HTBLVA, Wien V, Spengergasse

Höhere Lehranstalt für Informatik
Ausbildungsschwerpunkt Informatik - Tag

DIPLOMARBEIT

Gesamtprojekt

Mask Detection and Distance Measurement Software to protect against COVID19

in Kooperation mit

Austrian Institute of Technology GmbH

Implementing and creating algorithms for property detection of visual data.
Lukas Gäbler 5AHIF Betreuer: Ing. Mag. Harald Zumpf

Backend and system architecture for storage procedures of visual data.
Moritz Patek 5AHIF Betreuer: Ing. Mag. Harald Zumpf

Parameter and calibration procedures and HID for accurate distance measurement.
Tobias Pressler 5AHIF Betreuer: Ing. Mag. Harald Zumpf

Schuljahr 2021/22

Abgabevermerk:

Datum: übernommen von:

Berufsbildende Schulen

Erklärung

Die unterfertigten Kandidaten/Kandidatinnen haben gemäß § 34 Abs. 3 Z 1 und § 37 Abs. 2 Z 2 des Schulunterrichtsgesetzes in Verbindung mit den Bestimmungen der „Prüfungsordnung BMHS, Bildungsanstalten", BGBl. II Nr. 177/2012 i.d.g.F. die Ausarbeitung einer Diplomarbeit/Abschlussarbeit mit folgender Aufgabenstellung gewählt:

Mask Detection and Distance Measurement Software to protect against COVID19 (Gesamtprojekt)

Individuelle Aufgabenstellungen im Rahmen des Gesamtprojektes:

- Lukas Gäbler (5AHIF): **Implementing and creating algorithms for property detection of visual data**
- Moritz Patek (5AHIF): **Backend and system architecture for storage procedures of visual data**
- Tobias Pressler (5AHIF): **Parameter and calibration procedures and HID for accurate distance measurement**

Die Kandidaten/Kandidatinnen nehmen zur Kenntnis, dass die Diplomarbeit/Abschlussarbeit in eigenständiger Weise und außerhalb des Unterrichtes zu bearbeiten und anzufertigen ist, wobei Ergebnisse des Unterrichtes mit einbezogen werden können, die jedenfalls als solche entsprechend kenntlich zu machen sind.

Die Abgabe der vollständigen Diplomarbeit/Abschlussarbeit hat in digitaler und in zweifach ausgedruckter Form bis spätestens **05.04.2022** beim zuständigen Betreuer/der zuständigen Betreuerin zu erfolgen.

Die Kandidaten/Kandidatinnen nehmen weiters zur Kenntnis, dass ein Abbruch der Diplomarbeit/Abschlussarbeit nicht möglich ist.

Kandidaten/Kandidatinnen:	**Datum und Unterschrift bzw. Handysignatur:**
Lukas Gäbler (5AHIF)	16.09.21 Lukas Gäbler
Moritz Patek (5AHIF)	16.09.21
Tobias Pressler (5AHIF)	16.09.21

Abstract

At the end of the year 2019, the COVID-19 outbreak lead to a number of restrictions, implemented by governments all around the world in order to protect their citizens from a COVID infection and to prevent the collapse of their national health systems. In order to evaluate the compliance of these restrictions, we created a system for *mask detection and distance measurement software to protect against COVID-19 (MDDM)*. Using computer vision and artificial intelligence, we were able to ensure accurate measurements between multiple subjects. We were also able to accurately detect, the state of face-mask wearing. Subsequently, the system classifies each subject's compliance in regard to social distancing and mask obligation. To achieve this, the team conducted studies on various ways of distance measuring between subjects on images. First, it was important to estimate the conversion factor from pixel to meters depending on the tilt angle and the height of the camera. Next, various ways for object detection were studied to create a knowledge base for methods that are used in MDDM. Finally, artificial intelligence methods and algorithms for distance extrapolation on images are analyzed for the use in our project's distance measurement system.

Abstract (German)

Ende des Jahres 2019 führte der COVID-19-Ausbruch zu einer Reihe von Beschränkungen, die von Regierungen auf der ganzen Welt eingeführt wurden, um die Bürger vor einer Infektion mit dem Virus zu schützen und den Zusammenbruch des nationalen Gesundheitssystems zu verhindern. Um die Einhaltung dieser Beschränkungen zu bewerten, haben wir eine Software zur Maskenerkennung und Abstandsmessung zum Schutz vor COVID-19 (MDDM) entwickelt. Durch den Einsatz von Computer Vision und künstlicher Intelligenz waren wir in der Lage, genaue Messungen zwischen mehreren Personen durchzuführen. Wir waren auch in der Lage, genau zu erkennen, ob eine Person eine Gesichtsmaske trägt. Anschließend klassifiziert das System die Compliance der einzelnen Probanden in Bezug auf soziale Distanz und Maskenpflicht. Um dies zu ermöglichen, führte das Team unterschiedliche Studien durch. Zunächst war es wichtig, einen Umrechnungsfaktor von Pixel in Meter zu erhalten, der vom Neigungswinkel und der Höhe der Kamera abhängt. Als nächstes wurden verschiedene Methoden zur Objekterkennung untersucht, um eine Wissensbasis für die in MDDM verwendeten Methoden zu schaffen. In der darauf folgenden Arbeit werden Methoden der künstlichen Intelligenz und Algorithmen zur Entfernungsmessung auf Bildern analysiert.

Acknowledgements

In the name of the team, I would like to formally thank everyone who made this project possible.

For the complete duration of this project, we got outstanding support from our PRE teacher and project supervisor, Harald Zumpf.

Together with professor Zumpf we came up with the idea of Babyelefant which turned out to be the most interesting project we worked on during our time at HTL Spengergasse.

In addition to Harald Zumpf, we want to thank our teachers at HTL Spengergasse, for their year long support and for equipping us with the knowledge and skills to tackle this project.

- Lukas Gäbler

Implementing and creating algorithms for property detection of visual data

Lukas Gäbler
HTL Spengergasse
gae18805@spengergasse.at

2021/2022

Contents

To my family, for helping me pursue my dreams
and
To Tobias and Moritz, for enduring me during this work

1 Abstract

The advancements in machine learning, artificial intelligence, and especially computer vision over the last few years are enormous. Starting with cars driving themselves just by analyzing images and videos and acting based on that information (Tesla, 2022), all the way to helping scientists and archaeologists discover more details about ancient artifacts. The variety of possible use cases of computer vision is endless.
In order to accurately describe the ways and methods to detect objects and the properties of these objects in visual data, multiple challenges are to be solved.

2 Introduction

The applications of artificial intelligence (AI) and machine learning in the field of computer vision have proven to be advanced very fast over the last few years. On several occasions, humans got out-performed by machines already. AlphaGo, the Go player based on machine learning created by DeepMind, was the first computer program to defeat a professional human Go player, the first to defeat a Go world champion (DeepMind, 2017). But not only when it comes to games, but machines are also able to outperform humans. With technologies like YOLO, which stands for You Only Look Once, (Redmon, Divvala, Girshick, & Farhadi, 2015) the field of computer vision has advanced a lot.

Machine learning algorithms build a model that uses some sort of training data, which then can be used to classify new data, predict possible future data, make an agent act in an environment, create data that has never existed like that before, (Karras, 2022) and much more. Given the correct training data, machine learning can be used to detect objects on visual data.
The application of artificial intelligence and machine learning to the field of computer vision has proven inevitable for multiple reasons. On one hand, advancements in computing and data storage capabilities have brought machine learning back into the mainstream. On the other hand, the never-ending thrive by companies to build the best *artificial intelligence vision system* leads to the biggest technology companies in the world, competing for that title. Therefore companies like Google, Facebook, and Amazon release their own algorithms and libraries and developers have a wide range of possible ways to implement an application. Although all these libraries solve the same problems, there are more factors that decide which one to use. For some, the community is bigger, and therefore it is easier to find help when something is not working and for others, the performance on large datasets is better.

In the following sections, this paper provides a brief introduction to machine learning and computer vision.
Finally, a brief overview of computer vision applications and current development in artificial intelligence is given.

3 What is Machine Learning

It is not easy to define the term "machine learning". There are hundreds of different definitions that try to accurately describe it. Alan Turing often mentioned as the godfather of artificial intelligence described an abstract computing machine consisting of a limitless memory and a scanner that moves back and forth through the memory (Amini, 2021).
Alan Turing died in 1954 and since then a lot has changed in the world of artificial intelligence. Although machine learning and artificial intelligence have evolved a lot since then, his idea of what is now known as the Turing machine is still a present topic in the world of computer science.
Besides Alan Turing's way to view this topic, there are many other definitions. For instance, Arthur L. Samuel defined machine learning as follows:

"Field of study that gives computers the ability to learn without being explicitly programmed.", Arthur L. Samuel (Esposito, Bheemaiah, & Tse, 2017)

Over the last 10 years, machine learning gained extreme popularity. Convolutional neural networks (CNNs) in combination with supervised learning can be used to detect what is an image. General adversarial networks consisting of multiple CNNs can be used to generate images of people who never lived. (Karras, 2022). Reinforcement learning can be used to teach robots how to walk or to do certain tasks. There are many different algorithms that can be found in nearly any modern application.

3.1 Basic Concept of Machine Learning

Machine Learning algorithms start with linear regression. Although usually not referred to as machine learning or learning algorithms, linear regression is the most basic form of such.
Linear regression is used to find the function which best connects a set of given points. This pattern can also be found when it comes to machine learning. In general, it is trying to understand or find a pattern of given data in order to make accurate predictions. Certainly, linear regression is not considered machine learning, however, the similarities between these close topics help to explain the basic concepts of machine learning.

3.2 Different Types of Learning Algorithms

The most common differentiation between learning algorithms in machine learning is the one of supervised learning and unsupervised learning. Although, these two types cover a large area of applications, some other machine learning algorithms, such as Reinforcement Learning or Recommender Systems, gained a lot of popularity during the last few years and therefore more applications are using those learning algorithms. They are neither supervised or unsupervised or sometimes both.

3.2.1 Supervised Learning

This learning algorithm is by far the most known one in the world of machine learning. Everything starting from simple regression to deep neural networks can be classified as supervised. Supervised learning describes the learning process of an algorithm that is able to learn when a target y is given for each input x in a training set. The most common applications for supervised learning are classification and regression. In classification, a hypothesis is predicted in order to separate the given data, creating classes to distinguish from, while in regression, a hypothesis is predicted in order to mock the given data most accurately.(Ng, 2022)

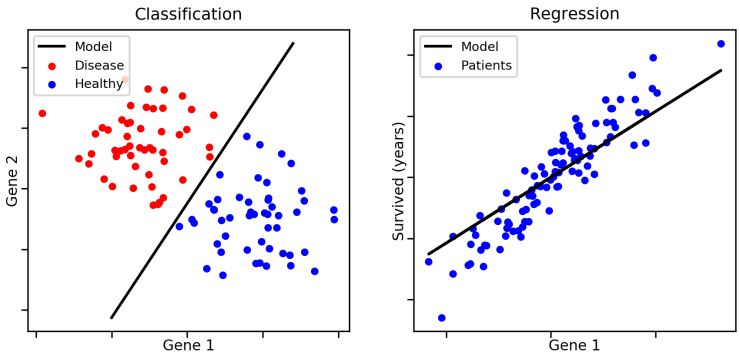


Figure 1: Supervised Learning - Classification and Regression

Figure 1 uses an example of patients, who are classified, based on genes if they have a disease or not. The red dots represent the class of people with a disease and the blue dots represent the class of people who are healthy.
The regression, on the right-hand side, shows that each dot is associated with a quantitative target value. In this case, the target value is the number of years this patient survived.

A dataset in supervised learning consists of an input vector and an output vector. The parameters of the algorithm then are learned and evaluated on this set, figuring out a correlation between an element in the input vector and the corresponding element in the output vector.

$$\begin{aligned} inputs &= \begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix} \\ outputs &= \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{aligned} \tag{1}$$

The width of the input matrix represents the number of features and the height represents the number of training set examples. The height of the output matrix must be the same as the number of features in the input matrix.

3.2.2 Unsupervised Learning

In contradistinction to supervised learning (3.2.1), unsupervised learning deals with unlabelled data. In difference to labeled data, unsupervised learning offers a great opportunity to make use of unlabelled data in a way that would otherwise not be considered. As the methods of unsupervised learning are mostly about finding groups of data that appear to belong together, they are sometimes referred to as 'clustering'.

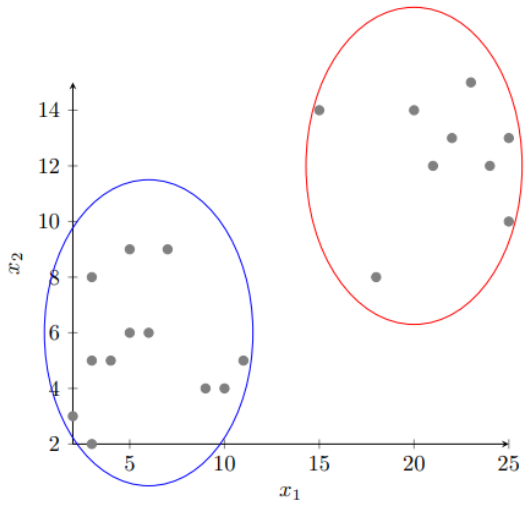


Figure 2: Unsupervised Learning

k-Means
Just because of the similar name, the k-Means algorithm only shares a few similarities with the k-Nearest Neighbor algorithm. The k-Means algorithm partitions a data set into k clusters according to the data point's distance to the nearest mean (Lloyd, 1982).

Although the standard implementation of the k-Mean algorithm, as introduced by (Lloyd, 1982) is an NP-hard [1] problem and is not guaranteed to converge to a global minimum, it is often used as a basis for more optimal implementations.

Hierachical Clustering
Another often-used group of clustering algorithms use hierarchical clustering methods. This method builds a cluster-tree, where nodes are groups of data points. There are two major approaches to this type of clustering. The first one is called agglomerative clustering. When using this bottom-up approach, every data point starts in it's own cluster. The second one, the opposite to the first approach, is called divisive clustering (L. & O., 2005).

3.2.3 Reinforcement Learning

Reinforcement learning is known for being able to defeat grandmasters in chess or for beating high scores in Atari games. The key concepts behind reinforcement learning are relatively simple.

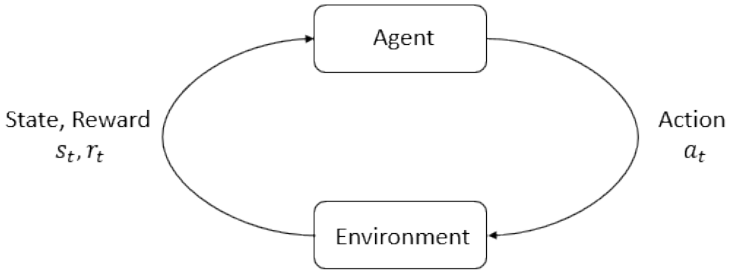


Figure 3: Reinforcement Learning Environment

The main players in reinforcement learning are the **agent** and the **environment**. The environment is where the agent lives in and interacts with. It is the world in which the agent acts. In each phase. the agent sees a (possibly only partial) observation of the state of the environment and then chooses what to do and acts. The world changes when the agent acts with or in it, but it can also change by itself. (OpenAI, 2020).

[1] "A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem" (Naseera, 2020)

The agent perceives a **reward** from the environment, a number that tells it how good or bad the current world state is. The goal of the agent is to maximize it's reward. The methods or reinforcement are techniques that the agent can learn behaviours to achieve it's goal. (OpenAI, 2020)

Policies
The agent decides what action to take, based on it's policy. It can be deterministic, in which case it is denoted by μ:

$$a_t = \mu(s_t)$$

or it may be stochastic, in which case it is denoted by π:

$$a_t \sim \pi(\cdot|s_t).$$

In deep reinforcement learning, we deal with parameterized policies. Parameterized policies are which outputs are computable functions that depend on multiple parameters (could be the weights and biases of a neural network) which can be adjusted to change the behavior, using some optimization algorithm.

We often denote the parameters of such a policy by θ or ϕ, and then write this as a subscript on the policy symbol to highlight the connection:

$$a_t = \mu_\theta(s_t)$$

$$a_t \sim \pi_\theta(\cdot|s_t)$$

(OpenAI, 2020)

Policy Optimization
Methods in this family represent a policy explicitly as $\pi_\theta(a|s)$. The parameters θ are optimized either by gradient ascent on the performance objective, or by maximizing local approximations of J(π_θ). For each update, only the data collected while acting according to the most recent version of the policy is used. The optimization, also includes learning and approximator Vϕ(s) for the on-policy value function Vπ(s), which is used in figuring out how to update the policy.

Q-Learning
Q-Learning methods learn an approximator Qθ(s,a) for the paradigmatic action-value function. The updates can use data collected at any point during training, regardless of how the agent explored the environment when the data was collected. The corresponding policy is obtained via the connection between Q^* and π^*: the actions taken by the Q-learning agent are given by

$$a(s) = \arg\max_a Q_\theta(s, a)$$

(OpenAI, 2018)

3.3 Deep Learning

3.3.1 Architecture of a Neural Network

An artificial neural network tries to mimic a human brain by using the same structure as found in nature. The human brain consists of around 86 billion neurons (How Many Neurons Are in the Brain?, 2018) connected by even more synapses which all together build a huge interconnected network - our brain. Artificial neural network try to copy the exact same structure - we just do not have the computing power to run a neural network with 86 billion neurons. The key difference between our brain and an artificial neural network is, that in a neural network, the neurons are grouped in different layers (Nagyfi, 2018). Figure 4 shows the structure of a simple neural network, consisting of an input layer, three hidden layers and one output layer.
The computation which is happening in a neuron can be described by the following equation:

$$\hat{y} = g\left(\mathbf{w}^T\mathbf{x} + b\right), \tag{2}$$

where g is an activation function, x are the inputs, w are the weights and b is a bias (Goodfellow, Bengio, & Courville, 2016, p. 127).
Inputs enter a neuron after being scaled by some weight w and are summed ($\sum$). An *activation function* (g) is applied to this result, which determines whether the neuron 'fires'. This process is called **forward propagation** .

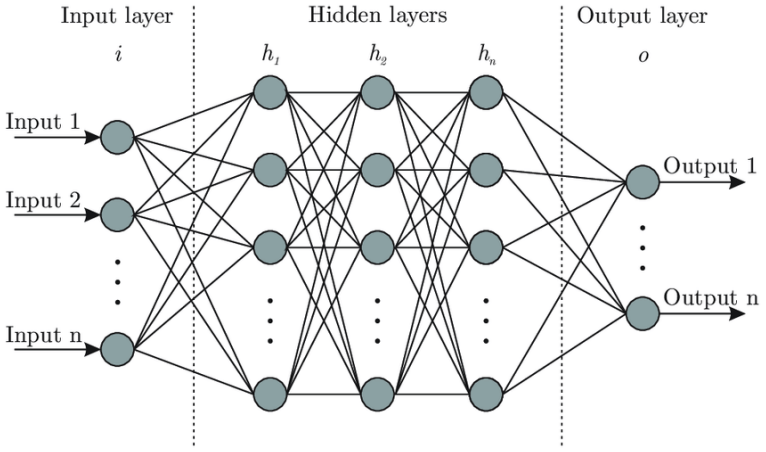


Figure 4: Simple neural network structure
(Bre & Gimenez, 2017)

3.3.2 Cost Function

The cost function determines how accurate a predicted value from a neural network was. Therefore, it is one of the most important aspects of a neural network. After determining the cost function, the goal is to find it's minimum. The following graph visualizes the goal:

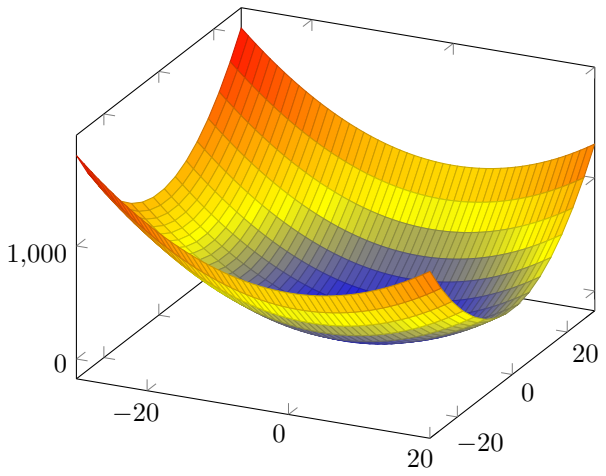


Figure 5: Cost function

During the optimization process, the goal is to find the global minimum, which is presented as the blue are in Figure 5 (Sanderson, 2018).

Optimization - Gradient Descent
The cost function (graphed in Figure 5 the important question to ask here is which direction do you need to step to decrease the output of the function. By calculating the gradient the direction of the steepest increase can be calculated.
The algorithm to compute this gradient direction, then take a small "step" downhill and do it again (Sanderson, 2018). If the value of the cost function decreases, it means the neural network performs better.
The parameters of a neural network are initialized randomly and are being adapted until the cost function is minimized. That way the neural network achieves better results.

