stored and hence no further action is required. Potential saved images can be used to monitor and log events in the environment, but are not attributes that affect the intended system and are therefore for the sake of simplicity neutralized.

5.9   Alternative uses

As stated in section 1, there was a desire to find alternative uses. Solving the problem alone provides the opportunity for further development, but also the use of cameras. By solving the problem, detecting the number of people in a given area, for instance sub-areas originated, as optimizing surface usage. For instance, if the system were to be used in a train, the number of carriages could be regulated by traffic (number of people) in a given time interval. The economic factor will then be positively affected, less fuel is needed and also less environmental impact, which in the end can benefit the common person. Another usage is measuring the average distance between people in the given area. Further studies can be made by exploring cultural differences or also as mentioned to follow given recommendations and / or restrictions. The alternative uses are extensive and due to the fact that similar more advanced systems are in use, the field is sought after and useful.

Chapter 6

Future work

The system introduced in this thesis is considered to be a proof of concept, showing that a thermal camera can contribute to a more robust implementation of a system with the task of counting the amount of people in a given area observable by said thermal camera in conjunction with a normal RGB camera. During the development and testing of said system, some ideas thought of were not implemented. New ideas also emerged during the development such as improvements and alternative usage which was not implemented.

To begin with, initially the idea was based on mapping the area meant to be observed by having multiple cameras at different angles. After examining the option, it appeared unrealistic both in terms of time constraints and financially unsustainable. Therefore, the choice was made to abandon the initial idea and proceed with a set of cameras. Adding a camera with a different angle to the implementation would add more coverage, but mainly prevent misses where people are obstructed by objects or captured in unfavorable positions.

Simply changing out the cameras might also prove beneficial, using cameras with improved wide-angle lenses would give the system more coverage. There are also camera alternatives that include both RGB and thermal video streams, implementing the system with one of these would take away the currently necessary step of making sure to align the cameras so you can observe the corresponding pixels.

When it comes to the implementation of the Google Cloud Platform, while using cloud run makes it easier to test, if you do not need the website for an actual deployment, the first API that receives image data seems to make a good case for a migration over to cloud functions.

Alternative uses of the system is not utilized, described in section 5.9, which provides an opportunity for further development.

_____ Chapter 7

Conclusion

_____

The paper is a proof of concept and the main goal is to examine if it is possible to detect the amount of people in a given area using modern technology with a limited financial budget. Using affordable and relevant components, cloud computing and publicly available object detection models, it is possible to develop a system that detects the amount of people in a given area. Depending on the model used, the accuracy and speed will be affected. Models can be optimized with regard to the metrics that are considered important for the specific area of use. Thermal data is a useful complement. Depending on the intended area of use, it can be crucial for a satisfactory result. In addition to the main goal, the system can be used for alternative purposes and thereby create added value to the end product.

References

[1]   Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. DOI: `10.1109/TNNLS.2018.2876865`.

[2]   T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755.

[3]   J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[4]   I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725–1730.

[5]   *Data protection in the EU*, Accessed on: 2021-06-14. [Online]. Available: `https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en`.

[6]   B. N. K. Sai and T. Sasikala, "Object detection and count of objects in image using tensor flow object detection api," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 542–546. DOI: `10.1109/ICSSIT46314.2019.8987942`.

[7]   C. W. Zhao, J. Jegatheesan, and S. C. Loon, "Exploring iot application using raspberry pi," *International Journal of Computer Networks and Applications*, vol. 2, no. 1, pp. 27–34, 2015.

[8] R. I. Pereira, I. M. Dupont, P. C. Carvalho, and S. C. Jucá, "Iot embedded linux system based on raspberry pi applied to real-time cloud monitoring of a decentralized photovoltaic plant," *Measurement*, vol. 114, pp. 286–297, 2018, ISSN: 0263-2241. DOI: `https://doi.org/10.1016/j.measurement.2017.09.033`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S026322411730605X`.

[9] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: `1511.08458 [cs.NE]`.

[10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. DOI: `10.1109/ICEngTechnol.2017.8308186`.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malikn, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision ICCV*, 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: `1506.01497 [cs.CV]`.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *Ssd: Single shot multibox detector*, Cham, 2016.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016.

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. arXiv: `1804.02767`. [Online]. Available: `http://arxiv.org/abs/1804.02767`.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *Mobilenetv2: Inverted residuals and linear bottlenecks*, 2019. arXiv: `1801.04381 [cs.CV]`.

[18] R. K. Srivastava, K. Greff, and J. Schmidhuber, *Highway networks*, 2015. arXiv: `1505.00387 [cs.LG]`.

[19] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: `1512.03385 [cs.CV]`.

[20]   J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, *Selective search for object recognition*, 2013-09-01. Accessed on: 2021-03-30. DOI: `10.1007/s11263-013-0620-5`. [Online]. Available: `https://link.springer.com/content/pdf/10.1007/s11263-013-0620-5.pdf`.