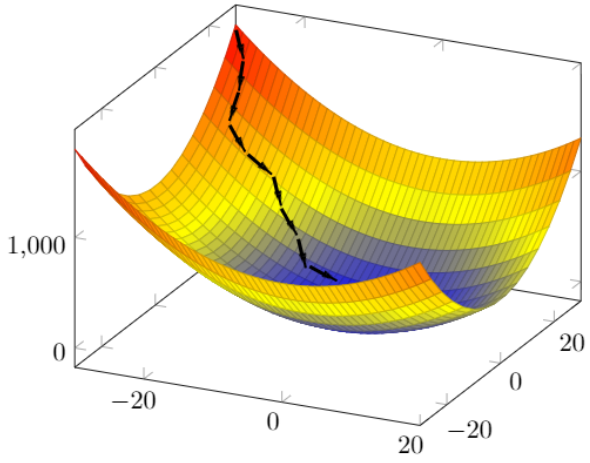


Figure 6: Cost function with gradient descent

3.3.3 Backpropagation

In order for the neural network to learn, the weights and biases on the connections between the nodes need to be adapted for the cost function to reach it's minimum. As neural networks consist of multiple layers, this has to be done for each layer. Backpropagation is essentially the terminology for minimizing the cost function. To achieve that, the calculated error has to be propagated back in the neural network, starting with the last layer (the output layer) and ending with the first hidden layer.

$$\delta^L = a^{(L)} - y^{(i)} \tag{3}$$

where:
δ^L is the partial derivative of the last layer
$a^{(L)}$ is the activation of the last layer
$y^{(i)}$ is the actual result of training example i

Equation 3 shows the back-propagation of the output layer. In order to backpropagate a hidden layer l, layer l+1 must also be considered:

$$\delta^l = ((\theta^{(l)})^T \delta^{l+1}) \times a^{(l)} \times (a - a^{(l)}) \tag{4}$$

The calculated partial derivative for layer l will be multiplied with the activation of layer l.
When all gradients of all layers are computed, the weights of the neural network can be updated respectively. (Ng, 2021)

4 Computer Vision

4.1 What is Computer Vision

Computer Vision is described as the field of study surrounding how computers see and understand digital images and videos. Computer vision spans all tasks performed by biological vision systems, including "seeing" or sensing a visual stimulus, understanding what is being seen, and extracting complex information into a form that can be used in other processes. This interdisciplinary field simulates and automates these elements of human vision systems using sensors, computers, and machine learning algorithms. (DeepAI, 2021)

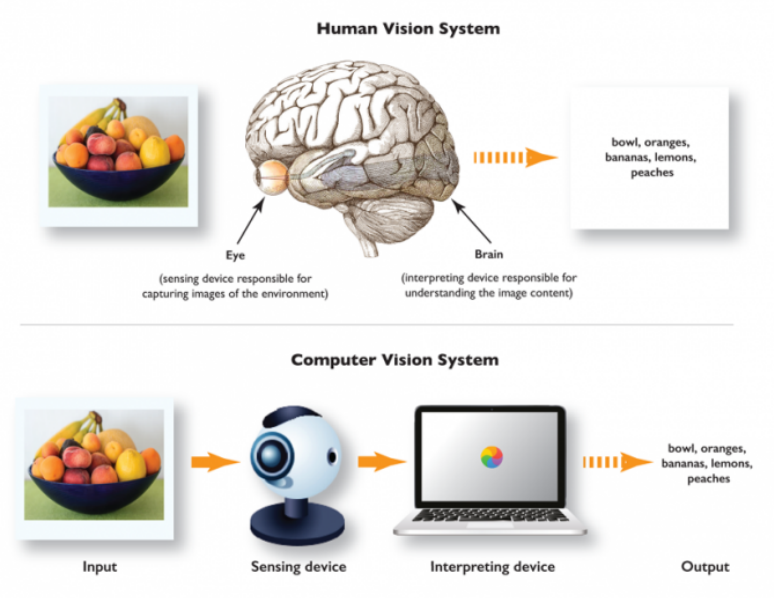


Figure 7: Human Vision and computer vision systems process visual data in a similar way

(Babich, 2020)

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving, and whether there is something wrong in an image.

Computer vision has to do all these tasks in much less time. Instead of retinas, optic nerves, and a visual cortex it has cameras, data, and algorithms. Human capabilities are quickly surpassed when a trained system is able to inspect products or watch a production asset and quickly analyzes thousands of products a minute.(IBM, 2019)

As described in section 4.4 computer vision has a wide range of applications. From automotive to manufacturing to energy and utilities. It is expected for the computer vision and hardware market to reach $ 48.6 billion (Marr, 2019).

4.2 Basic concepts of Computer Vision

The most basic need of Computer Vision is data. Computer Vision needs data to learn. It runs analyses of data over and over until it discerns distinctions and ultimately recognizes images. To train a computer to recognize people on an image, it needs to be fed vast quantities of images with people to learn the differences from other objects.

By using Deep Learning (Section 3.3) and convolutional neural networks in combination with supervised learning (Section 3.2.1) a computer is able to learn to recognize objects on visual data. The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions start to come true. It is then recognizing or seeing images in a way similar to humans (IBM, 2019).

A Convolutional Neural Network first detects hard edges and simple shapes. A CNN is used to understand single images. A recurrent neural network (RNN) is used in more or less the same way for video applications. It helps computers understand how pictures in a series of frames are connected or related to each other.

4.3 Different types of Computer Vision

Besides classifying images there are more types and use cases for computer vision. As described in section 4.4 there are a lot more applications. From self-driving cars (section 4.4.1) and computer vision in medicine (section 4.4.2) to law enforcement with face detection (section 4.4.3) and historical analysis (section 4.4.4). The field of computer vision is already big and it is growing every day. There are different ways to tackle different problems. The way computer vision is mostly used is either object detection or image segmentation.

4.3.1 Object detection

Object detection is the name for the process when a neural network is able to detect objects on images or videos.

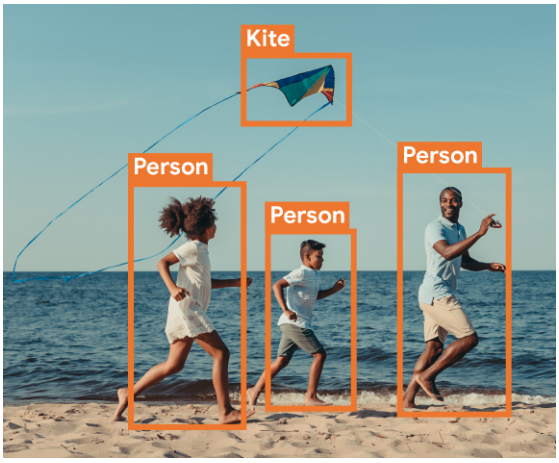


Figure 8: Example of object detection in an image
(Rathod & Huang, 2020)

Figure 9: Example of property detection on visual data
(Teng, 2021)

It is also possible to teach a computer how to recognize the properties of visual data. The traffic light detection in Figure 9 is only an example. With computer vision, it is also possible to teach a computer how to get the measurements of an object or to recognize street signs.

4.3.2 Image segmentation

In an image classification task the network assigns a label (or class) to each input image. However, if you want to know the shape of that object, just detecting it isn't enough. In this case you will want to assign a class to each pixel of the image. This process is known as segmentation. In computer vision, a segmentation model returns much more detailed information about the image. (Google, 2021) The training process is basically still the same. The training data just looks different.

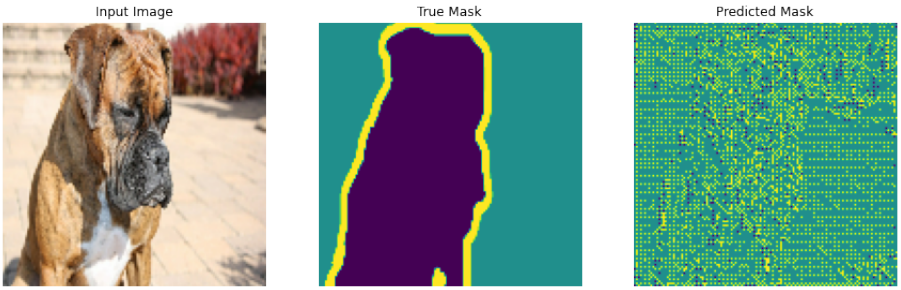


Figure 10: Image Segmentation Data
(Google, 2021)

In Figure 10 the predicted mask is the output of an untrained neural network. After the training is done, the prediction will look more like the true mask, although it won't be exactly the same.

4.4 Applications of Computer Vision

The most known application of computer vision are self-driving cars. The car company Tesla Motors places 8 cameras around the car to provide a 360° view at up to 250 meters of range. (Tesla, 2022)
Today computer vision is already used to support doctors when it comes to analyzing medical images like x-rays or magnetic resonance imaging images.
Law enforcement institutions around the world are already using face detection technology based on machine learning and artificial intelligence. There are a lot of controversies around the usage of this technology as it can be biased based on the training data. For example, facial recognition systems are a lot less accurate when it comes to correctly identifying female people with darker skin. (Najibi, 2020)

4.4.1 Self driving cars

The most popular application of computer vision are self-driving cars. Tesla Motors achieved a lot of progress in that field over the last few years. By placing eight cameras around the car, it is possible to detect the lanes of a street, street signs like speed limits or stop signs, and road lights. All that information combined makes it possible for the car to drive itself. The included system analyzes all the collected information and acts based on it. The autopilot is able to steer automatically, change lanes, park and a lot more. The car even "remembers" how it was parked into a parking lot and can drive out by itself. Pedestrians are protected by several security features. The automatic emergency braking makes it possible for the car to detect objects the car may impact and breaks accordingly. (Tesla, 2022)

Figure 11: Tesla Computer Vision

4.4.2 Computer Vision in medicine

When it comes to medicine, computer vision can be used for diagnostics or for therapy. In Diagnostics the possibilities are ginormous. New ways to combine computed tomography and magnetic resonanc imaging images with diffusion tensor imaging tractography and the use of image segmentation protocols make it possible to 3D model the skull base, tumor, and five eloquent fiber tracts. (Gargiulo, Íris Árnadóttir, Gíslason, Edmunds, & Ólafsson, 2017)

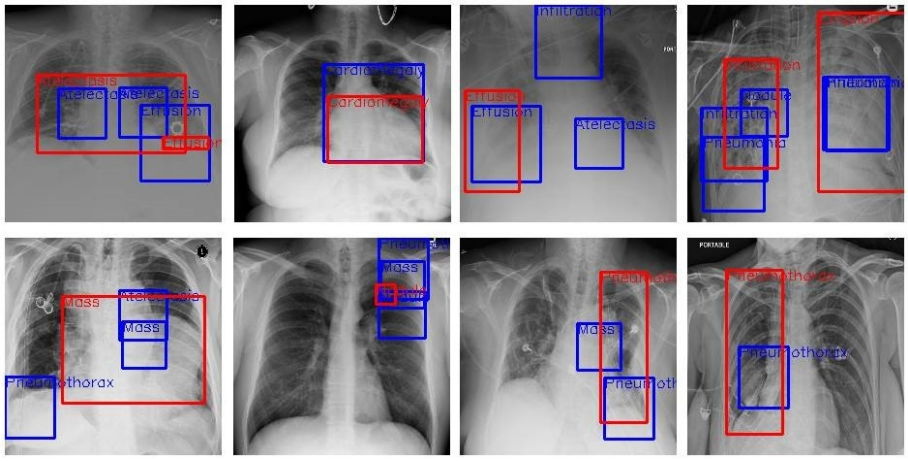


Figure 12: Chest X-Ray with outputs
(Tang, 2019)

As shown in figure 12 computer vision is already used to help doctors find diseases.

Computer vision can be integrated into endoscopic procedures for scope guidance, lesion detection and lesion diagnosis. Applications include esophageal cancer screening, detecting gastric cancer, detecting stomach infections and even finding hookworms. Scientists have built entire medial AI devices designed for monitoring. Beyond the analysis of disease states, computer vision can serve the future of human health and welfare through applications such as screening human embryos for implantation. (Esteva, Chou, & Yeung, 2020)

4.4.3 Face detection and recogonition

Face recognition got advertised as the most promising application in the field of computer vision. Face detection can be used for a substantial part of face recognition operations. The method of face detection in images is complicated as human faces are not always the same and factors like pose, position and orientation, skin color, facial hair, glasses and image resolution also play a big role when it comes to detecting faces.

With the help of many techniques, we can identify faces with higher accuracy.
The first step is to transform the picture from RGB to grayscale because it shows the contrants more clear.
As the second step, one will use a feature algorithm to find the location of the human faces in the picture. Most of the human faces share some universal properties like the eyes region is darker that it's neighbour pixels and the nose is brighter than the eyes. The applied feature algorithm will also be used for edge detection, line detection and for detecting eyes, nose, mouth, etc. in the image.

Figure 13: Face detection example
(Farley, Coulter, Sharkey, Christiani, & Kennedy, 2022)

As shown in figure 13 a classifier can also be combined with another one. In this case it's the face detection, combined with a classifier to predict the age and gender of the detected person.

4.4.4 Historical Analysis

Computer vision had barely contributed to the archaeological domain, however, there are a few use cases where it supports the science. There are applications for a content-based image retrieval system for historical glass and an automatic system for medieval coin classification. The content-based image retrieval system finds artifact drawings in a reference collection that are most similar to a photograph or drawing of a excavated historical glass. The similarity measurements are based on the outer shape contours of the artifacts. The system can speed up the process of classifying historical glass and make it more objective preliminary results on modern coin data. (Maaten, Boon, Lange, Paijmans, & Postma, 2007)

"Computer vision is also able to solve an increasing number of problems when it comes to art. In some cases, these computer methods are more accurate than even highly trained connoisseurs, art historians and artists. Rigorous computer ray-tracing software sheds light on claims that some artists employed optical tools. Computer methods will not replace tradition art historical methods of connoisseurship but enhance and extend them. As such, for these computer methods to be useful to the art community, they must continue to be refined through application to a variety of significant art historical problems." (Stork, 2009)

4.5 Recent developments in Computer Vision

Computer vision is among the most preogressive and rapidly growing fields. According to Grand View Research (SuperAnnotate, 2021), the global computer vision market size was valued at $11.32 billion in 2020 and is expected to expand at a compound annual growth rate of 7.3% from 2021 to 2028.

4.5.1 Computer Vision in the Cloud

As a lot of data is required for a neural network to work properly, a lot of computing power is needed to handle so much data. Using cloud services like Google Cloud, Amazing AWS or Microsoft Azure solves several challenges like network accessibility, bandwith, latency and computing power.
Using cloud services is especially popular when it comes to project where real-time data processing is needed. Such projects include self-driving cars, drones, etc.

4.5.2 Computer Vision for food quality evaluation

With increased expectations for food products of high quality and safety standards, the need for accurate, fast and objective quality determination of these characteristics in food products continues to grow. Computer vision provides one alternative for an automated, non-destructive and cost-effective technique to accomplish these requirements. This inspection approach based on image analysis and processing has found a variety of different applications in the food industry. Considerable research has highlighted its potential for the inspection and grading of fruits and vegetables. Computer vision has been successfully adopted for the quality analysis of meat and fish, pizza, cheese, and bread. Likewise grain quality and characteristics have been examined by this technique.

5 Conclusion

The development of computer vision over the last few years has shown, that machines can help a lot when it comes to safety and especially when it comes to supporting humans by analyzing visual data. It is evident that machine learning is capable of solving new kinds of problems that couldn't be solved before. Instead of manually defining a deterministic algorithm, a neural network approximates a function through a stochastic process of analysing a large dataset.

Without any doubt, it can be said that machine learning is a field that offers a wide range of new ways of solving problems. Although the idea of self-learning algorithms has existed for some time now, artificial intelligence experiences a big rise in popularity and general interest.

Modern frameworks such as Tensorflow and PyTorch make it possible to quickly develop and train capable models. Algorithms like YOLO (Redmon et al., 2015), implemented in these frameworks makes it easier and faster for developers to implement applications and achieve accurate results.

References

Amini, A. (2021). MIT Introduction to Deep Learning. Retrieved from https://www.youtube.com/watch?v=5tvmMX8r_OM

Babich, N. (2020). What is computer vision and how does it work. Adobe. Retrieved from https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/

Bre, F., & Gimenez, J. M. (2017). A neural network architecture. Retrieved from https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051

DeepAI. (2021). Computer vision. Retrieved from https://deepai.org/machine-learning-glossary-and-terms/computer-vision

DeepMind. (2017). Alphago. Retrieved from https://deepmind.com/research/case-studies/alphago-the-story-so-far

Esposito, M., Bheemaiah, K., & Tse, T. (2017, May). What is machine learning? The Conversation. Retrieved from https://theconversation.com/what-is-machine-learning-76759

Esteva, A., Chou, K., & Yeung, S. (2020). Deep learning-enabled medical computer vision. Retrieved from https://doi.org/10.1038/s41746-020-00376-2

Farley, P., Coulter, D., Sharkey, K., Christiani, T., & Kennedy, D. (2022). Face detection with computer vision. Retrieved from https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-detecting-faces

Gargiulo, P., Íris Árnadóttir, Gíslason, M., Edmunds, K., & Ólafsson, I. (2017). New directions in 3d medical modeling: 3d-printing anatomy and functions in neurosurgical planning. Retrieved from https://doi.org/10.1155/2017/1439643

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. (http://www.deeplearningbook.org)

Google. (2021). Image segmentation. Retrieved from https://www.tensorflow.org/tutorials/images/segmentation

How many neurons are in the brain? (2018). Retrieved from https://www.brainfacts.org/in-the-lab/meet-the-researcher/2018/how-many-neurons-are-in-the-brain-120418

IBM. (2019). What is computer vision. Retrieved from https://www.ibm.com/topics/computer-vision

Karras, T. (2022). This person does not exist. Retrieved from https://thispersondoesnotexist.com/

L., R., & O., M. (2005). Clustering methods. Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Retrieved from https://doi.org/10.1007/0-387-25465-X_15

Lloyd, S. (1982). Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2), 129-137. doi: 10.1109/TIT.1982.1056489

Maaten, L. v. d., Boon, P., Lange, G., Paijmans, H., & Postma, E. (2007). Computer vision and machine learning for archaeology.

Marr, B. (2019). 7 amazing examples of computer and machine vision in practice. Forbes. Retrieved from https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/#3dbb3f751018

Nagyfi, R. (2018). The differences between artificial and biological neural networks. Retrieved from https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

Najibi, A. (2020). Racial discrimination in face recognition technology. Retrieved from https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/

Naseera, S. (2020). P, np, np-hard np-complete problems. Retrieved from https://www.jntua.ac.in/gate-online-classes/registration/downloads/material/a159262902029.pdf

Ng, A. (2021). Backpropagation intuition. Retrieved from https://www.coursera.org/learn/machine-learning

Ng, A. (2022). Supervised learning. Retrieved from https://www.coursera.org/learn/machine-learning

OpenAI. (2018). Kinds of rl algorithms. Retrieved from https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#what-to-learn-in-model-free-rl

OpenAI. (2020). Key conecepts in rl. Retrieved from https://spinningup.openai.com/en/latest/spinningup/rl_intro.html#key-concepts-and-terminology

Rathod, V., & Huang, J. (2020). Tensorflow 2 meets the object detection api. Retrieved from https://blog.tensorflow.org/2020/07/tensorflow-2-meets-object-detection-api.html

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. CoRR, abs/1506.02640. Retrieved from http://arxiv.org/abs/1506.02640

Sanderson, G. (2018). Gredient descent, how neural networks learn. Retrieved from youtube.com/watch?v=IHZwWFHWa-w&t=73s

Stork, D. G. (2009). Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature. In International conference on computer analysis of images and patterns (pp. 9–24).

SuperAnnotate. (2021). Trends in computer vision. Retrieved from https://blog.superannotate.com/computer-vision-trends

Tang, T. (2019). Supervised learning for detections in medical images. Retrieved from https://github.com/thtang/CheXNet-with-localization

Teng, H. (2021). Traffic light detector. Retrieved from https://github.com/HevLfreis

Tesla. (2022). Autopilot. Retrieved from https://www.tesla.com/autopilot

List of Figures

Backend and system architecture for storage procedures of visual data

Moritz Patek
HTL Spengergasse
pat18969@spengergasse.at

2021/2022

Contents

To my family and girlfriend for helping me pursue my dreams
and
To Tobias and Lukas, for enduring me during this work

1 Abstract

With the evolution of our planet and technology, many questions need to be asked, such as what destructive consequences data breaches have on our society, and how to prevent them. In addition, it must be dealt with how a back-end basically works and how one can test them for their errors.
To answer these questions, the following paper explains the importance and functionality of back-end system architecture and the dangers that come from it.

2 Introduction

Data, the internet, servers, keywords that define our present. Fast and reliable data transfer is becoming more and more important, and plays a big role, in entertainment, healthcare, school system and many things more. The project "Babyelefant" is no exception especially when it comes to reliably, capability and security of a back-end.Through our project, and the increasingly importance of Back-end will the following paper be an introduction to Back-end system architecture, then about dangers that come from data breaches and how to prevent them and also an overview of the system architecture of "Babyelefant".

3 What is a Back-end

3.1 Introduction to the Back-end

Tim Berners-Lee, an engineer in Geneva, Switzerland had the idea of the web in the year 1990. The thought behind it was, to make it easier for everyone to create and share hypertext documents through the internet. To implement this, a way to send and receive data had to be created, the Back-end was born. (Gilyadov, 2020) The Back-end is the logical addition to the Front-end, it is responsible to provide the required data to the Front-end. Therefor, the Back-end can also be called the data access layer of software. It can be either accessed directly from the Front-end or called from an intermediate program.

Consequently, the back-end is all the technology needed to handle the approaching requests and create and send the response to the customer. This typically consists of the following three parts:

- The application. This is the application running on the server that is listening for requests, then retrieves the required information from the database, and sends the correct response to the requester.
- The database. Databases are used to store and organize data.
- The server. This is where the computing of the requests take part.

What are the main functions of the app? The server runs an application that contains the defined logic that handles the way the server is responding to various requests. Those responses are heavily influenced by the HTTP verb and the supplied URI (Uniform Resource Identifier). The combination of HTTP verb and the URI is called a route, and the server matching those is called routing.

A portion of the handler functions are the middleware. Middleware is all the source code, that is getting extended between the server getting a request and then sending a response. Thus, the middleware is responsible for computing the request of the client. Therefore, the middleware is capable of modifying the request object, querying the database, or computes the request some other way than this. At last, a middleware function will get called that ends the request-response cycle (typically middleware is not ending the request itself, instead they call other middleware functions) by sending an HTTP response to the client. Developers usually will use frameworks such as Express, Ruby, or in the case of "Babyelefant" Flask to define the logic of the routes.

What is a Server?
A server is simply a computer that got configured to listen for requests, then processes those and sends back a correct response. Typically there are specially made and configured machines for exactly this use case, this ensures the best price-performance ratio, which is especially important in commercial use. Nonetheless, you will often find yourself using your very own machine as a server, this is mostly

the case in the testing or development phases of a project or future product.

What kinds of responses can a server send?
The information that the server sends back can come in various structures. For instance, a server may present a HTML document, send information as JSON, or it may send back just a HTTP status code. You've likely seen the status code "404 - Not Found" when trying to call a website. This status code, for instance, always accrues then, when the called URI does not exist. However, there are far more status codes, that the server can send back to the client. Each one of these is information on what went wrong on the server. This is especially useful for developers and administrators of a website. This allows the precise and fast detection of errors in the source code of a server.

Why do we need a database and what is it actually?
Databases are ordinarily utilized in the back-end of web applications. The database provides an interface that allows the server to save data locally to free up the system's memory. Consequently storing data in a dedicated database saves up both the main memory of the machine and the CPU usage. Furthermore if a crash of the database accrues, the stored data can be retrieved again after rebooting. In most cases, the server has to handle many requests and many different users, therefore also different responses are required. That is why it is necessary to have database queries. A client might request information that is stored in the database, it is also possible that the client wants to manipulate data in the database or add data sets.

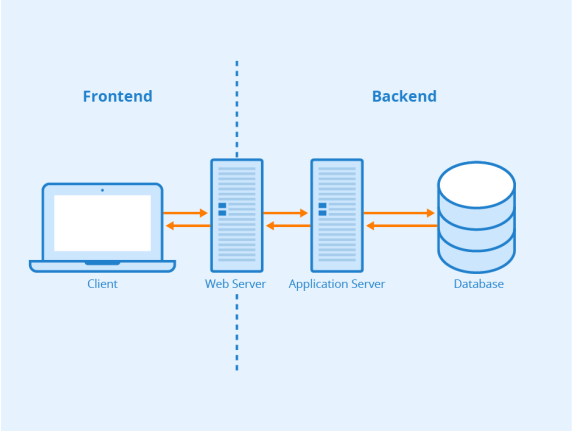


Figure 1: Components of the Web (Medium, 2020)

Figure 1 shows a diagram of a very simplified procedure of the communication of Front-End and Back-end. As shown the Front-End sends a request (represented by the arrows) to the server. There the request gets processed, if necessary, the

web server sends a request for data to the application server, where data will get collected from the database and send back too the Front-End again.

3.2 Purpose of Back-end

The most straight forward purpose of a Back-end is hosting. It is responsible for retrieving, computing and sending data to clients (users of a service).

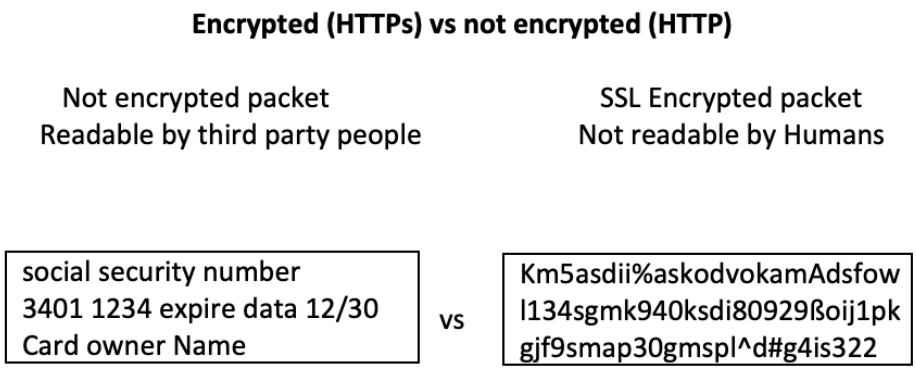
3.3 Functionality and breakdown of a Back-end

In the beginning, the server or Back-end receives a request in the form of an URL. This URL allows the Back-end to gather all the information it needs, to process the request.

A typical URL is build like the following:

`http://example.com/path?query=somevalue`

The beginning of the request is defining the protocol of the request. With that information, the server knows if the request is encrypted or not. There are two options for this parameter, one is HTTP (Hypertext Transfer Protocol) as shown in the example request. This means that the request is not encrypted. Secondly, there is HTTPs (HyperText Transfer Protocol Secure), this protocol indicates that the request is encrypted. However, this does not change how the request is handled by the server.



e

Figure 2: Difference between HTTP and HTTPs

Figure 2 is representing how the protocol is handling the request. HTTP is not encrypting the request which has several security drawbacks. Therefore the use of HTTPs is the industries standard and is insuring a secure and encrypted

connection.
In the second place is the host, but this has no logical effect on the request, the host or domain name only tells the Internet to which server the request is sent. Therefore, each server has its own host or domain. This does not change the request because every request to the server has the same domain.

The path tells the server what the user wants, thus you can say that the path tells the server which piece of code to run, to give the right response to the user. You can imagine the path as the directory path to a certain file.
Last, the URL also consists of query strings. The query string consists of query values which always have a query key. These parameters influence the response of the server to the specific requests.

However, the URL is not sufficient to tell the server exactly what to do. This is where the actions come into play, which are POST, GET, DELETE, and PUT. The URLS in combination with the request action, then deliver a correct response to the user. Another feature is that the server is only accessible from the outside, through predefined routes.Thus, other stored files, such as sensitive data on a server can be stored securely.
Therefore, it is also safe to run a database and a website on the same server, since the server exposes only the website to the Internet. Thus, the server acts as a barrier between the internet and the parts of a website.

3.4 Why a reliable and secure Back-end is important

August 2013, Yahoo's servers got hacked, roughly 3 billion accounts were effected by this data breach. In order to understand the danger posed by data breaches, a few questions must first be answered.

3.4.1 What are data breaches?

The definition of a data breach is as follows: A data breach unintentionally exposes sensitive personal data to the public.This data can be viewed and of course shared with or without malicious intent. This danger exists not only for companies but also for private individuals and governments. Simplified, one can say that every authority is exposed to the danger of a data breach especially without steadily improving security.
This issue has its starting point in the fast advancement of new technologies. Since nowadays, devices of all sorts, are equipped with access to the web, there are additionally an ever-increasing number of potential possibilities to abduct information during the use of systems. Due to the rapid development, it is difficult to protect them at the same time.
Taking everything into account, data breaches happen due to shortcomings of security Features or the false conduct of clients.
Affected devices are, for example, "smart home" products which in the past, had

many security vulnerabilities such as the lack of encryption. Since products are rarely or far too little tested for security vulnerabilities during their development, there will always be problems in the future. (kaspersky, 2022)

3.4.2 How do Data Breaches happen?

The assumption that only hackers are the cause of data breaches is simply unjustified. They are certainly a threat, but data breaches can also be the result of simple mistakes by users or errors in the company's own infrastructure. The following examples could be causes for a data breach:

1. **A Malicious Insider:**
   This is a person or organization that intentionally obtains access to users' data. This data is then passed on or used in other ways to harm a person or company. It is important to note that these people or organizations actually have the right to access the data, only their intentions are different.

2. **An Accidental Insider:**
   This type of data breach is much less dangerous. For example, an employee of a company that processes sensitive data turns to a computer that is unlocked. The employee is therefore not authorized to see the data, but does so unintentionally. Basically, no data is shared, but it is still called a data breach, because unauthorized persons had access to protected data.

3. **Malicious Outside Criminals:**
   This type of data breach is led by criminal individuals or organizations. They use various hacking methods to gain access to the systems and thus also to the data of companies or individuals. In most cases, the data is used to either harm the company itself or to extort money for secrecy.

4. **Lost or Stolen Devices:**
   Finally, lost devices such as laptops, hard drives, or smartphones are also types of data breaches, but in most cases not on a threatening scale. In most cases, these devices are encrypted themselves, so the risk of a threat can be drastically reduced.

3.4.3 Malicious Methods used to Breach data.

Data breaches are mostly the result of cyber attacks. The most commonly used are:

1. **Brute Force Attacks:** Brute force is a simple yet effective way to gain access to data. It works in such a way that a specially programmed software uses randomly generated passwords until the moment when one of these passwords is correct. This method is becoming more and more effective due to the increasing power of processors, especially when the password used for protection is not complex.
   Most of the time, the people/organizations that program such software also make use of malware, which makes it possible to crack the password with several computers at the same time.

2. **Phishing:** This is probably the most common method of data theft, sending emails or SMS to random people under the guise of a trusted company. Popular disguises for this method are for example Amazon, Facebook or even banks. The people then pass on their data "voluntarily".

3. **Malware:** Since most of our communication today takes place digitally, there is also for more and more devices specially developed software, but it is often the case that due to too few security tests, gaps in the software are found from time to time, which can be used by hackers. Malicious software, so-called malware, is then infiltrated into these found security gaps. This then reads the data traffic and sends it to the people who have installed this malware.

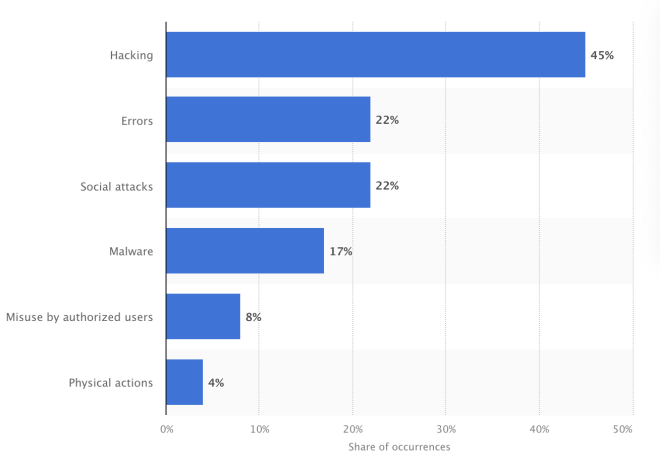


Figure 3: Most common action varieties in data breaches (kaspersky, 2020)

The figure shows the most common causes of data breaches worldwide in 2019.

3.4.4 What are the goals of Data Breaches?

The most common goal of a data breach is to gather personal user information of users of a particular company, this data is then sold or is used to harm the company that is taking care of the data.

The process of a data breach is the following: Identifying an organization to be victim of the data breach - finding vulnerabilities is the next and most complex step, such vulnerabilities can be all sorts of things, for example overdue software updates, ports that are open or even, as already mentioned, phishing attacks that have happened within a network - usually the mentioned malware is introduced, which then observes the data traffic and sooner or later also discloses the infrastructure of the network as well as passwords - now hackers have full access to the network, they can freely view data and of course also pass it on for their use.

3.4.5 How significant is the damage caused by a data breach?

The problem with being a victim of a data breach is that a simple password change is not enough, even if you try to cover up the consequences of a data breach, there are always consequences. Most of the time you lose money, reputation and other things.

Of course, the importance of data breaches is assessed differently in different sectors. You can divide them into three categories:

1. **Individuals:** With data breaches of individuals, the danger lies in identity theft. There are several types of data breaches, including social media accounts, bank information, access data to stock portfolios, and so on. From the moment people have this data, they can make unnoticed and anonymous payments under the name of the victim, spread false opinions and much more. Most of the time, however, it is the theft of money from the victim's accounts. As a result, the victims become heavily indebted and do not realize it, and it is difficult to prove that they have been the victim of an attack. However, there are far more consequences victims of data breaches can experience, it is important to protect your personal data as much as possible. In addition, there are websites that check whether your data has been in a data breach.
2. **Government organizations:** For government, data breaches are also associated with immense consequences. It can lead to the disclosure of secret operations of the intelligence service. Military operations, arms purchases, and soldier readiness can be made public.

Even government campaigns, plans, and unpublished government decrees or decisions can fall into the wrong hands. Also the criminals can possibly change said decrees in their favor. Therefore, data breaches are especially threatening to governments and their citizens, as they can lead to political as well as social unrest.

3. **business organisations:** The worst consequences, probably have to bear companies. A data breach is evidence of low security for users of the service. Of course, it also has financial consequences, as people who have become victims of the data breach may sue the company. The worst consequences, however, are that users lose confidence in the company and then switch providers.

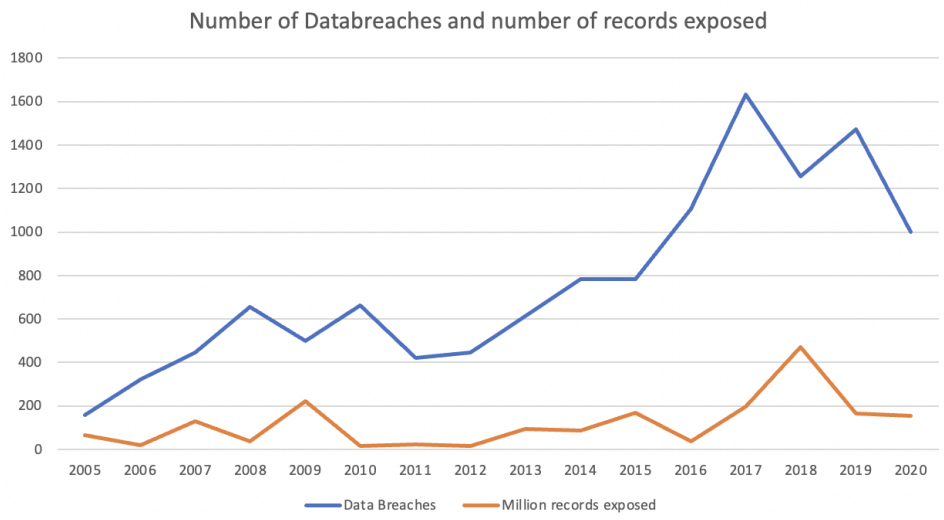


Figure 4: Data Breaches 2005-2017 (Statista, 2021)

The figure shows how the number of data breaches is developing. It is clear that with the years and the increasing technical development, there are also more and more security gaps that can lead to a data breach.

3.5 General Concept of Back-end System Architecture

What is Back-end System Architecture
According to (Medium, 2019), Back-end architecture is "... a conceptual representation of the components and sub-components that reflect the behavior of the system." Summarized is Back-End System Architecture the designing and planing of the construct, on which the Back-end will function on. It is especially important for scale-ability, sustainability and functionality and all of this at a reasonable use of hardware and other resources.

What are the requirements for the design of System Architecture?
There are several requirements to a good design System Architecture

1. High extend-ability: by adding hardware to the system, it can operate and handle more requests
2. High availability: the system has to operate 24-hour/365-day without any or as little downtime as possible
3. High performance: it has to have linear increase of performance when adding linear more hardware
4. High security: the system has to operate on the highest security standards, especially, when sensible information about users is computed.

For production ready Back-end, all mentioned points are important to achieve in system architecture design. Especially the ability of high extendibility is important in modern days, because of rising data trans-far rates and bigger files that are getting transferred, however, services are usually not starting out big, they are rather growing over time this is where the extendibilty is really showing its effect. Furthermore high performance respectively high efficiency is another highly important aspect to consider in the design of the system architecture. The reason for this is that hardware is an incredibly expensive investment, which companies generally want to avoid. A good example of implementing the mentioned requirements is Netflix.

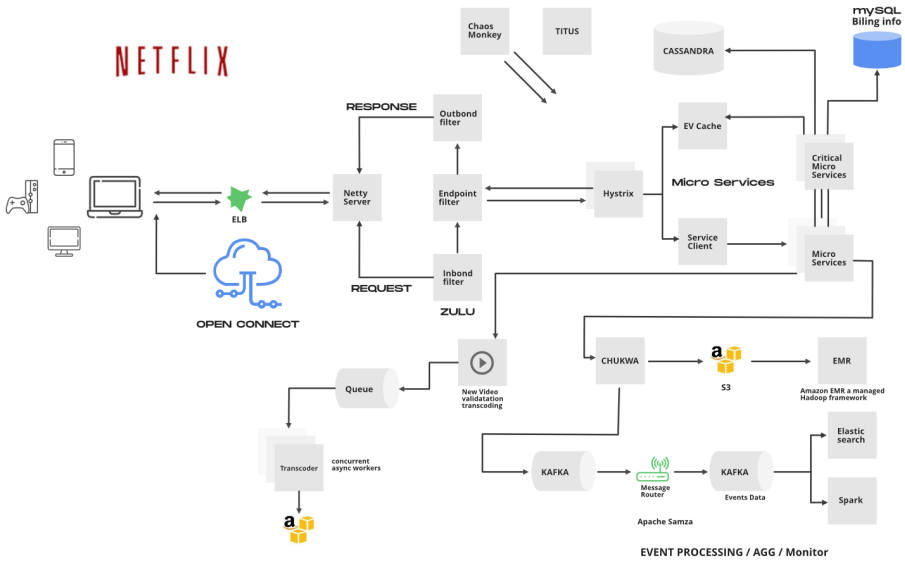


Figure 5: Netflix System Architecture (GeeksForGeeks, 2021)

The figure represents the system architecture of Netflix. Netflix's application mainly consists out of 3 major components. (GeeksForGeeks, 2021)

1. **The Client:** the client is what is used to playback the content Netflix is providing. Devices like phones and TVs are the most common used devices.
2. **Open Connect or Netflix CDN:** CDN (Content delivery network) is the network of distributed servers around various geographical location. This is used to ensure fast response times. Means when a client is trying to stream some kind of show, the client will connect to the closest available server and not the main Netflix server.
3. **The Back-End/Database:** Netflix's Back-End is not responsible for the streaming of its content. It is handling tasks like on boarding new content and distributing the new content on the CDN servers around the world.

3.5.1 Design of the used Back-end

We have implemented the system architecture of "Babyelefant" as follows:

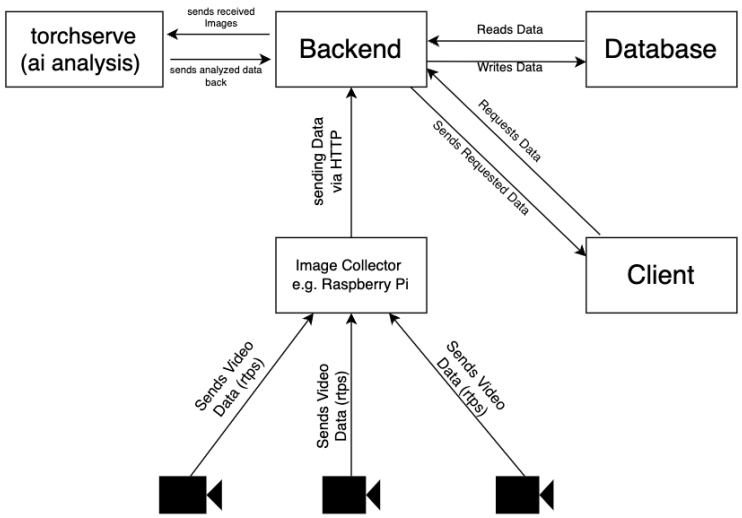


Figure 6: simplified system architecture of Babyelefant

The figure represents the system architecture of Babyelefant. The process is quite simple, the cameras of a client are connected to a local image collector. This image collector can be any pc. It can connect to multiple cameras and collects images coming from the cameras using the RTSP (Real-Time Streaming Protocol). These images are then sent from the image collector via HTTP to the backend. The backend now receives the images and forwards them to the torchserve, where the AI calculates distances and other values. These values are then stored in the database.

If the user now wants to view data, he makes a request to the server, where the live stream of the cameras is then forwarded to the client. Also data from the database is sent to the client, so that the client can view it.

3.5.2 The Advantages of the Design

Postgres is inherently more powerful because it supports concurrent writes without the need for read/write locks. In addition, Postgres is fully ACID compliant and fully supports transaction isolation and snapshots. In addition, Postgres has a high-security standard, scales very well, and is not error-prone during runtime.

3.6 Communication between Front-End Back-end

Why do the Back-end and the Front-end exist in the first place?

The Front-end and Back-end serve the purpose that a user can open a website and collect useful information at any time and from anywhere. An example of a user could look like the following:

- The client directs their program toward one of the site's URLs
- This immediately causes the clients browser to send out multiple requests to the server
- The clients browser waits until it gets a reaction/response from the server - then each section of the response, is used to to render a certain part of the website in the clients browser.
- The clients browser finished loading the web-page, and is now ready to be used. The user can interact with the website, those interactions are causing even more requests to be sent to the server.

To sum it up, the Back-end and Front-End are making the use of a web-page possible by sending requests and responses to each other.

When does the browser really communicate with the Back-end?

The communication between Front-End and Back-end is always build on HTTP requests sent from the browser, that are arriving at the Back-End. The considered requests can send data in, e.g. Json, or also in the request header, which has either the sense to get new data with certain parameters, or also to send new data to the Back-end. The used HTTP requests are built in the clients browser and then sent to the server.

When does a request get triggered?

Requests are mainly sent, when a client is clicking on a link, but there also a reason like:

- resources that have to be loaded in the HTML file itself - such as Images, style files or other files that are necessary to ensure a fully usable web-page
- code that gets executed in the background, often need further resources to function
- the browser has to request more URLs that are corresponding to the new page

How is the Back-end communicating with the Front-end
The front end and the back end usually communicate via HTTP request and response payloads. There are several content variations in which the Back-end is normally responding to requests:

- static files like CSS, JS or image files
- HTML files
- data in format of JSON
- no response body but a status code that the browser can then work with

Also the Front-end has several options of sending data:

- Form data like user name and password
- data in format of JSON
- simple HTTP requests without any body

3.6.1 What is an API?

An application programming connection point (API) is a bunch of strategies, conventions, and instruments for making programming applications. An API communicates a product part as far as its tasks, sources of info, results, and essential sorts. APIs characterize capacities that are free of their separate executions, so definitions and executions can contrast without influencing one another. A decent API works with program advancement by giving all the structure blocks. APIs are regularly given as a library that contains particulars for schedules, information structures, object classes, and factors. In different cases, particularly for SOAP and REST benefits, the API is essentially a detail of the remote calls given by the API buyer.
An API in particular can take many structures, e.g., a worldwide standard, for example, POSIX, seller documentation like the Microsoft Windows API, or programming language libraries like the C++ standard format library or the Java API.
An API contrasts from an Application Binary Interface (ABI) in that an API depends on source code, while an ABI is a double connection point. For instance, POSIX is an API, while the Linux standard base is an ABI.

What are types of an API?

1. **Private APIs (internal APIs):** those are only internal used APIs, means that a company might publish this interface to its employees. Other companies or private individuals, don't have access to the API
2. **Partner APIs (restricted APIs):** only partner companies or individuals that got chosen from the provider of the API are able to obtain data. This usually comes as a paid service.
3. **Public APIs:** this type of API is available for everybody, that wants to get use of the provided data. Typically there are no restrictions to this type of API.

(ComputerWeekly, 2020)

Furthermore there are several classifications of APIs, such as:

1. **Web-APIs:** this category of APIs are design in a way, to call ordinary files like HTML-pages via a simple HTTP protocol. This categorie of API is also called REST (Representational State Transfer) or RESTful API
2. **Local APIs:** this category is usually used to provide software- or middleware-services
3. **Program-APIs:** this category is based on RPC (Remote Procedure Call), this ensures that a part of the software appears as local for the rest

Why is an API so important for companies?
The interest and demand for APIs is steadily increasing, with technologies such as the Web, information-sharing software, and even technologies such as cloud computing continuing to drive this demand. Software that was once developed for a specific purpose is now often written with references to APIs that provide generally useful functionality, reduce development time and cost, and reduce the risk of errors.

APIs have steadily improved software quality over the past decade, and the growing number of web services provided by cloud providers via APIs is also driving the development of cloud, IoT (Internet of Things) and mobile applications.