[21]   J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22]   W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, 2006-12-01. Accessed on: 2021-04-01. DOI: `10.1038/nbt1206-1565`. [Online]. Available: `https://www.ifi.uzh.ch/dam/jcr:00000000-7f84-9c3b-ffff-ffffc550ec57/what_is_a_support_vector_machine.pdf`.

[23]   T. Yichuan, "Deep learning using support vector machines," *CoRR*, vol. 1306.0239, 2013. arXiv: `1306.0239`. [Online]. Available: `http://arxiv.org/abs/1306.0239`.

[24]   Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25]   H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face recognition using convolutional neural network and simple logistic classifier," in *Soft Computing in Industrial Applications*, V. Snášel, P. Krömer, M. Köppen, and G. Schaefer, Eds., Cham: Springer International Publishing, 2014, pp. 197–207, ISBN: 978-3-319-00930-8.

[26]   *Imagenet*, n.d. Accessed on: 2021-05-16. [Online]. Available: `https://www.image-net.org/index.php`.

[27]   *Imagenet, ILSVRC*, n.d. Accessed on: 2021-05-16. [Online]. Available: `https://www.image-net.org/download.php`.

[28]   A. Tharwat, "Classification assessment methods," in *Applied Computing and Informatics*, 2020.

[29]   *Google Cloud Platform, Cloud FUnctions*, n.d. Accessed on: 2021-05-30. [Online]. Available: `https://cloud.google.com/functions`.

[30]   *Google Cloud Platform, Cloud Run*, n.d. Accessed on: 2021-05-30. [Online]. Available: `https://cloud.google.com/run`.

[31]   *Google Cloud Platform, App Engine*, n.d. Accessed on: 2021-05-30. [Online]. Available: `https://cloud.google.com/appengine`.

[32] *Google Cloud Platform, Compute Engine*, n.d. Accessed on: 2021-05-30. [Online]. Available: `https://cloud.google.com/compute`.

[33] *Raspberry Pi OS*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://www.raspberrypi.org/documentation/raspbian/`.

[34] *Software*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://www.raspberrypi.org/documentation/linux/software/`.

[35] *Installing operating system images*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://www.raspberrypi.org/documentation/installation/installing-images/`.

[36] *raspi-config*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://www.raspberrypi.org/documentation/configuration/raspi-config.md`.

[37] *Raspberry Pi 4 Model B*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf`.

[38] *Exploring IOT Application Using Raspberry Pi*, n.d. Accessed on: 2021-03-20. [Online]. Available: `https://ijcna.org/Manuscripts/Volume-2/Issue-1/Vol-2-issue-1-M-04.pdf`.

[39] *Guidelines 3/2019 on processing of personal data through video*, 2019-07-10. Accessed on: 2021-03-08. [Online]. Available: `https://edpb.europa.eu/sites/edpb/files/consultation/edpb_guidelines_201903_videosurveillance.pdf`.

[40] *What if my company/organisation fails to comply with the data protection rules?* n.d. Accessed on: 2021-03-08. [Online]. Available: `https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/enforcement-and-sanctions/sanctions/what-if-my-company-organisation-fails-comply-data-protection-rules_en`.

[41] *International transfers of personal data*, n.d. Accessed on: 2021-03-08. [Online]. Available: `https://ec.europa.eu/info/law/law-topic/data-protection/international-dimension-data-protection/rules-international-data-transfers_en`.

Extra material

Test ID	Test Case	Expected Outcome	Result
TEST ID	TEST CASE	EXPECTED OUTCOME	RESULT
1	Enable RGB camera stream	Stream viewable from external device	✓
2	Enable thermal camera stream	Stream viewable from external device	✓
3	Save the most recent RGB image	Most recent RGB image saved	✓
4	Retrieve the most recent RGB data	Most recent RGB data retrieved	✓
5	Retrieve the most recent data from thermal camera	Most recent thermal data retrieved	✓
6	Compatibility test of TEST ID 1-5 on SBC	TEST ID 1-5 function on SBC	✓
7	Integrate functional API in the cloud	Functional API in the cloud	✓
8	Integrate object detection model in the cloud	Model integrated in the cloud	✓
9	Integrate storage and custom handling in the cloud	Storage and handling integrated	✓
10	Send the most recent RGB image to the cloud	Most recent RGB image received	✓
11	Apply object detection model on the most recent RGB image	Model output saved and result recorded	✓
12	Send the most recent data from thermal camera to the cloud	Most recent thermal data received	✓
13	Validate output from object detection model by using thermal data	Validated output saved and result recorded	✓
14	Log all results and present current state to the end user	Log results and current state presented to the end user	✓

Table A.1: Overview test data

LUND
UNIVERSITY