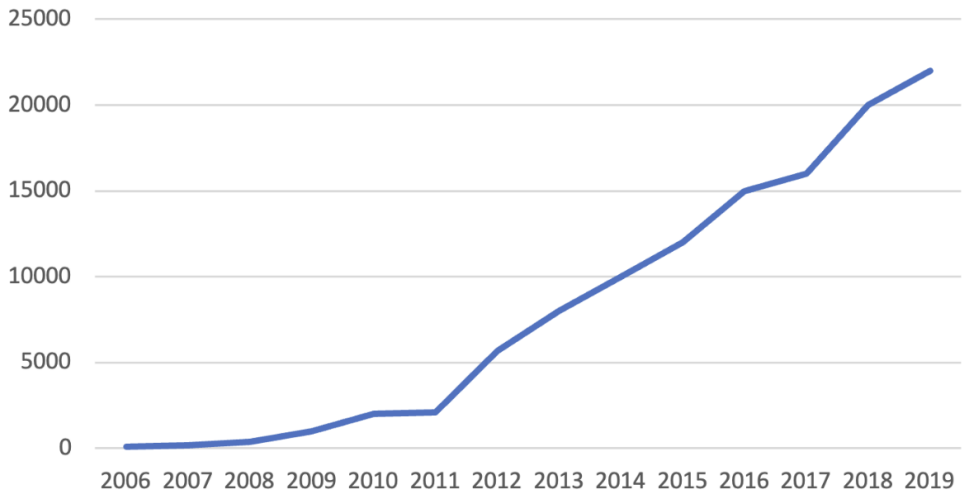


Figure 7: Growth of numbers of web APIs since 2006

The Figure shows, the growth of the numbers of web APIs available since the year 2006. It is clear to say, that APIs get more and more popular and necessary for most businesses.

APIs in the cloud?
The above mentioned cloud cumputing, enables new possibilities in software development. Software can be divided into reusable components, which can be scaled as needed. This leads to many APIs services shifting to so-called functional programming and lambda services which can be scaled immediately in the cloud as needed.

APIs as a service
The trend of offering APIs as a service and thus making them available to the public as a tool is becoming increasingly popular. However, this also means that APIs as such also require correspondingly controlled development and deployment.

SOA(service-oriented architecture) and **Microservices** are well-known examples of service APIs. APIs are increasingly being developed as services. This is why many software developers are talking about the fact that APIs will soon be developed as services.

API testing
Like any software, APIs must be tested for correctness. This usually has the purpose of testing whether a valid response to the request is returned with the given specifications of the user. These tests are usually performed as part of ALM (Application Lifecycle Management).After the API has been published, it must also be tested whether the API can be accessed as planned or whether there are security vulnerabilities.

Types of API testing
The process of testing and validating an API takes lots of resources and time, it is an essential step, before releasing the finished product to the public. With the increasing complexity and importance of APIs, more and more testing methods have evolved. The following are currently the most popular and most used. (Wallarm, 2020)

1. **Validation Testing:** Validation testing is performed at the end of the development phase and is therefore one of the most important steps in software development. Validation testing ensures that the API has been developed correctly, that it returns correct results, and that it does so while working efficiently and safely.

2. **Functional Testing:** This type of testing calls certain functions and processes of the API to check if the API acts correctly in certain scenarios.

3. **Security Testing:** This type of testing verifies that the implementation of the API has been done without security vulnerabilities. Most often, these tests also include validation of encryption methods and access control of the API. Also user authorization scenarios and authorization is tested.

4. **UI Testing:** UI Testing is the testing of the user interface with which the user accesses the API. However, this testing method does not deal with the API but mostly with the user interface. These tests do not provide useful information about the API itself, but they can provide valuable data about the usability and efficiency of the associated user interface.

5. **Load Testing:** Load testing usually takes place after the completion of the entire codebase or a specific part of it. This testing method checks whether the developed solution really works as planned. These tests are performed at low but also at a high load of the system.

6. **Penetration Testing:** Penetration testing is the second significant test block in the development, validation, and approval period of an API. Users with next to zero information about the API attempt to infiltrate the server utilizing different techniques. If an attacker finds a loophole in these tests that poses a security risk, he reports the error to the provider of the API.

7. **Runtime and error detection:** This test method relies on the actual usage of the API. The method focuses on one of the following aspects: monitoring, execution errors, resource leaks, or error detection. With the gathered information, providers can fix errors in the codebase that could lead to the failure of certain parts of the API

8. **Fuzz testing:** Fuzz testing is another step in the verification of the security features of an API. It involves sending random data (also called "noise") to the server to determine if the system will crash or have other unplanned consequences. Thus, this method exhausts the limits of the API by simulating the worst case.

Best practice for API testing

To the different types of tests, one must of course also consider the execution of these, there are some good practices that should be followed.

1. **Using Realistic Data:** This may sound obvious, but testing with data that is as realistic as possible is best because it allows you to run real-life scenarios. This leads to more accurate test results, far-reaching insights into the behavior of the API on specific and also uncovers bugs in the software.

2. **Triggering Negative Results:** Positive experimental outcomes are not dependably the right or significant thing to focus on while testing an API. Numerous developers only test for positive outcomes, however, it should be noted that you often only find errors if you provoke them. That's why API tests should rather focus on achieving unsatisfactory results in order to hide the software or to fix bugs.

3. **Storing Test API Responses:** Another aspect that good API tests take into account is that you should save the test responses of the API. This data is important to be able to reconstruct the development of the API. This data can help to detect errors or to get insight into how the required computing power and therefore the costs develop.

4. **Reusing API Tests for Performance And Security Tests:** Another essential factor for good and efficient API testing is that you should use already created API tests, for example already created for function testing, also for security or performance tests. In the best case, one uses the already defined simulated real data.

(SoapUI, n.d.)

3.7 Microservices

3.7.1 What are microservices?

Microservices, or the microservice architecture, make it possible to easily scale the service if needed. However, services are not directly connected to each other. Basically, you can say that every function is a microservice. A microservice would be, for example, logging in. More precisely, microservices are an information technology architecture pattern in which complex application software is generated from independent processes that communicate with each other using language-independent programming interfaces. The services are largely decoupled and perform a small task. (Stackoverflow, 2020)

3.7.2 How do microservices work?

The microservice architecture partitions a product framework into an enormous number of individual, little and independent services. These run as individual processes (thus usually also distributed over a network) and communicate with one another through simple network interfaces.
Compared with conventional "monolithic" software, this architecture offers several benefits - for instance, individual services can use different technologies. While Java as a runtime environment and MySQL as a database management system may be suitable for one service, Node.js and a Mongo database may be better candidates for another. Encapsulation into individual services thus enables development teams to choose the best tool for the purpose.
The division into several individual components can also simplify the delivery of new versions: Instead of deploying the complete system in one piece (a potentially risky undertaking, especially for complex software systems), individual services can now be delivered independently of one another (or even replaced by new implementations).

In contrast to conventional service-oriented architectures (SOAs), the focus of microservice architectures is on simplicity. The functional scope of a single service should be as manageable as possible. (NetApp, n.d.)

3.7.3 Why should microservices be used?

In contrast to the old-fashioned monolithic applications, microservices have the following advantages. They are confident and can fail without the whole system collapsing. Also, it is much more manageable, so if you expect the developed software to be worked on later by many people, you should switch to microservices. Another advantage of microservices is that the architecture offers great potential for scaling. For this reason, microservices should be used above all when it can be assumed that the number of users will increase significantly. (Stackoverflow, 2020)

3.7.4 How does a typical microservice architecture look like?

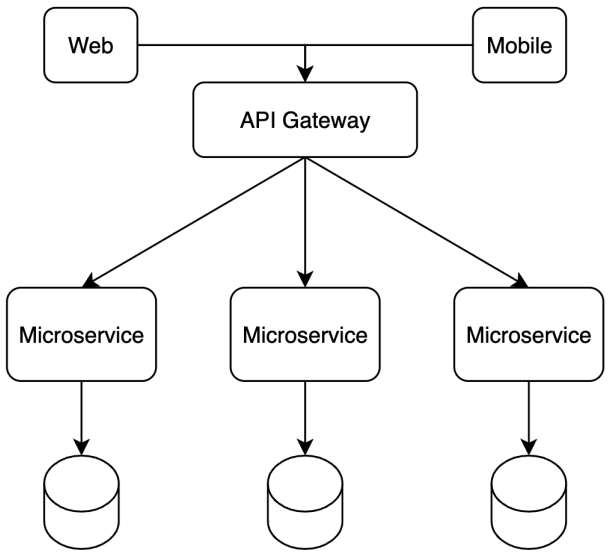


Figure 8: Simplified architecture of microservices

The figure shows a simplified version of the structure of a microservice system architecture. Each microservice has its own database and can function completely independently from each other. Each microservice has a specific function. In most cases, each microservice is developed by a separate team.

4 Storing Data

As programs and files become larger and more complex, memory capacity and also the method of storage must continue to evolve.In the following section therefore the topic data storage is worked off.

4.1 What is Data Storing

Data storage essentially means that documents and files are digitally captured and stored in some type of storage system. These systems may rely on optical, electromagnetic, or other media to protect and recover data when needed. Data storage facilitates the backup of records for monitoring and rapid recovery in the event of a surprise processing crash or cyberattack.
To store said data, there are several options, the most popular being hard drives, USB drives or even clout solutions. Data storage systems should be reliable, cost-effective and secure. (CDW, 2021)

4.2 Types of Data Storage

Direct Area Storage, also called Direct-Attached Storage (DAS), is, as the name implies, direct storage. This storage is often located in close proximity and is directly connected to the computer that accesses it. Often, this is the only computer connected to it. DAS can also provide good local backup services, but sharing is limited. DAS devices include floppy disks, optical disks (CDs and DVDs), hard disk drives (HDD), flash drives and solid-state drives (SSD).

Network-based storage allows multiple computers to access data over a network, making it more suitable for data sharing and collaboration. The ability to store data off-site also makes them better suited for backups and data protection. Two common network-based storage configurations are network-attached storage (NAS) and storage area network (SAN). (Price, 2020)

4.3 What Database did we use and why?

For the "Babyelefant" project, we decided to use a PostgreSQL database. This offers excellent scaling potential and the ability to store large amounts of data quickly and securely. Another aspect is that, due to the sensitive data, we needed a database that guarantees the highest security. This is the case with PostgreSQL.

4.4 What are the Difficulties regarding our Project

Essentially, the trouble with storing data was tracking down an approach to securely and quickly store complex, large, and most importantly, lots of data. Since "Babyelefant" was intended to work with a vast number of cameras that include a consistent stream of information, we had to be careful not to overload the database when storing data.

4.5 Image Storage

4.5.1 How are Black and White pictures stored?

To understand how images are stored, one must first understand what image data is.

Figure 9: Picture of black and wight eight (analyticsvidhya, 2021a)

The figure shows a picture of an eight, which has a resolution of 24x16. The computer does not see the image itself but only the data for the pixel. This is shown in the following illustration.

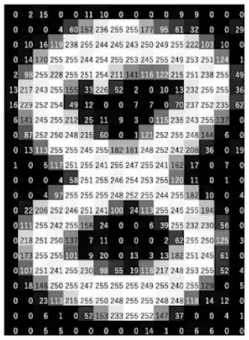


Figure 10: Picture of eight divided into boxes (analyticsvidhya, 2021b)

These small boxes are called pixels, so the resolution is the dimensions of the image in x and y coordinates. Each pixel has a white value which can range from 0 (black) to 255 (white). Thus images are stored as numbers in a so-called matrix.

4.5.2 How are Colored Pictures stored?

Colored images are stored in a similar way, but with some additional information necessary to actually display the image. Therefore, the image is divided into three images, one with only red colors, one with only blue colors, and one with only green colors. Again, each pixel can have a value from 0 to 255 to represent different color tones. These matrixes are then saved and when the image is viewed these three layers are merged back into one image. The following image represents how a normal image is divided into its primary colored components.

Figure 11: Picture divided into the 3 color layers (analyticsvidhya, 2021c)

4.6 Popular Providers for Cloud Storage

1. **Firebase:** Firebase is a product of Google, which dates back to 2011. It provides services such as authentication, real-time databases, and storage. One of the most popular products is the real-time database from Firebase. This is a cloud storage solution that runs with NoSQL. Data is updated in real-time for all users and is provided in JSON format.
   The advantages of Firebase Realtime Storage are, for example, that it is easy to manipulate data as an admin and that this changed data is then quickly sent to all clients connected to the session. Also, the behavior in case of loss of connection to the database is excellent, since the client can interact with the data in any case, and afterward, when it is reconnected, it can see data that was modified or recorded when the client could not connect. However, the biggest advantage over competitors is that the database constantly provides its clients with updates, which means that as a client you don't have to make constant requests to get the latest data. (GeeksforGreeks, 2020)

2. **AWS** One of the most popular and generally used AWS services for securely storing data is Amazon S3. Amazon Simple Storage Service (Amazon S3) is an object storage service with incredible performance, data availability, security, and scalability. Clients of all sizes and enterprises can store and protect any amount of data for practically any use case, including data lakes, mobile apps, and native applications. In addition, AWS provides features that are ideal for optimizing costs, organizing data, and customizing access control for specific uses. (AWS, 2021)

The following chart illustrates an estimate from 2016 about the development of cloud storage users:

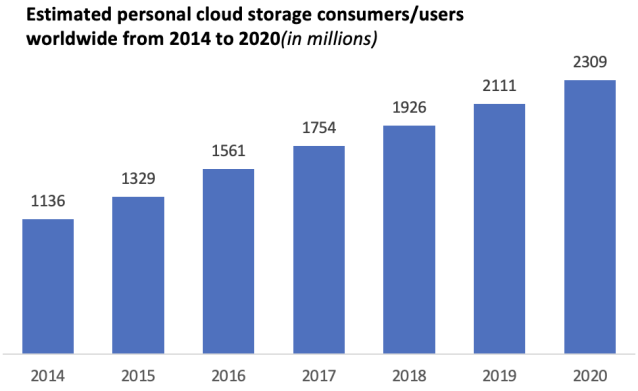


Figure 12: Number of cloud storage consumers (Statista, 2016)

List of Figures

References

analyticsvidhya. (2021a). How images are stored in the computer? Retrieved from https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/#:~:text=Images%20are%20stored%20in%20the%20form%20of%20a%20matrix%20of,the%20intensity%20of%20each%20pixel.

analyticsvidhya. (2021b). How images are stored in the computer? Retrieved from https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/#:~:text=Images%20are%20stored%20in%20the%20form%20of%20a%20matrix%20of,the%20intensity%20of%20each%20pixel.

analyticsvidhya. (2021c). How images are stored in the computer? Retrieved from https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/#:~:text=Images%20are%20stored%20in%20the%20form%20of%20a%20matrix%20of,the%20intensity%20of%20each%20pixel.

AWS. (2021). Amazon s3 objektspeicher für den abruf beliebiger datenmengen von jedem ort aus. Retrieved from https://aws.amazon.com/de/s3/?did=ft_card&trk=ft_card

CDW. (2021). How do microservices work? Retrieved from https://www.cdw.com/content/cdw/en/articles/datacenter/what-is-data-storage.html

ComputerWeekly. (2020). Programmierschnittstelle (application programming interface, api). Retrieved from https://www.computerweekly.com/de/definition/Programmierschnittstelle-API

GeeksForGeeks. (2021). Statistics on personal data breaches and disclosures in the united states in the years 2005 -2017. Retrieved from https://www.geeksforgeeks.org/system-design-netflix-a-complete-architecture

GeeksforGreeks. (2020). How to save data to the firebase realtime database in android? Retrieved from https://www.geeksforgeeks.org/how-to-save-data-to-the-firebase-realtime-database-in-android/

Gilyadov, L. (2020). history of front- and back-end development. Retrieved from http://www.developer-cheatsheets.com/history.html

kaspersky. (2020). Most common action varieties in data breaches worldwide in 2019. Retrieved from https://www.statista.com/statistics/221390/share-of-hacking-methods-across-organizations/

kaspersky. (2022). What are data-breaches. Retrieved from https://www.kaspersky.com/resource-center/definitions/data-breach

Medium. (2020). Front end vs back end. Retrieved from https://tinyurl.com/56wu2b25

NetApp. (n.d.). How do microservices work? Retrieved from https://www.netapp.com/knowledge-center/what-are-microservices/#:~:text=A%20microservice%20attempts%20to%20address,rest%20of%20the%20microservices%20architecture.

Price, M. (2020). How do microservices work? Retrieved from https://www.datacenters.com/news/different-data-storage-types-which-is-right-for-your-business

SoapUI. (n.d.). 5 best practices for data driven api testing. Retrieved from https://www.soapui.org/learn/api/5-best-practices-for-data-driven-api-testing/

Stackoverflow. (2020). What are microservices. Retrieved from https://stackoverflow.com/questions/46575898/what-is-a-microservice

Statista. (2016). Forecast number of personal cloud storage consumers/users worldwide from 2014 to 2020. Retrieved from https://www.statista.com/statistics/499558/worldwide-personal-cloud-storage-users/#:~:text=The%20statistic%20depicts%20the%20number,be%20using%20personal%20cloud%20storage.

Statista. (2021). Statistics on personal data breaches and disclosures in the united states in the years 2005 -2017. Retrieved from https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

Wallarm. (2020). What is api testing? Retrieved from https://www.wallarm.com/what/what-is-api-testing-benefits-types-how-to-star

Parameter and calibration procedures and HID for accurate distance measurement

Tobias Pressler
HTL Spengergasse
pressler.tobias@gmail.com

2021/2022

Contents

To my parents, for helping me on my way
and
To Lukas and Moritz, for supporting me
and
To Anna, for always being by my side

1 Abstract

Accurate distance measurement using images is crucial in tasks like crowd counting or ensuring that pedestrians keep a given distance between each other. Images cannot be directly used for this task, as perspective distortions are introduced by the camera. Therefore, the images have to be transformed into the bird's eye perspective, as no distortions are present at this viewing angle on the ground plane. In most cases, no specialised hardware is available at the scenes to retrieve the information needed from the environment to transform the image.

This paper therefore introduces a method for predicting the homography matrix needed to transform the images into the bird's eye perspective from a single image. Additionally, a reference distance is used be able to calculate all distances on the ground plane.

2 Introduction

In 1993, the CMOS camera sensor was invented by a team at the NASA Jet Propulsion Laboratory (Fossum & Hondongwa, 2014). Based on this invention, camera sensors became smaller, cheaper and more power efficient. In our modern world, nearly every device has a camera sensor build in and with the introduction of 4G and 5G, uploading a live video feed to the internet became possible. Additionally, almost every public place is being filmed by a webcam and surveillance systems are being used everywhere. For tasks like checking if distance is kept between passengers or crowd counting where also crowd density is being calculated the easiest solution would be to directly use these images. It would require the least amount of physical work at the targeted location and setup would be fast and easy. Unfortunately, cameras introduce perspective distortions into the final image. Humans have developed stereo vision, which uses the displacement between objects seen by both eyes to perceive depth, therefore the distortions do not impair their everyday life. Specialised hardware is available which also uses two cameras next to each other to perceive depth, but this approach is expensive and would need those cameras to be installed. However, with the introduction of neural networks, the needed information can be predicted and therefore an image without the distortions in the bird's eye perspective can be constructed.

3 Homogeneous coordinates

Homogeneous coordinates allow for multiple operations to be expressed using matrix multiplication, therefore it is often used in computer vision tasks. In a 2D Euclidean space, rotation and scaling can be done using matrix multiplication. To rotate a point (x, y) by 90 degrees and scale it up by a factor of 2, the operation can be expressed using

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

However, translation cannot be expressed using matrix multiplication in the Euclidean coordinate system. Homogeneous Coordinates therefore introduce an additional dimension, called the projective space. When transforming a point into the corresponding homogeneous coordinates an additional dimension s which is equal to 1 is added:

$$x = \begin{bmatrix} x \\ y \end{bmatrix} \quad \Rightarrow \quad \widetilde{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

Additionally, homogeneous coordinates can be transformed back into the corresponding Euclidean vector form by dividing each element of the vector, except for the last one, by the last element s.

$$\widetilde{x} = \begin{bmatrix} x \\ y \\ s \end{bmatrix} \quad \Rightarrow \quad x = \begin{bmatrix} x/s \\ y/s \end{bmatrix} \tag{3}$$

As s cannot only be 1, homogeneous coordinates which are a multiple of another homogeneous coordinate by a factor of s therefore represent the same point in the Euclidean space. For example, $(3, 3, 1)$ and $(9, 9, 3)$ are the same 2D vector $(3, 3)$. When s is equal to 0, the homogeneous coordinate represents a point at (∞, ∞) as $(x/0, y/0)$ always approaches ∞. Infinite Euclidean coordinates can therefore be expressed without using ∞ using homogeneous coordinates (Jia, 2020).

4 Pinhole camera model

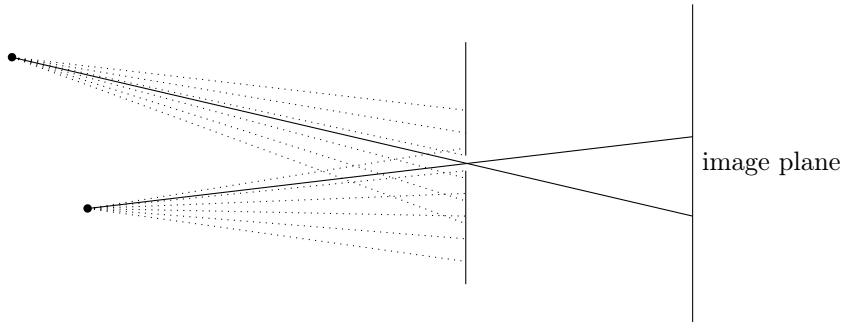


Figure 1: Pinhole camera

A pinhole camera works by only capturing light rays, that pass through the pinhole of the camera, all other light rays are blocked. In theory, the pinhole, also called camera centre has to be infinitely small to correctly block all unwanted light rays. Every ray that enters the pinhole is being mapped on the image plane. As all light rays cross the pinhole, the resulting image is flipped horizontally and vertically (Ray, 2002, p. 34).

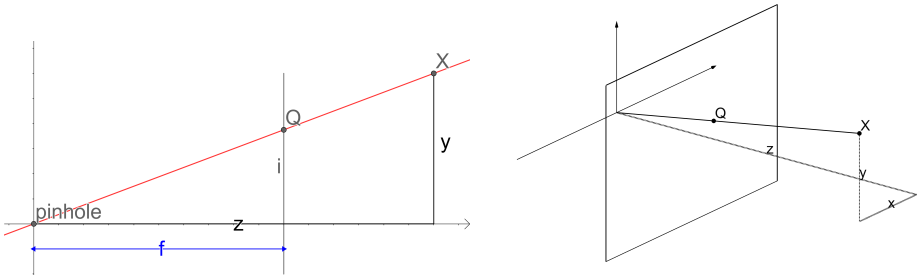


Figure 2: Geometry of a pinhole camera

Figure 2 visualises the pinhole camera model, which is used to mathematically describe a pinhole camera. The line perpendicular to the camera centre, in this case, the z-axis, is called the principal axis and the image plane i is drawn in front of the pinhole for simplification. The image plan is spaced apart from the camera centre by the length f, also called the focal length. Using the model, a point in 3D space with coordinates $X = (x, y, z)$ is mapped to the point Q on the image plane by joining X with the pinhole (Hartley & Zisserman, 2004, p. 154). The coordinates of the point Q can then be calculated using similar triangles, therefore

$$Q = (\frac{fx}{z}, \frac{fy}{z}, f) \quad \Rightarrow \quad (\frac{fx}{z}, \frac{fy}{z}) \tag{4}$$

As seen on equation (4), the z-coordinate can be removed, as it will always be equal to f, therefore the final coordinate is in 2D-space. Equation (4) can also be expressed using homogeneous coordinates:

$$\begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5}$$

As the principal point, where the principal axis intersects with the image plane, is in general not at $(0, 0)$, a transform is needed to adapt the coordinates (Hartley & Zisserman, 2004, p. 156). As we used homogeneous coordinates in equation (5), we can easily add the transform into the existing equation

$$\begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{6}$$

where p_x and p_y are the coordinates of the principal point. Most cameras in the real world have their principal point at the centre of the image plane, therefore $(\frac{w}{2}, \frac{h}{2})$.

To simplify equation (6), the focal length and coordinates of the principal point are separated into an extra matrix. It is called camera calibration matrix K being:

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{7}$$

The rewritten equation now being:

$$Q = K[I|0]X \tag{8}$$

where $[I|0]$ is a 3x3 identity matrix plus a zero column vector, X is the world point represented by a homogeneous 4x1 vector and Q is the resulting point on the image plane, but as a homogeneous vector (Hartley & Zisserman, 2004, p. 156).

The point X of equation (8) is expressed in the camera coordinate system, meaning that the origin of the Euclidean coordinate system is at the camera centre and the principal axis is pointing down the z-axis. In general however, the points in space will be expressed using the world coordinate system. Therefore, the points have to be translated and rotated to convert them to the camera coordinate system (Hata & Savarese, 2022). The point X in the camera coordinate system can be calculated using

$$X = R(x - C) \tag{9}$$

where x represents a point in the world coordinate system using an inhomogeneous 3x1 vector, R is a 3x3 rotation matrix and C is the coordinate of the camera centre in the world coordinate system. The point x is therefore translated and then rotated into the cameras coordinate system (Hartley & Zisserman, 2004, p. 156).

Combining this with equation (8) results in

$$Q = KR[I| - C]x \tag{10}$$

As the parameters of K are only defined by the camera the image was taken by, they are called the internal camera parameters, it therefore never changes for a camera. On the other hand, R and C are called external camera parameters as they are defined by the external environment, therefore the position of the camera in the world coordinate system (Hata & Savarese, 2022).

Finally, equation (10) can be truncated to its final form:

$$Q = K[R|t]x = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{11}$$

4.1 Aperture

The ideal pinhole camera model assumes, that the pinhole is infinitely small. As this is not possible in the real world, the pinhole will always have a given size called the aperture. Ideally, when the camera centre would be infinitely small, only one light ray of each point in the real world would enter the pinhole and would be mapped on the image plane. When the aperture is too big, multiple light rays with different angles but the same origin can enter the camera centre and therefore result in a blurred image. On the other hand, when the aperture is too small, the resulting image gets very dark as most light rays are blocked by the small aperture (Hata & Savarese, 2022).

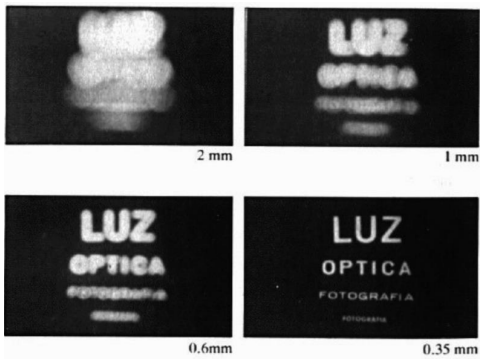


Figure 3: Different aperture sizes (Hecht, 2010)

When making the pinhole as small as possible like the pinhole camera model specifies, the image not only gets darker, but also the phenomenon of diffraction can be observed. The light waves bend around the edge of the aperture and the image therefore gets blurry (Ray, 2002, p.34). When the aperture is a perfect circle, then the optimal pinhole size with the sharpest image can be calculated found with the Frauenhofer approximation using

$$d = \sqrt{2.44}\sqrt{f\lambda} \tag{12}$$

where λ is the wavelength of the light passing through, which in this case would be $550nm$ for yellow-green light and f is the focal length as described in the pinhole camera model in section 4 (Hata & Savarese, 2022).

4.2 Camera lenses

As only very little light passes through the pinhole, a long exposure time is needed to create a bright image. Modern cameras record videos at a high image rate, therefore the original pinhole camera would not be suitable. To mitigate the conflict between sharpness and brightness, modern cameras use lenses. With a properly placed and sized lens, all light rays emitted by the point X are being refracted by the lens and therefore converged to a single point Q (Hata & Savarese, 2022).

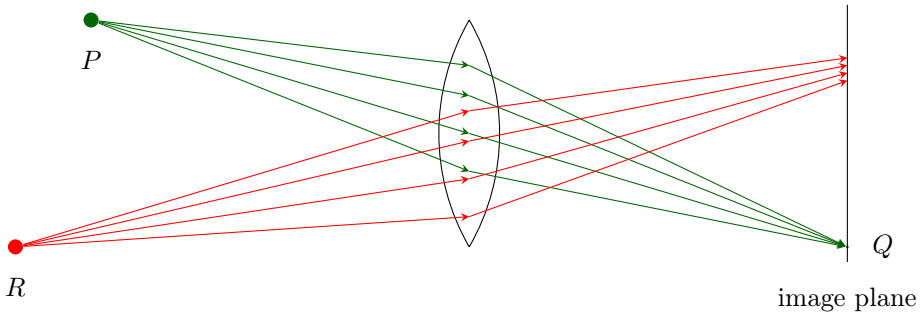


Figure 4: Setup of a lens model

As seen on figure 4, points only converge into a single point, when the distance from the object to the lens, or from the lens to the image plane is adjusted correctly. This can be observed with point R, which is further away from the lens than point P and therefore results in multiple points on the image plane rather than one, which will look blurred or out of focus on the resulting image. Consequently, lenses have a specific distance at which objects are focused (Hata & Savarese, 2022).

4.2.1 Thin lens model

To mathematically describe the lens shown in figure 4, the thin lens model can be used. A thin lens is defined as having a thickness that is much less than the radii of curvature of the lens surfaces, therefore $R_1 \gg d$ and $R_2 \gg d$ on figure 5. Due to the small thickness, optical effects can be ignored which simplifies ray tracing calculations (Ray, 2002, p. 45).

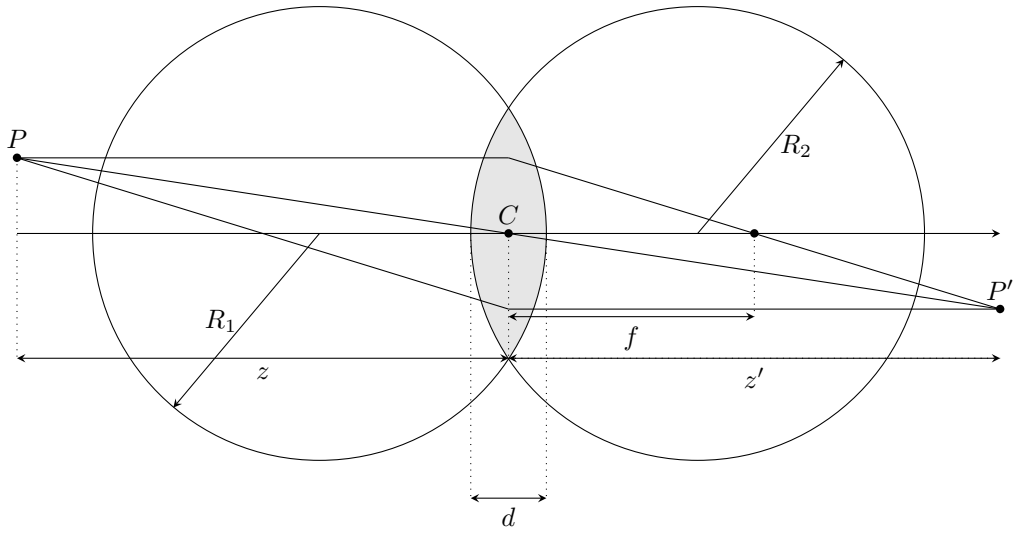


Figure 5: Thin lens model

Figure 5 visualises the theoretical ray tracing when using a thin lens. The light rays of point P are redirected by the lens and converge into a single point P'. The rays passing through a thin lens follow three specific rules:

- A ray passing through the centre C of the lens does not change its direction
- A ray that passes through the lens in parallel to the axis on one side passes through the focal point f on the other side
- A ray that passes through the focal point f on one side will be parallel to the axis on the other side

(Fields, 2020)

When using a lens, in contrast to the pinhole camera, the focal length f is the distance between the lens and the point where all rays that were parallel to the axis before entering the lens meet. It can be calculated using the thin lens equation, being:

$$\frac{1}{|z'|} + \frac{1}{|z|} = \frac{1}{f} \tag{13}$$

where z is the horizontal distance between the point P and the camera centre and z' is the horizontal distance between the point P' and the camera centre (Ray, 2002, p. 36). All in all, the pinhole camera model can still be used to relate a point in 3D space with its corresponding point on the image plane, therefore equation (11) can also be used with modern cameras. Nevertheless, it is important to notice, that the focal length $f = z'$ when using the before mentioned equation (Hata & Savarese, 2022).

4.2.2 Circle of confusion

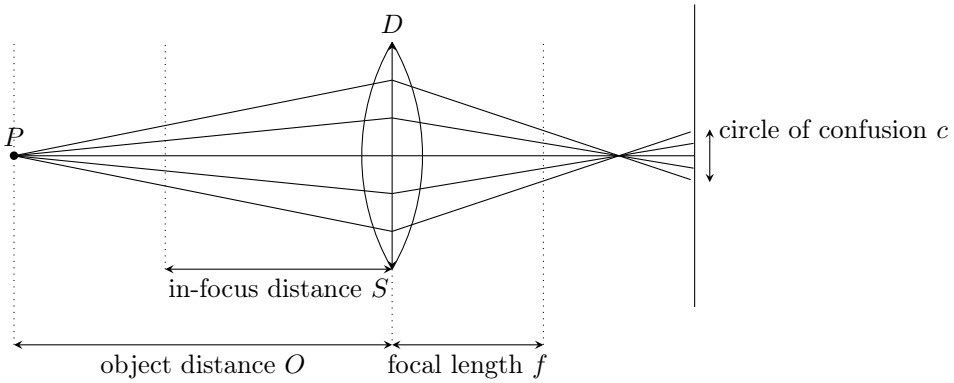


Figure 6: Circle of confusion

As seen on figure 6, if a point P is not focused, it does not converge into a single point on the image plane. Consequently, a circle rather than a point is formed. The resulting circle is called the circle of confusion. As the point should also be a point on the sensor, the sharpness or how much a point is focused can be expressed using the size of the circle of confusion. Therefore, if something is perfectly focused, the diameter of the circle of confusion would be 0. In the real world, a lens can never focus all rays perfectly, therefore the diameter of the circle of confusion will only approach 0. The best-case scenario produces the circle of least confusion which has the smallest diameter possible for a given lens (Ray, 2002, p. 216). The diameter of the circle of confusion for a given focal length f, object distance O and the aperture size D can be calculated using:

$$c = D\frac{|O - S|}{O}\frac{f}{S - f} \tag{14}$$

where S is the distance, at which the point P would be completely focused on the image plane (Ray, 2002, p. 2017).

4.2.3 Depth of field

The depth of field is defined as being the distance between the nearest and farthest object which is in focus on a given lens. As mentioned in section 4.2.2, the circle of confusion can be used to measure the amount a given point is in focus. Therefore, we can define if an object is in focus, by specifying a maximum circle size. Consequently, the depth of field is the distance between the nearest and farthest object with the maximum circle size (Durand & Freeman, 2006). As a pinhole camera has every object in focus, its depth of field is nearly infinite. To calculate the depth of field, the object distance O of figure 6 needs to be calculated for both the nearest and

farthest object:

$$F = \frac{fSD}{fD - c(S - f)} \quad N = \frac{fSD}{fD + c(S - f} \tag{15}$$

where c is the maximum size of the circle of confusion which corresponds to an acceptably sharp focus. As we now have the nearest and farthest object that is still focused, calculating the depth of field is only a matter of subtracting the nearest from the farthest object's distance (Ray, 2002, p. 218).

As seen on equation (14), the diameter of the circle of confusion is directly proportional to the aperture size. As the aperture size decreases, the circle of confusion size also decreases, therefore the depth of field increases (Hata & Savarese, 2022).

4.2.4 Field of View

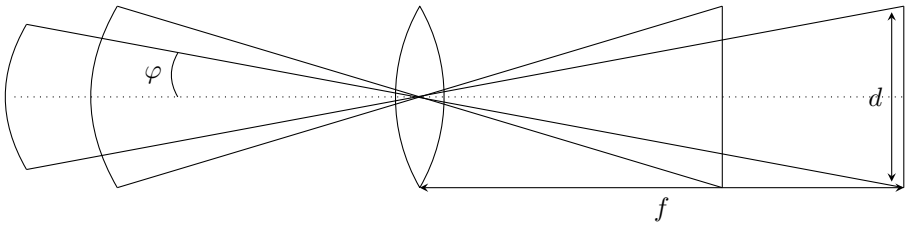


Figure 7: Field of view

The field of view of a camera describes the extent of the world that is being captured by a given lens. It is defined as being the angle between the axis that intersects with the camera centre and the most outer ray that can still be captured by the sensor (Ray, 2002, p. 136). The focal length f and the sensor size d on figure 7 can be used to calculate the field of view φ with:

$$\varphi = 2 \arctan \frac{d/2}{f} \tag{16}$$

As seen on figure 7, the focal length directly influences the field of view. When the lens is further away from the image plane, therefore increasing the focal length, then the field of view decreases. In exchange, the captured scene is zoomed in, as it is still covering up the whole image plane. This is also done in real cameras, as the lens is moved away from the sensor to zoom in (Durand & Freeman, 2006).

4.2.5 Distortions

The thin lens model assumes that the lens is perfectly formed. As this is not true for real world lenses, the resulting image may be distorted. The light rays passing thought the lens have to be more redirected when they enter the lens at its edge, therefore the distortions get stronger at the edges of the lens. In general, the

distortions are noticeable, as straight lines appear curved or deformed on the final image (Ray, 2002, p. 94).

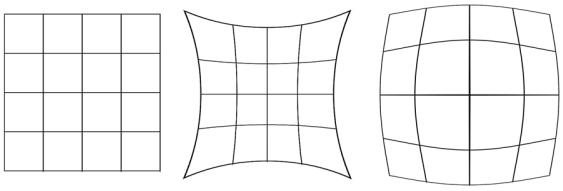


Figure 8: Pincushion and barrel distortions (Hata & Savarese, 2022)

The most common distortion type is called radial distortion. It causes the magnification of the image to increase or decrease. If the magnification increases, then the radial distortion is classified as pincushion distortion, when it decreases as barrel distortion. Radial distortions are caused by lenses that do not have a uniform focal length across its different portions (Hata & Savarese, 2022).

5 Bird's Eye View

As described in the introduction, images cannot be directly used to measure distances in the real world. Due to the way modern cameras capture an image, information is not saved that would be needed to make precise measurements. The system therefore relies on images, that were taken out of a bird's perspective, because proportions are not distorted (Abbas & Zisserman, 2019). The angle relative to the ground plane has to be exactly 90°, additionally, the ground has to be flat to achieve an accurate distance reading. With a known distance in the real world between the points on the image, a uniform conversion rate can be calculated, so distances between all points on the image can be calculated.

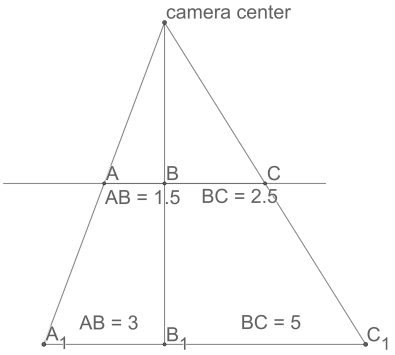


Figure 9: Pinhole camera model with a flat ground plane

As seen on Figure 9, which represents the pinhole camera model where A_1, B_1 and C_1 are mapped to the points A, B and C on the image plane, the conversion rate is uniform for AB and BC. The conversion rate in this case being .5, as $3 * 0.5 = 1.5$ and $5 * 0.5 = 2.5$.

The distances of AB and BC on figure 9 can also be calculated using equation (4) with $z = 4$ and $f = 2$:

$$AB = \frac{2 * 3}{4} = 1.5 \ \ \& \ \ BC = \frac{2 * 5}{4} = 2.5 \tag{17}$$

The conversion rate can therefore be derived from equation (17) being the ratio of the focal length to the distance of the object to the camera (f/z).

6 Homography

As the images used as inputs were taken by CCTV cameras, it cannot be assumed that every image was taken in the bird's eye perspective. The images therefore have to be transformed from the perspective the images were taken in, into the bird's eye perspective. This is done with the concept of homography. Homography describes the relation between two images of the same surface taken by a pinhole camera. This is done by setting the z-value of the 3D coordinates to 0, therefore all points have to be on the same plane (Collins, 2007).

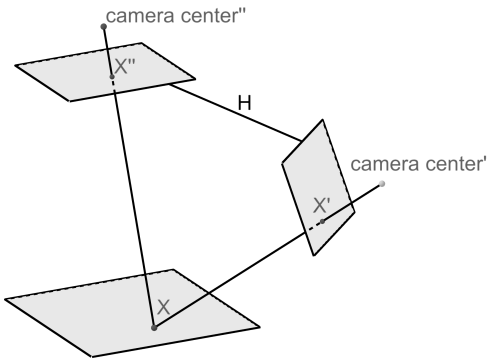


Figure 10: Planar homography

Figure 10 shows two image planes and the point X being mapped on them. Point X' is a point on an existing image that was taken by a camera and point X'' is mapped on an imaginary image plane by using point X' and the corresponding homography (Roth, 2010). This is done using

$$X'' = MX' \tag{18}$$

where M is a 3x3 matrix called homography matrix. This matrix relates the pixel coordinates of the two images (Roth, 2010). The transformed image should be in the bird's eye view, therefore rotation matrices have to be calculated to compensate for the individual angles of the axis. Consequently, the following procedure is followed:

Remove camera roll

As the camera has to be pointed straight down from above the scene, the roll of the camera has to be removed. To make accurate measurements, the z-axis of the camera has to be aligned with the z-axis of the world. Therefore, the angle α is equal to the camera's roll relative to the ground plane.

$$R_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

Remove camera pitch

Similar to the roll, the camera's tilt has to be adjusted to create an image in the bird's eye perspective. However, in this case, the camera's x-axis has to be rotated to align the view of the camera perpendicular to the ground plane. This is done by adding $\pi/2$ to the camera's tilt relative to the ground plane, therefore $\beta = tilt + \frac{\pi}{2}$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \tag{20}$$

Composition

As the rotation around the y-axis only rotates the ground plane, the distances between the objects do not change, therefore no rotation matrix for the y-axis is needed. The homography matrix can now be calculated using:

$$M = KR_zR_xK^{-1} \tag{21}$$

where K is the camera calibration matrix, defined in equation (7) (Roth, 2010). An image can now be transformed from its original perspective to a bird's eye perspective by calculating the position of each pixel on the new image with equation (18).

6.1 Homography applied to an image

Figure 11: Homography applied to an image

As seen on figure 11, the parking lot looked wider in the front and narrower in the back before the transform. By applying a perspective transform on the image, it is converted into the bird's eye perspective and the dimensions of the parking lot are corrected (Abbas & Zisserman, 2019). Nevertheless, as objects or humans may be taller than the ground plane, which differs from the assumption that every point lies on the same plane, these objects will be obscured on the final image. As the goal is to calculate distances between objects, more precisely points, distorted objects do not matter (Collins, 2007).

7 Camera calibration

In the previous section the method of transforming an image into a bird's eye perspective was explained. As seen in equation (21), to perform the transform, the internal camera parameters and the rotation of the camera relative to the ground plane are needed. The parameters could be manually entered into the system, but numerous efforts have been made for estimating those parameters called camera calibration. Camera calibration is an important preprocessing step not only for distance measurement but also for object detection (He, Gkioxari, Dollár, & Girshick, 2018; Ge, Liu, Wang, Li, & Sun, 2021) and crowd counting (Liu, Lis, Salzmann, & Fua, 2019) as perspective distortions also effect the crowd density. The camera calibration procedure can be categorised in two main groups:

7.1 Manual calibration procedure

The most common manual camera calibration procedure is done by using a chessboard or another well-defined pattern. When taking at least 10 pictures with a static camera position and the calibration object at a set distance to the camera, these images can be used to calculate the internal and external camera parameters. This is done by first analysing the images to find the individual edges on the chessboard called image coordinates, then a point is used as the origin for defining the object coordinates. For example, the point on the right of the origin would have the coordinates $(1, 0)$ if measured in square units, or $(10, 0)$ if each square is 10mm wide. These coordinates have a z-value of 0, as the distance to the camera was kept stationary when the images were shot. The image and object coordinates are used in an algorithm by Zhengyou Zhang (Zhang, 2000), which then estimates the internal and external camera parameters. As this method utilises multiple images and a reference object, the result is very accurate, but it requires a lot of time to calibrate multiple cameras.

7.2 Automatic calibration procedure

As the manual calibration procedure is not always viable, multiple methods for an automatic camera calibration procedure were introduced. The vast majority of them use deep learning to predict the required camera parameters. As most methods do not predict both internal and external camera parameters, they can be split into two groups:

7.2.1 Prediction of internal camera parameters

As defined in equation (7), to estimate the internal camera parameters only the focal length f is needed, because the camera centre offset can be calculated using the input image. Furthermore, as mentioned in equation (16), the focal length is directly related to the field of view and the image size, so a direct prediction of the focal length would not be practical. Numerous methods therefore predict the field of view and then use the image size to calculate the focal length (Hold-Geoffroy et al., 2018; Workman, Greenwell, Zhai, Baltenberger, & Jacobs, 2015) . In some cases the vertical vanishing point is predicted as it is also directly related to the focal length (Abbas & Zisserman, 2019). To predict the field of view a convolutional neural network (CNN) is utilised as they perform best on images (Simon, Fond, & Berger, 2018; Workman, Zhai, & Jacobs, 2016). Newer research has shown, that a CNN in combination with a transformer (Vaswani et al., 2017) performs even better on this task (Lee et al., 2021). Nevertheless, all methods have in common, that they predict the field of view by spitting the range of possible angles, in this case $[0.2, 1.8]$ radians, into uniform 256 steps, also called bins. By using the softmax activation function, the CNN outputs a probability for each angle (Hold-Geoffroy et al., 2018).

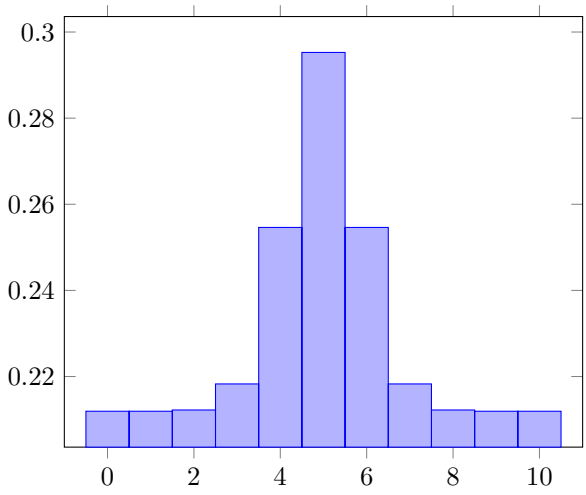


Figure 12: Example output of a softmax function

The bin with the highest probability is the output of the neural network, in the case of figure 12 the output would be 5. The predicted field of view can then be used to calculate the focal length using equation (16). This approach is called regression-by-classification. Regression refers to predicting a numeric value and classification only predicts the probability for a class. As the final output has to be an angle the problem encountered would need a regression, but by using bins that correspond with values, it can be solved by using classification (Abbas & Zisserman, 2019).

7.2.2 Prediction of external camera parameters

In contrast to the focal length, the pitch and roll of the camera which are needed in equation (19) and equation (20) to compensate the perspective distortion, are predicted directly (Zhu et al., 2021). The roll and the pitch are also predicted by using bins, however, the distribution is not uniform. To make finer estimations around 0, the bins are smaller around this value. The width of the bins follow an inverse normal distribution with $\mu = 0$ and $\sigma = 0.5$ (Hold-Geoffroy et al., 2018), the network therefore predicts the probability of the angle being the pitch or roll of the input image. Finally, the angle with the highest probability can be used to calculate the homography matrix (Hold-Geoffroy et al., 2018).

8 Multi-headed neural networks

As described in section 7.2, multiple convolutional neural networks would be needed to predict the pitch, roll and the focal length. The neural networks would all have the same layer architecture, it would therefore do similar operations on the same image multiple times. This approach needs considerable amounts of computational power and time to predict all wanted parameters, therefore, multi-headed neural networks were introduced.

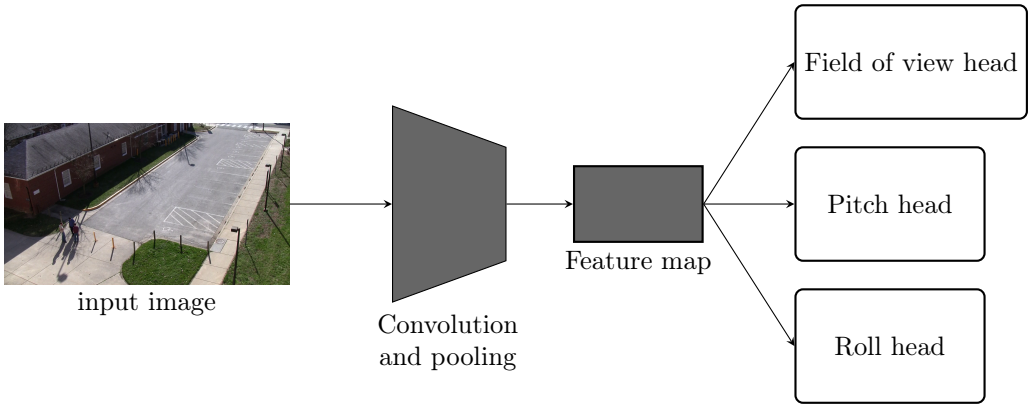


Figure 13: Multi-headed neural network

As seen on Figure 13 the input image is passed to the convolution and pooling layers (O'Shea & Nash, 2015). The convolution and pooling layers can also be stacked multiple times behind each other (Krizhevsky, Sutskever, & Hinton, 2012). The final output of those layers are called a feature map. In a common CNN, those layers are followed by multiple fully connected layers which then predict the final result. When utilising a multi-headed network, the network is split into a backbone, in this case, the convolution and pooling layers and multiple heads, in this case, the field of view, pitch and roll head. The heads consist of fully connected layers, which output the final prediction. The network therefore starts of by predicting the feature map and then uses it to estimate the final three camera parameters. As the features are only extracted once by the neural network, it is more efficient than multiple convolutional neural networks.

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \log \frac{p(x_i)}{q(x_i)} \tag{22}$$

To train the model, the Kullback-Leibler divergence (22) of the three heads are summed up and used as loss (Hold-Geoffroy et al., 2018).

9 Conclusion

To accurately measure distances only using camera images, a predefined calibration procedure has to be executed. Therefore, errors can be introduced into the system by either the real-world reference distance or the internal and external camera parameter prediction. Nevertheless, the final measurements are still very accurate. Additionally, as the calibration procedure has to be executed only once per camera, the measurement process itself does not need significant computational resources. A lot of work has been done in the different fields necessary for this approach to work. The automatic approach to camera calibration by using neural networks has only emerged in the last decade but already shows very accurate results (Hold-Geoffroy et al., 2018).

In conclusion, a lot of needed parameters can be extracted from a single image, but a reference distance is still needed, as distances are hard to predict reliably for different scenes (Zhu et al., 2021).

References

Abbas, S. A., & Zisserman, A. (2019). A geometric approach to obtain a birds eye view from an image. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Retrieved from https://arxiv.org/abs/1905.02231 doi: 10.1109/iccvw.2019.00504

Collins, R. (2007). *Planar homographies.* http://www.cse.psu.edu/~rtc12/CSE486/lecture16.pdf. (Accessed: 2022-02-19)

Durand, F., & Freeman, B. (2006). *Focus and depth of field.* https://groups.csail.mit.edu/graphics/classes/CompPhoto06/html/lecturenotes/22_DepthDefocus.pdf. (Accessed: 2022-02-19)

Fields, D. (2020). *Geometric optics ii thin lenses.* http://physics.unm.edu/Courses/Fields/Phys2310/Lectures/lecture18.pdf. (Accessed: 2022-02-19)

Fossum, E. R., & Hondongwa, D. B. (2014). A review of the pinned photodiode for ccd and cmos image sensors. *IEEE Journal of the Electron Devices Society*, *2*(3), 33-43. doi: 10.1109/JEDS.2014.2306412

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (Second ed.). Cambridge University Press, ISBN: 0521540518.

Hata, K., & Savarese, S. (2022). *Cs231a course notes 1: Camera models.* https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf. (Accessed: 2022-02-02)

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). *Mask r-cnn.*

Hecht, E. (2010). Hecht's response. *The Physics Teacher*, *48*(1), 5–6. Retrieved from https://doi.org/10.1119/1.3274347 doi: 10.1119/1.3274347

Hold-Geoffroy, Y., Sunkavalli, K., Eisenmann, J., Fisher, M., Gambaretto, E., Hadap, S., & Lalonde, J.-F. (2018). *A perceptual measure for deep single image camera calibration.*

Jia, Y.-B. (2020). *Com s 477/577 homogeneous coordinates.* https://faculty.sites.iastate.edu/jia/files/inline-files/homogeneous-coords.pdf. (Accessed: 2022-02-14)

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012, 01). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, *25*. doi: 10.1145/3065386

Lee, J., Go, H., Lee, H., Cho, S., Sung, M., & Kim, J. (2021). *Ctrl-c: Camera calibration transformer with line-classification.*

Liu, W., Lis, K., Salzmann, M., & Fua, P. (2019). *Geometric and physical constraints for drone-based head plane crowd density estimation.*

O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks.*

Ray, S. (2002). *Applied photographic optics: Lenses and optical systems for photography, film, video, electronic and digital imaging.* Focal. Retrieved from https://books.google.at/books?id=cuzYl4hx-B8C

Roth, D. G. (2010). *Homography.* http://people.scs.carleton.ca/~c_shu/Courses/comp4900d/notes/homography.pdf. (Accessed: 2022-02-19)

Simon, G., Fond, A., & Berger, M.-O. (2018, September). A-contrario horizon-first vanishing point detection using second-order grouping laws. In *Proceedings of the european conference on computer vision (eccv)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need.*

Workman, S., Greenwell, C., Zhai, M., Baltenberger, R., & Jacobs, N. (2015). Deepfocal: A method for direct focal length estimation. , 1369–1373. doi: 10.1109/ICIP.2015.7351024

Workman, S., Zhai, M., & Jacobs, N. (2016). *Horizon lines in the wild.*

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334. doi: 10.1109/34.888718

Zhu, R., Yang, X., Hold-Geoffroy, Y., Perazzi, F., Eisenmann, J., Sunkavalli, K., & Chandraker, M. (2021). *Single view metrology in the wild.*

List of Figures

DOCUMENTATION

Mask Detection and Distance Measurement Software to protect against COVID19

Verlauf

23.09.2021 um 12:03 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler, Moritz Patek, Tobias Pressler) wurde eingereicht.

23.09.2021 um 12:07 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler) wurde vom Betreuer / von der Betreuerin akzeptiert.

27.09.2021 um 10:59 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler, Moritz Patek, Tobias Pressler) wurde vom zuständigen Abteilungsvorstand abgelehnt.

27.09.2021 um 14:45 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler, Moritz Patek, Tobias Pressler) wurde eingereicht.

28.09.2021 um 08:25 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler) wurde vom Betreuer / von der Betreuerin akzeptiert.

29.09.2021 um 08:27 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler) wurde vom zuständigen Abteilungsvorstand akzeptiert.

30.09.2021 um 16:50 Die Themenstellung "Mask Detection and Distance Measurement Software to protect against COVID19" (Lukas Gäbler) wurde vom Direktor / von der Direktorin genehmigt.

Schule

Höhere Bundeslehr- und Versuchsanstalt für Textilindustrie WIEN (5)

Abteilung(en)

Hauptverantwortlich: Informatik - Tag

AV

Hauptverantwortlich: Robert Jelinek

Abschließende Prüfung

2022

Betreuer/innen

Hauptverantwortlich: Harald Zumpf

Ausgangslage

Various scientific publications have proven that the distance between two subjects is a major indication and severe reason for the spread of the COVID-19 virus. Additionally, masks, FFP2- as well as face-masks, are measurements to minimize the risk of spreading the COVID-19 virus and therefore prevent a collapse of the health system in Austria.

Projektteam (Arbeitsaufwand)

Name	Individuelle Themenstellung	Klasse	Arbeitsaufwand
Lukas Gäbler (Hauptverantwortlich)	Implementing and creating algorithms for property detection of visual data	5AHIF	160 Stunden
Moritz Patek	Backend and system architecture for storage procedures of visual data	5AHIF	160 Stunden
Tobias Pressler	Parameter and calibration procedures and HID for accurate distance measurement	5AHIF	160 Stunden

Projektpartner

AIT (E-Mail: Bernhard.Haslhofer@ait.ac.at)
Giefinggasse 4, 1210 Wien, Bernhard Haslhofer

Untersuchungsanliegen der individuellen Themenstellungen

The goal is to investigate the possibility of accurately measuring the distance between objects using computer vision based on AI.

Lukas Gäbler: Implementing and creating algorithms for property detection of visual data

Tobias Pressler: Parameter and calibration procedures and HID for accurate distance measurement

Moritz Patek: Backend and system architecture for storage procedures of visual data

Zielsetzung

A system that analyzes the compliance of social distancing and mask
wearing by utilizing existing CCTV cameras. The resulting data can be viewed by the end user and is visualized in meaningful graphs.

Geplantes Ergebnis der Prüfungskandidatin/des Prüfungskandidaten

Lukas Gäbler: Implementing and creating algorithms for property detection of visual data

Tobias Pressler: Parameter and calibration procedures and HID for accurate distance measurement

Moritz Patek: Backend and system architecture for storage procedures of visual data

Meilensteine

01.10.2021 Pre study completed

07.01.2022 Prototype completed

01.02.2022 Feature complete, start, testing and refinement

26.03.2022 Rollout and acceptance test

Rechtliche Regelung (mit dem/den Projektpartner/n erfolgt durch)

Kooperationsvereinbarung

Dokumente

Eidesstattliche_Erklärung_Diplomprojekt.pdf

Kooperationsvertrag_Diplomprojekt.pdf

Änderungen im Kooperationsvertrag mit Projektpartner AIT und Betreuer Prof. Zumpf abgestimmt.

KOOPERATIONSVEREINBARUNG

zwischen

AIT Austrian Institute of Technology GmbH

(in der Folge **„der Projektpartner, die Projektpartnerin"**)

und

Lukas Gäbler
Geb. Dat.: 07.04.2003
Neurissener Anger 23, 1220 Wien

Moritz Patek
Geb. Dat.: 16.05.2003
Maria-Theresia-Park 8, 7111 Parndorf

Tobias Pressler
Geb. Dat.: 10.04.2003
Rosenbüchelgasse 40, 2500 Baden

(in der Folge **„das Projektteam")**

PRÄAMBEL

Das Projektteam und der Projektpartner/die Projektpartnerin beabsichtigen gemäß der Verordnung des Bundesministers für Bildung, Wissenschaft und Forschung über die abschließenden Prüfungen in den berufsbildenden mittleren und höheren Schulen sowie in den höheren Anstalten der Lehrerbildung und der Erzieherbildung (Prüfungsordnung BMHS und Bildungsanstalten), BGBl II Nr. 177/2012 i.d.g.F., die Planung und Durchführung eines Diplomprojektes, welches die Erstellung von Arbeitsergebnissen, für das in §1 genannte Projekt, zum Gegenstand hat.
Durch die Zusammenarbeit soll insbesondere den Mitgliedern des Projektteams die Möglichkeit eingeräumt werden, im Rahmen ihrer schulischen Ausbildung bei der Durchführung eines Diplomprojektes an die Verhältnisse im technischen Berufsleben herangeführt zu werden, um dabei die in der Schule erworbenen theoretischen Kenntnisse und Fähigkeiten in der Praxis anzuwenden bzw. zu erweitern. Hingewiesen wird in diesem Zusammenhang auf den unentgeltlichen Charakter dieser Vereinbarung.

§1
Gegenstand

Gegenstand ist die Erstellung von Arbeitsergebnissen zum Thema der Diplomarbeit im Sinne des § 8 Prüfungsordnung BMHS:

Mask detection and distance measurement to protect against COVID-19

Der Projektpartner/die Projektpartnerin wird jedoch darauf hingewiesen, dass es sich um ein Projekt im Zusammenhang mit der schulischen Ausbildung handelt und daher jede Haftung des Projektteams, insbesondere in Hinsicht auf die Unentgeltlichkeit des Vertrages, ausgeschlossen ist.

§2
Laufzeit

Die vorliegende Kooperation tritt am 09.09.2021 in Kraft und wird bis zur erstmaligen Abgabe des schriftlichen Teils der abschließenden Arbeit abgeschlossen.

§ 3
Rechte und Pflichten des Projektteams

Das Projektteam verpflichtet sich, die im Gegenstand genannten Arbeiten sorgfältig und unter möglichster Schonung der Interessen des Projektpartners/der Projektpartnerin durzuführen. Das Projektteam verpflichtet sich zur Geheimhaltung aller ihm zur Kenntnis gelangenden Geschäfts- und Betriebsgeheimnisse gegenüber Dritten, ausgenommen Lehrkräfte und Prüfer, die im Rahmen von SchUG §34 (3) 1. und SchUG §37 (4) die Betreuung des Projektteams innerhalb ihrer Dienstverpflichtung durchführen. Projektfortschritt, End- und Zwischenergebnisse dürfen im Rahmen des Schulunterrichts und bei Schul- und schulbezogenen Veranstaltungen vorgeführt werden.

§4
Rechte und Pflichten des Projektpartners/ der Projektpartnerin

Alle Informationen werden von AIT ohne jedwede Gewährleistung übermittelt; weder die Richtigkeit, Vollständigkeit noch die Nutzung für einen bestimmten Zweck werden gewährleistet oder garantiert.
Sollte das Projektteam im Rahmen dieser Kooperationsvereinbarung ein Werk schaffen, dem Schutz iSd UrhG zukommt, hat der Projektpartner das uneingeschränkte Werknutzungs- und Verwertungsrecht. Das Projektteam verzichtet unwiderruflich auf alle Ansprüche einer Vergütung (der Verzicht gilt auch für Ansprüche aus dem PatG).
Das Projektteam wird AIT jeweils eine Kopie jeglicher schriftlicher Arbeiten und Präsentationen, welche unter diesem Vertrag erstellt wurden, spätestens mit der Einreichung/Präsentation dieser Arbeiten unaufgefordert übermitteln.

§5
Einsicht und Präsentation

Da die Tätigkeit des Projektteams auch Inhalt bzw. Grundlage der an der Schule HTBLuVA Wien V, Spengergasse 20 zu erstellenden Diplomarbeit ist, berechtigt der Projektpartner/die Projektpartnerin die zuständigen Organe des Bundes zur Einsicht und Kontrolle, um die Aufgaben gem. Prüfungsordnung BMHS, SchUG bzw. SchUG BKV zu erfüllen. Das Projektteam ist auch berechtigt, Ergebnisse der Diplomarbeit bei der Präsentation und Diskussion der Diplomarbeit zu präsentieren.

§6
Änderungen

Änderungen dieser Vereinbarung bedürfen der Schriftform. Sollte ein Schüler/ eine Schülerin, der/ die Mitglied des Projektteams ist, während der Laufzeit dieser Vereinbarung aus der HTBLuVA Wien V, Spengergasse 20 ausscheiden (Abmeldung vom Schulbesuch), bleibt die Kooperationsvereinbarung für die verbleibenden Unterzeichner, mit Rücksichtnahme auf eine etwaige Reduktion des Projektumfanges, aufrecht.

Ort, am *Wien, 17.09.2021*
.....
AIT Austrian Institute of Technology GmbH

SIGNATURE INFORMATION	**Signatory**	Ross Clarence King III
	Date/Time-UTC	2021-09-20T17:29:42+02:00
	Verification	Information about the verification of the electronic signature can be found at: https://www.signaturpruefung.gv.at
Note	This document is signed with a qualified electronic signature. According to Art. 25 para. 2 of the Regulation (EU) No 910/2014 of 23. July 2014 ("eIDAS-Regulation") it shall have the equivalent legal effect of a handwritten signature.	

.....
i.V. Dr. Ross King
Head of Competence Unit
Data Science & Artifical Intelligence

Signiert von:	Bernhard Haslhofer
Datum:	23.09.2021 07:37:44

Dieses mit einer qualifizierten elektronischen Signatur versehene Dokument hat gemäß Art. 25 Abs. 2 der Verordnung (EU) Nr 910/2014 vom 23. Juli 2014 ("eIDAS-VO") die gleiche Rechtswirkung wie ein handschriftlich unterschriebenes Dokument.

Dieses Dokument ist digital signiert!

Prüfinformation:
Informationen zur Prüfung der elektronischen Signatur finden Sie unter:
www.handy-signatur.at

www.a-trust.at
A-TRUST einfach sicher

.....
i.A. Dr. Bernhard Haslhofer
Senior Scientist & Thematic Coordinator
Data Science & Artificial Intelligence

Wien, 16.09.21
.....
Ort, am

.....
Lukas Gäbler

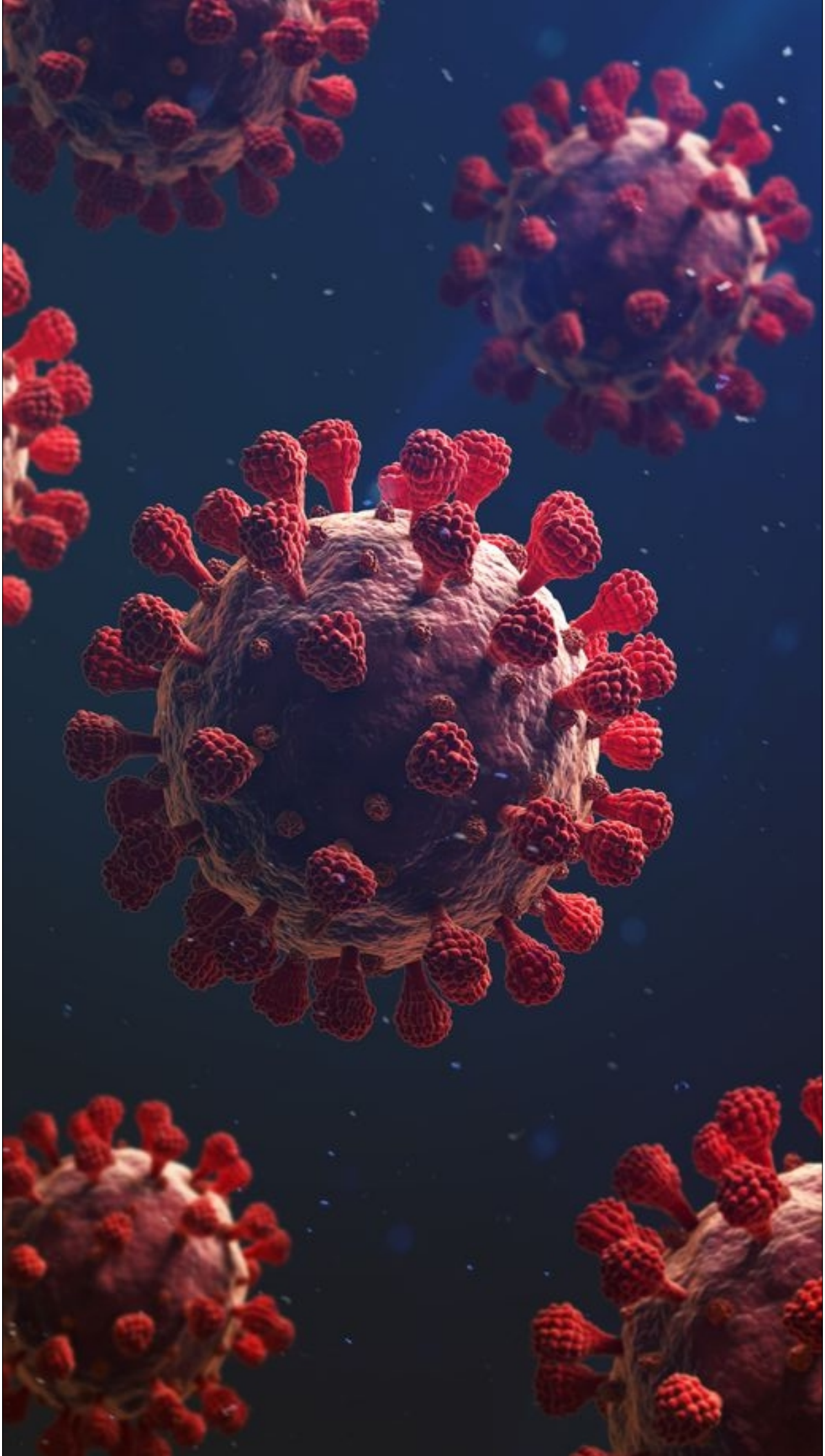
.....
Moritz Patek

.....
Tobias Pressler

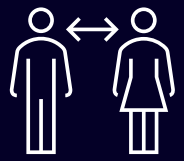
Abstandsmessung mittels künstlicher Intelligenz zur Covid-Prävention

Lukas Gäbler

12.10.2021



Wie kann man sich schützen?

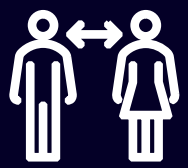


Abstand halten

Masken tragen

Impfen lassen

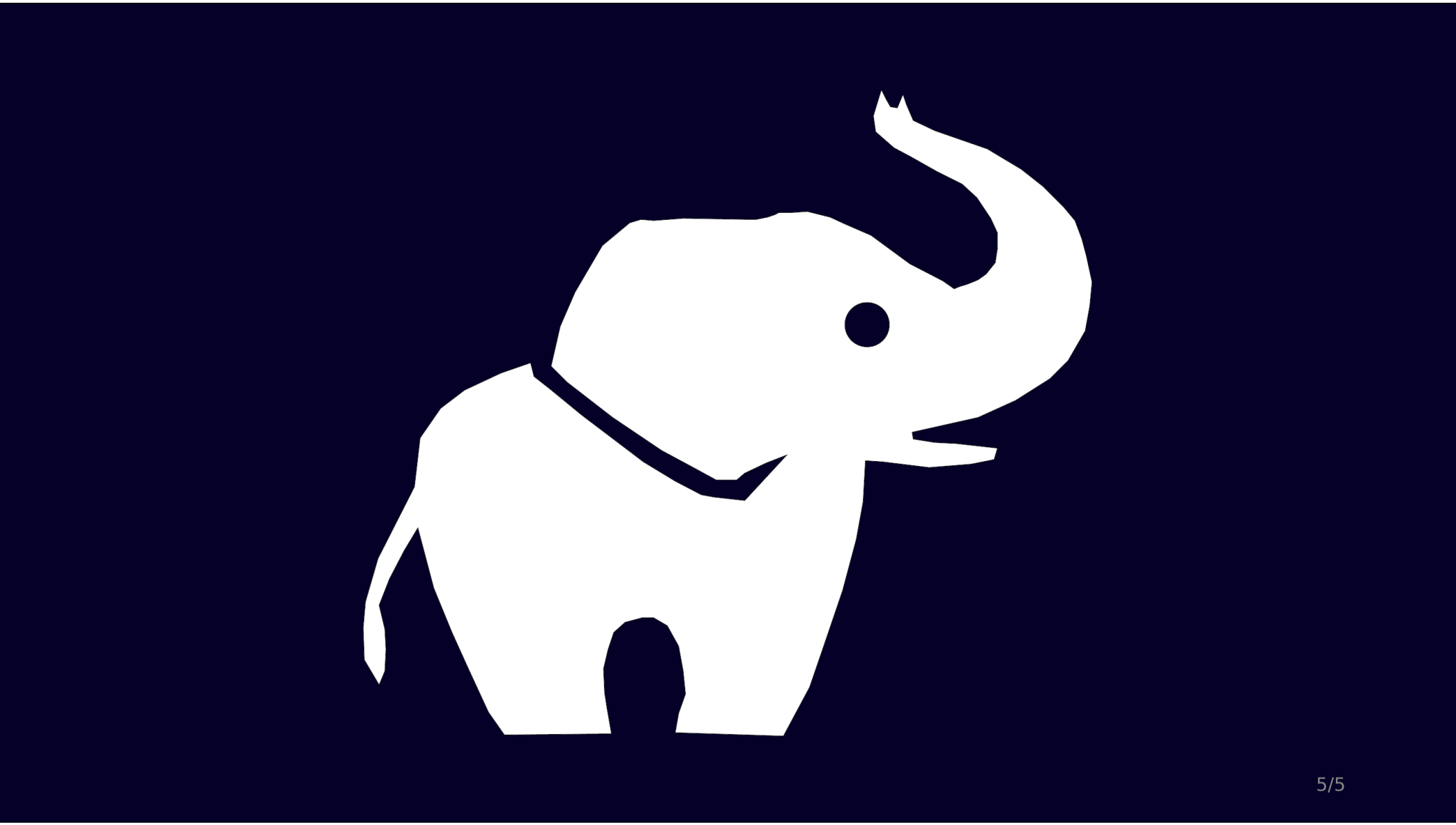
Wie kann man sich schützen?

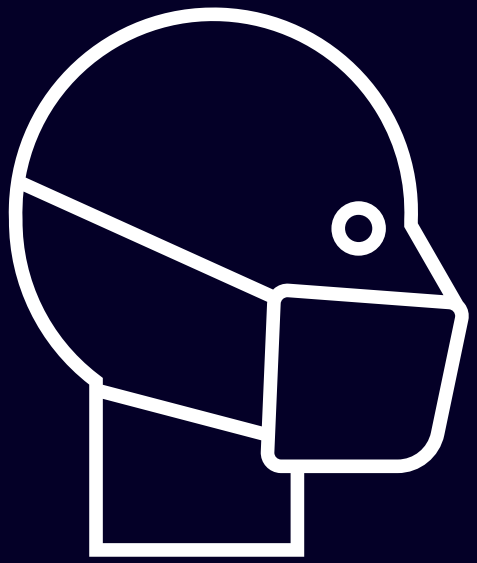


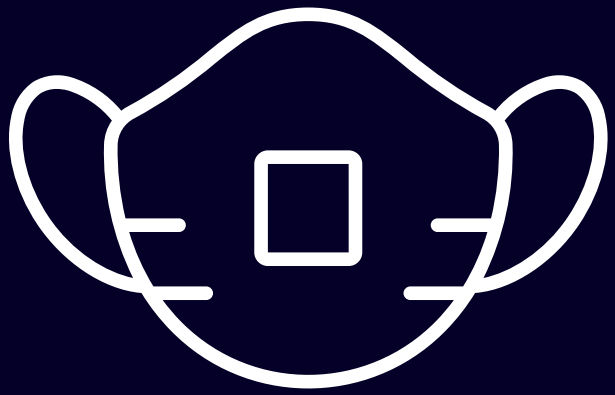
Abstand halten

Masken tragen

Impfen lassen



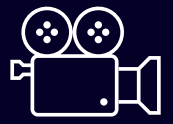




Was macht das Projekt besonders?

Skalierbarkeit

Datenschutzkonform

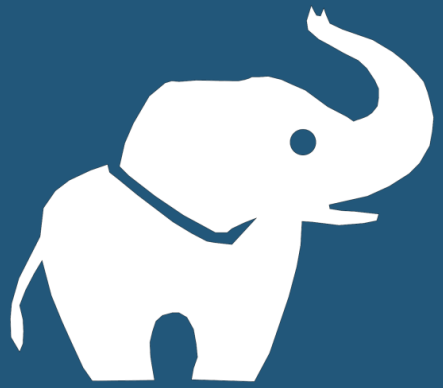


nur 1 Kamera notwendig

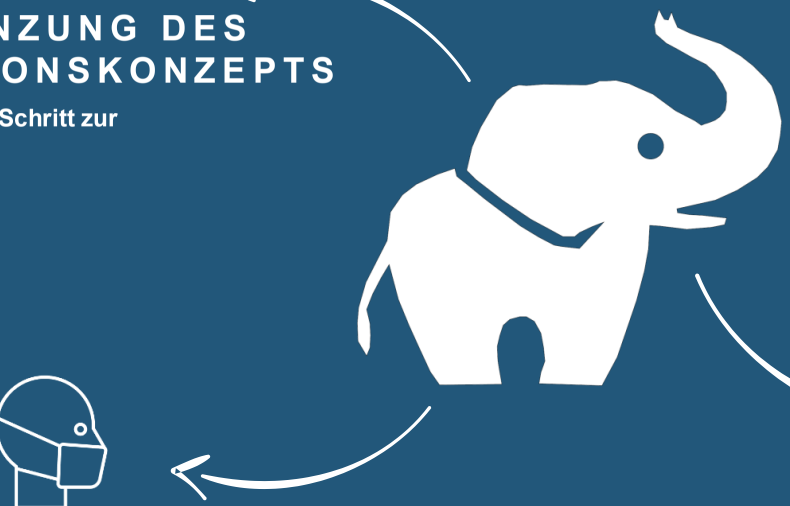

Babyelephant
Home
Dashboard
Events
Zahnaerztekammer-
Burgenland
Aktionen
Abmelden
Eventdashboard
2021-04-12 10:29:33
minimaler Abstand
durchschnittlicher Abstand
2.5
2.0
1.5
1.0
0.5
0

BABYELEFANT

14.03.2022



WAS IST BABYELEFANT?



ERGÄNZUNG DES PRÄVENTIONSKONZEPTS

für einen weiteren Schritt zur Normalität

ABSTANDSMESSUNG

Automatisches Messen und Prüfen des Mindestabstands

MASKENERKENNUNG

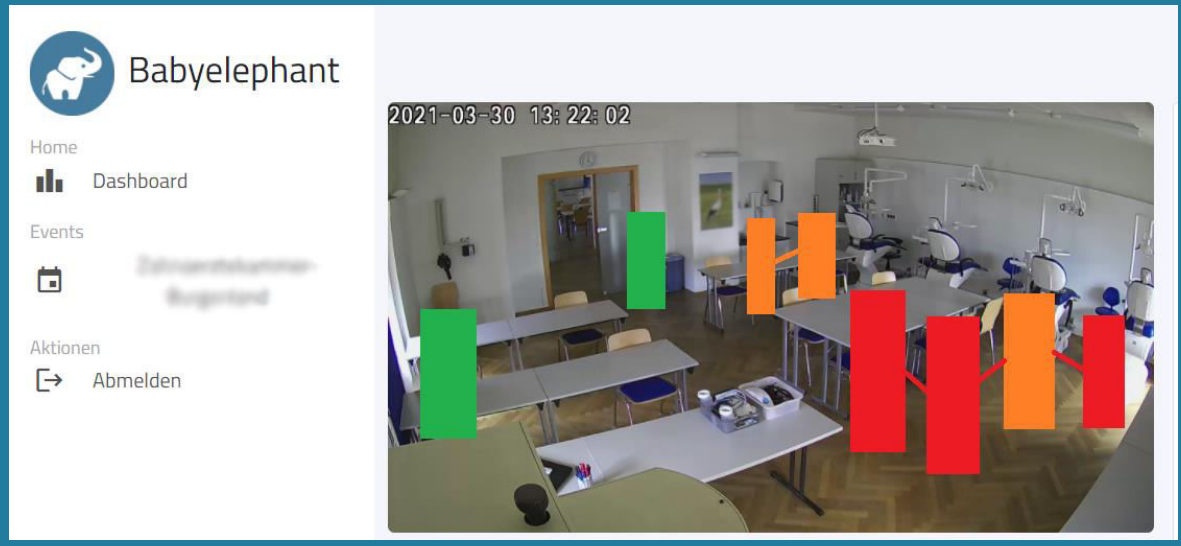
Automatisiertes erkennen der Leute die keine Masken tragen

DATENSCHUTZKONFORM

Es werden keine personenbezogenen Daten gespeichert (DSGVO konform)

LIVESTREAMS

PERSONENERKENNUNG
mittels Künstlicher Intelligenz



ANONYMISIERUNG
durch unkenntlich machen der Personen

ABSTANDSMESSUNG
durch spezielle Algorithmen

VISUALISIERUNGEN

DATEN IN ECHTZEIT

auf der Website als Graphen dargestellt

AUSKUNFT ÜBER EINHALTUNG

der Maskenpflicht sowie im Bezug auf den Mindestabstand

WARUM BABYELEFANT?

MAßNAHMEN ZUSAMMEN EINHALTEN

für weniger Infizierte die auf die Intensivstationen müssen

KÜNSTLICHE INTELLIGENZ

Modernste Technologien um Österreich sicherer zu machen

INFEKTIONSZAHLEN DRÜCKEN

damit ein ausgiebiges Sozialleben sowie mögliche Öffnungen sicherer werden

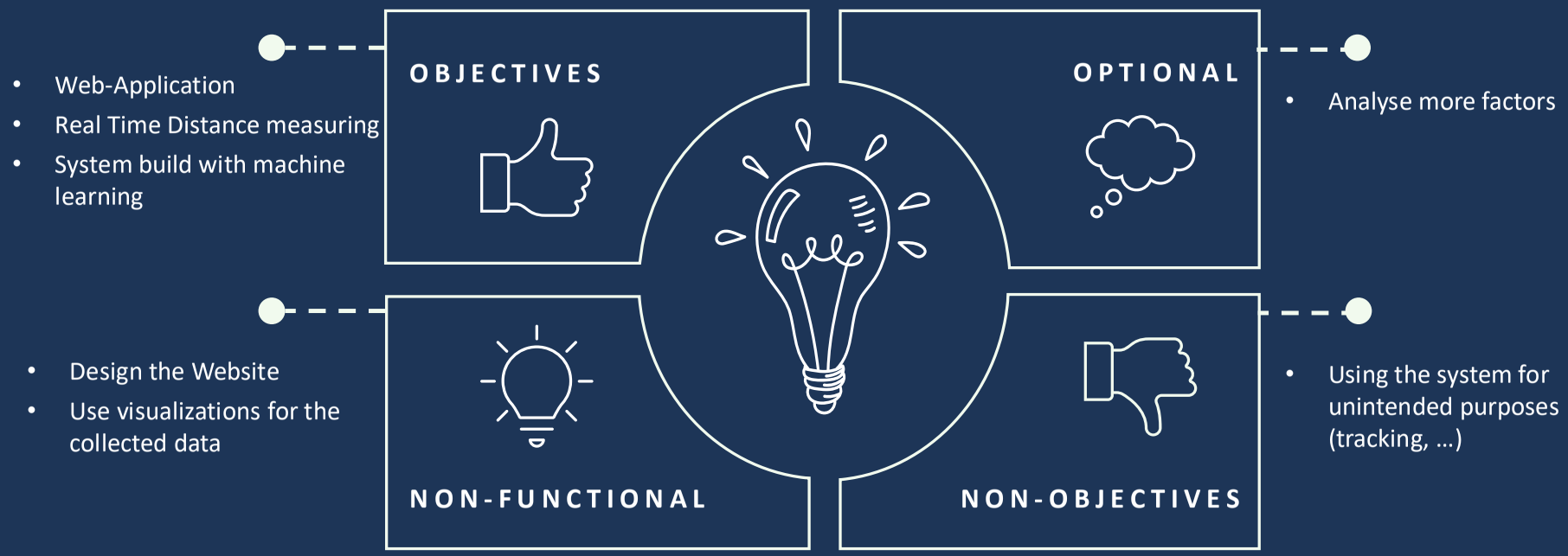
LOKALE MAßNAMEN

setzen, um dort zu handeln wo es notwendig ist und nicht dort wo alle Maßnahmen eingehalten werden

BABYELEFANT

Status Report – 14.02.2022

OBJECTIVES & NON-OBJECTIVES



- Web-Application
- Real Time Distance measuring
- System build with machine learning

- Analyse more factors

- Design the Website
- Use visualizations for the collected data

- Using the system for unintended purposes (tracking, …)

SWOT

STRENGTHS

- WEB BASED
- AI POWERED

WEAKNESSES

- CAMERAS REQUIRED

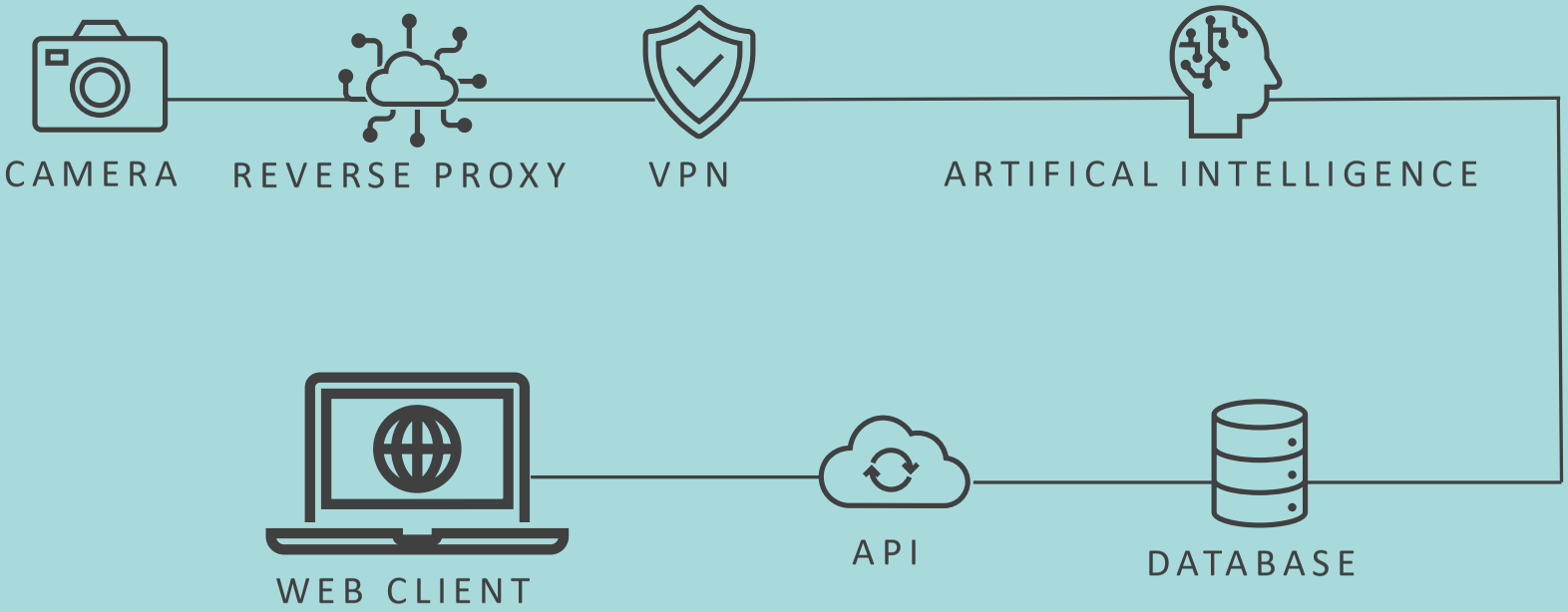
OPPORTUNITIES

- COVID-19
- GROWING MARKET (Computer Vision)

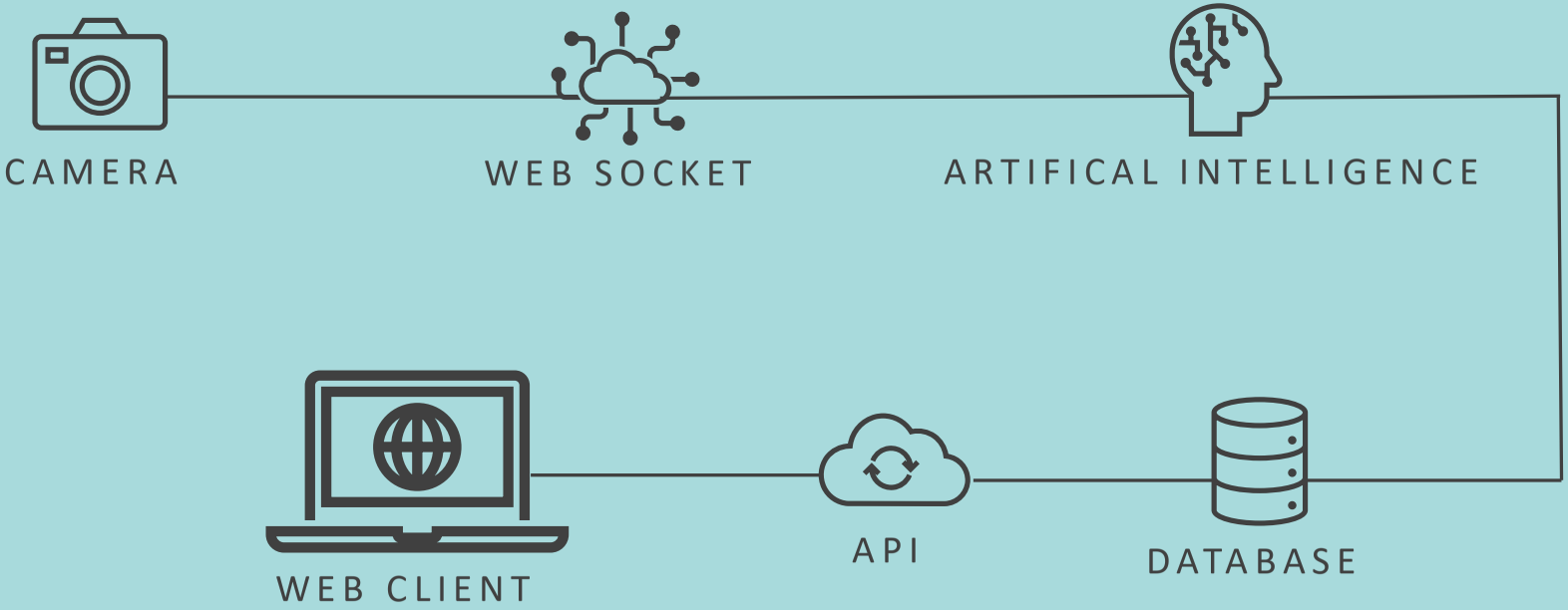
THREATS

- GREATER COMPETITORS

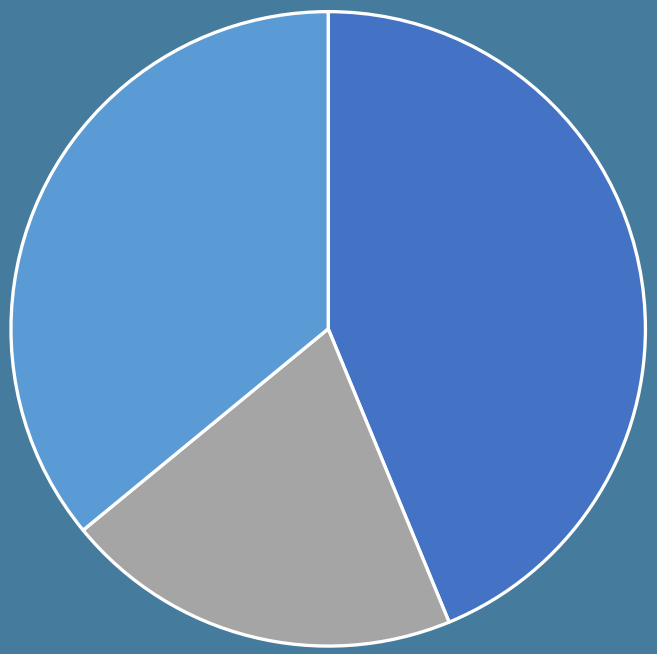
SYSTEM ARCHITECTURE - OLD



SYSTEM ARCHITECTURE - NEW



TIME TRACKING



PROJECT ENVIRONMENT ANALYSIS

POLITICAL

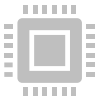
- COVID-19 → great demand for solutions

SOCIAL

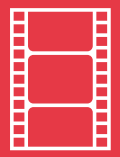
- MANY POTENTIAL USE CASES

TECHNOLOGICAL

- CONSTANT IMPROVEMENTS IN AI

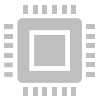




CHALLENGES

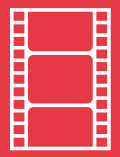


GET A BETTER FRAME RATE

FINDING THE RIGHT PEOPLE

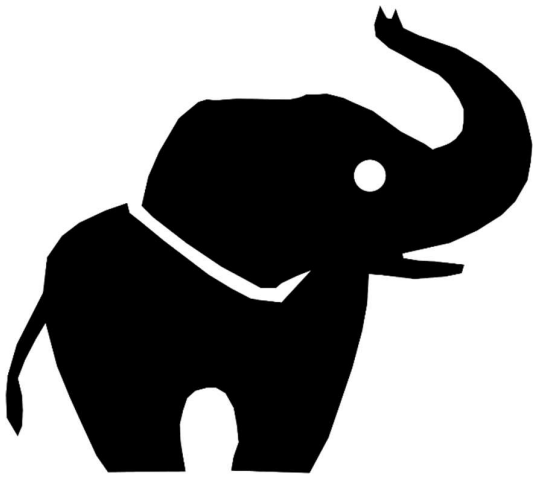




SOLUTIONS



CODE OPTIMIZATION

CONTACT POLITICIANS



Babyelefant

Privacy Policy

Contents

We believe privacy is a fundamental human right that's why it is a very important topic to us. The best way to protect your "personal data" is to not collect it in the first place.

If personal data is processed, this use is in accordance with the regulations of the General Data Protection Regulation (Regulation (EU) 2016/679 of the European parliament and of the council). This privacy policy document contains information on what data is collected and recorded by our website and how we use it. Please read the following to learn more about it.
If you have additional questions or require more information about our privacy policy, do not hesitate to contact us by email at: gae18805@spengergasse.at.

What does this privacy statement cover?

This privacy statement explains how we handle personally identifiable information ("Personal information") that we gather when you access or use Babyelephant.

What Rights Do You Have Regarding Your Personal Data?

You have certain rights with respect to your Personal Data, including those set forth below. For more information about these rights, or to submit a request, please email gae18805@spengergasse.at. Please note that in some circumstances, we may not be able to fully comply with your request, such as if it is extremely impractical, if it jeopardizes the rights of others, or if it is not required by law, but in those circumstances, we will still respond to notify you of such a decision. In some cases, we may also need you to provide us with additional information, which may include Personal Data, if necessary to verify your identity and the nature of your request.

Access: You can request more information about the Personal Data we hold about you and request a copy of this data by contacting us.
Rectification: If you believe that any Personal Data, we are holding about you is incorrect or incomplete, you can request that we correct or supplement that data.
Erasure: You can request that we erase some or all your Personal Data from our system. This may come at the consequence of us having to delete your account and any associated data with it.
Withdrawal of Consent: If we are processing your Personal Data based on your consent, you have the right to withdraw your consent at any time.
Portability: You can ask for a copy of your Personal Data in a machine-readable format. You can also request that we transmit the data to another collector where technically feasible.
Objection: You can contact us to let us know that you object to further use or disclosure of your Personal Data for certain purposes.
Restriction of Processing: You can ask us to restrict further use or disclosure of your Personal Data for certain purposes.

What personal information is collected by Babyelephant?

We do not collect any personal information.

What Security Measures Do We Use?

We are using appropriate technical and organizational measure based on the type of Personal Data and applicable processing activity. For example, we anonymize all data which is being processed during the usage of Babyelephant.

Questions or concerns

If you have any questions or concerns regarding privacy, please send us a detailed message at gae18805@spengergasse.at. We will make every effort to resolve your concerns.

Babyelefant

Contents

Project Management

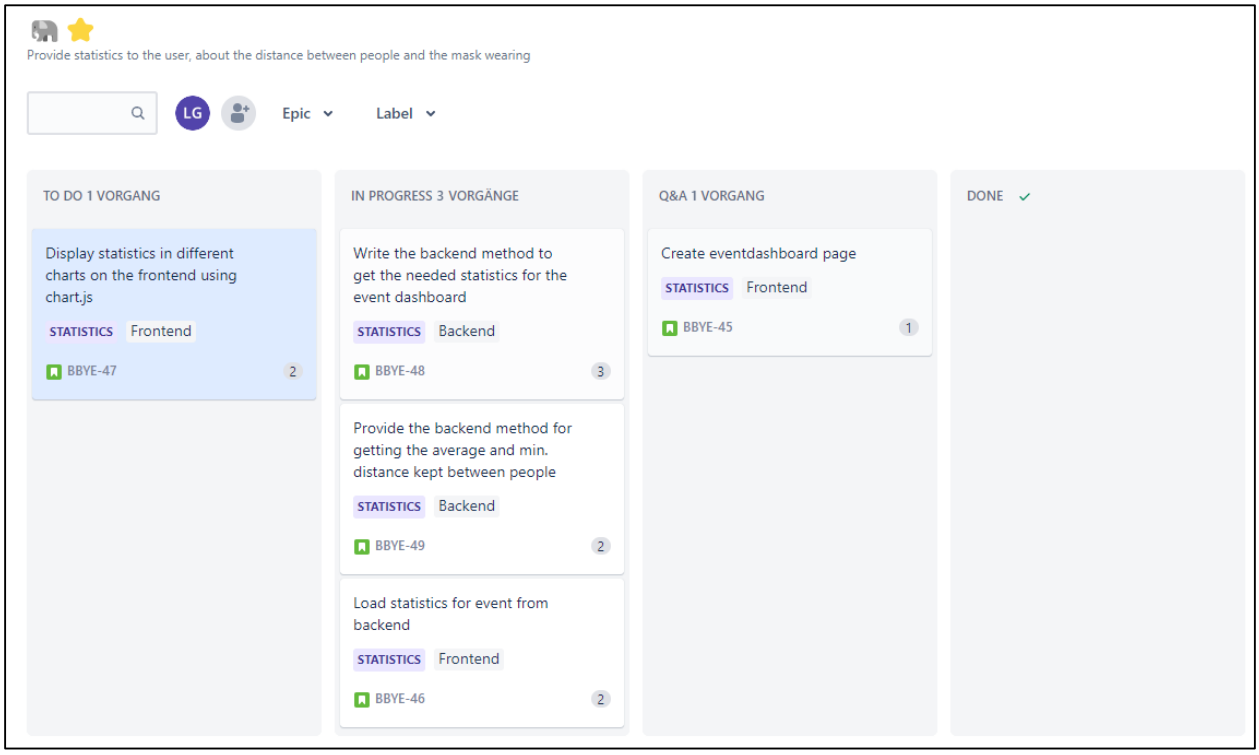
The Babyelefant project is managed using agile methods. In the following text, details regarding management and development practices are given. All of the provided information is available to the team members on the project's OneDrive space.

Definitions

Development tasks are managed using a Kanban board on Atlassian Jira. Work is scheduled into **2-week-long** sprints. At the beginning of each sprint, developers are assigned user stories according to their time budget and responsibilities. The goal of any spring is to finish the assigned user stories and merge them to the master branch after a review. Due to the team's preferences, sprints start and end on Fridays.

The Jira board is customized to contain the following four columns:

- **To Do:** Stories from the backlog selected for the current sprint are placed in the To Do column.
- **In Progress:** As soon as a team member starts working on a story, they are free to move it to the In Progress column.
- **Q&A:** The story is moved to the Q&A column if it is finished or if a pull request has been created. For any tasks in the Q&A column, a peer review session with another team member is scheduled.
- **Done:** If the implementation of the user story meets the Definition of Done it is moved to the Done column.



Definition of Done

The definition of Done is basically just a checklist. During a review, the reviewer and assigned check an implementation for the following requirements:

- Build passing
- Tests implemented
- Code review and approved

A development story may only be moved from **Q&A** to **Done** if all points on the checklist are met.

Regular Meetings

One of the most important rituals when it comes to agile projects are regular meetings. Unless longer times are necessary, meetings are time-boxed at 45 minutes. If a decision cannot be made in this timespan, a team member is given the task to do more research on the topic and a separate meeting is scheduled for another day.

Daily Stand Ups

In order to support a daily collaboration, all team members should participate in daily standup meetings where they quickly describe the work, they finished the day before and the work they are planning on finishing that day. Other team members should actively look for opportunities to help others solving problems if they are stuck during the daily standup.

Because members of the project team are generally busy with school-related tasks in the morning, a channel on Microsoft Teams was created where each team member writes what has been done, what will be done and if you there are any problems or challenges where help is needed.

Once all team members have written their daily messages, everyone of the team has an overview of what is currently happening. Team members are strongly discouraged from not writing their daily messages even if they have some problems where they are stuck.

Review, Retro, Planning

Every second week, one session for sprint review, every other week one session for retro and one session for planning is conducted. All of those meetings are time-boxed

Every second week, a session for sprint review, retro and planning is conducted. Even though these are technically three meetings in one, it is time-boxed at one hour to reduce unnecessary discussions.

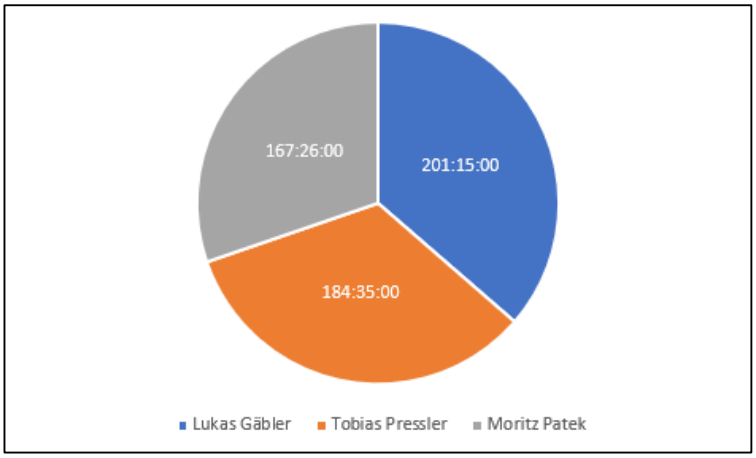
In the review meeting, the stories from the previous sprint are discussed and demonstrated. Team members talk about technological problems faced / solved during the sprint and any

potential questions. Strategic next steps are briefly discussed to reduce the time required in the planning section.

Time Tracking

To ensure accurate and easy time tracking of the time spent on the development of Babyelefant, we used the App “Toggl”. In order to make the best use of this, some general guidelines should be kept in mind:

- **Time should be tracked correctly, and it should be noted what was done:** Always when tracking time it should also be added to that record what was done.
- **Track as you go:** Writing time records as soon as possible after finishing a period of work helps keep the time tracking as accurate as possible.
- **Keep it accurate:** If not certain of how much time was spent on a certain task, time should be estimated as well as possible.
- **Track everything:** Generally seen, every task related to Babyelefant should be tracked. In addition to development, this includes meetings, research and creation of used documentation as well as project-management related documents.



Time tracking records for the project team

Development Practices

The main development method used in Babyelefant should be **Test-Driven Development (TDD)**. The focus lies in the following principles:

- **Code Review:** If you have developed a feature alone, schedule a meeting with another team member to sit down for paired code review. The author explains the code step-by-step to the reviewer, who looks for ways to improve the code. The reviewer may also look at the pull request ahead of time to prepare comments which are then discussed with the author during the meeting.
- **Unit Testing:** Unit testing is a continuous and integral part of the development life cycle. Test-Driven Development means that a test that fails at first is written and then the code to make the test green is implemented.

- **Code Style:** Special care should be taken to enforce the coding practices in Babyelefant in order to create a well-thought out and designed code base. Focus on the following principles:
    - **Don't reinvent the wheel:** If there is a pre-existing solution to a problem, use it.
    - **Low code:** Aim to solve problems in as few lines of code as possible.
    - **Worse is better:** Code with less functionality that is good at what it does is better than code with large functionality that is harder to user / more prone to bugs.
    - **You aren't gonna need it:** Only program features if you need it, not if you think you might need it later.
    - **Release early and often:** During development, if something is done and code-reviewed, merged in to the master branch a pipeline should be triggered to deploy the feature.

Team Members and Roles

The team members on the Babyelefant development team don't have strictly defined roles, but rather fluidly switch between tasks depending on the current state of development and their available time budget.
Below are the three developers on the Team and the tasks they generally assume.

Lukas Gäbler
Project Management
Front-end development

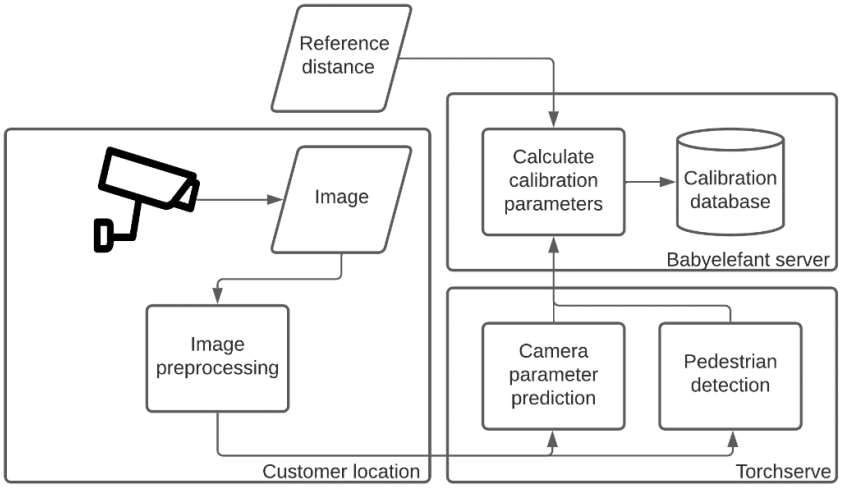
Tobias Pressler
Machine Learning
Infrastructure

Moritz Patek
Back-end development

System Architecture

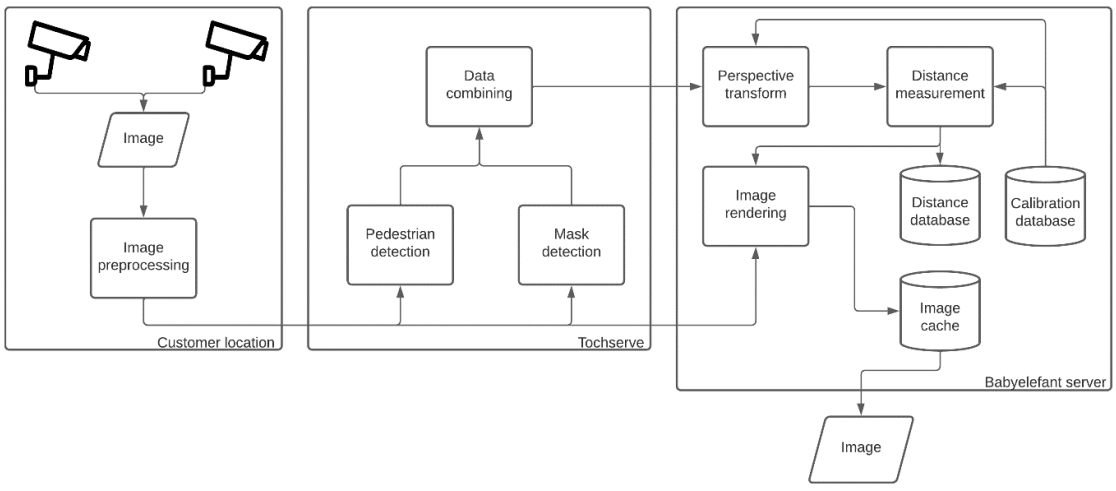
The Babyelefant server consist of two different systems: the calibration and the measurement system. Nevertheless, both operate partially in the cloud and at the customer's location.

Calibration



The system starts of by retrieving an image from the chosen **camera**. The image is then passed to the image **pre-processing server**, which is responsible for converting the file format given by the camera via the RTSP protocol to a JPEG image which is then send to the **Torchserve** server. Upon receiving, the image is distributed to the **Pedestrian detection** and to the **Camera parameter prediction**. The resulting data is used together with a **reference distance** to calculate the calibration parameters which are then stored for later use in the **calibration database**.

Distance measurement



Distance measurement is done by a similar system. As **multiple cameras** may exist at the customer's location, all images are collected and converted into a JPEG image like mentioned in the calibration system. These images are then used to predict the position of **pedestrians** and faces together with a rating whether a **mask** is worn or not. Before transferring the data to the Babyelefant server, those two predictions are combined to a person with or without a mask prediction. The image locations are then **transformed** into the bird's eye perspective with the use of the calibration parameters of the **calibration database**. Now the distances between the pedestrians can be **measured** and converted to real world distances by using additional calibration parameters fetched from the calibration database. For statistical purposes, the distance data is stored in the **distance database**. The data can now be used to draw visual information on the original image which is then stored in the **image cache**. A more detailed description of the specific components is given below.

Docker

As the Babyelefant ecosystem is made up of multiple servers, Docker and docker-compose were used to easily manager, develop and deploy the babyelefant server stack. Docker-compose manages the docker container and injects the necessary environment variables and configurations. The Babyelefant ecosystem consists of:

- **Image client:** The image client is located at the customer's location and is responsible for fetching the images from the cameras, converting the images into the JPEG format and sending them to the server.
- **Tochserve:** Torchserve provides a reliable and extendable server environment for PyTorch based neural networks. It is used to host the three models necessary for the system to work and exposes a REST and gRPC API. Additionally, the models can be scaled automatically based on the limits set in the configuration.
- **Redis:** As described earlier, the rendered images are stored in an image cache. As redis is an in-memory datastore, the image can be saved and retrieved very fast. Additionally, as the images are only temporarily stored for clients to view, it is not necessary to save them to disk, therefore speed can be maintained.
- **PostgreSQL:** To store the calibration parameters and the distance statistics, a PostgreSQL database is used.
- **Babyelefant server:** The Babyelefant server connects all before mentioned components. It manages the database and exposes a REST API for retrieving the analyzed images, manage users and providing the collected statistics to the clients.

Camera parameter prediction

In the calibration procedure, the camera parameters like pitch, roll and focal length are being predicted by a neural network.

Calculate calibration parameters

To calculate the necessary camera parameters, a reference distance, the location of two people and the predicted camera parameters are needed. At first, the homography matrix is computed, as it is needed to transform the two bottom center locations into the bird's eye perspective. By using the reference distance, a real-world to image conversion ratio can be calculated. Both the conversion ratio and the homography matrix are then stored in the calibration database.

Pedestrian and mask detection

The pedestrian and mask detection both use the YOLOv5 architecture implemented with the PyTorch deep learning framework. Pedestrian detection is responsible for predicting the bounding boxes of each person on the image. On the other hand, the mask detection predicts the position of each face and classifies if a mask is worn. Separating those tasks showed more accurate results than directly classifying if a person wears mask.

Data combining

As the final statistics does not need the position of the faces, the output of the pedestrian and mask detection can be merged, resulting in data which shows the position of each person and whether a mask is worn. This is done by calculating the overlapping area of each face prediction to the pedestrian prediction and merging each pair with the highest value.

Perspective transform

As cameras introduce a perspective distortion to each image, distances cannot be measured directly. Therefore, each point is being transformed into the bird's eye perspective using the homography matrix computed while calibrating.

Distance measurement

The resulting points of the perspective transform are used to calculate the distances between every point. To convert them into real-world distances, the reference distance from the calibration database is used. When a give distance is lower than a set threshold it is passed on, all other measurements are ignored.

Image rendering

In the end, the gathered information is used to render the bounding boxes of the pedestrians and the distances between them. This is done using the Python Library PIL which provides useful functions for drawing text and figures on images. The resulting image is then exported as a JPEG image.

Software Requirements Specification

for

Babyelefant

Version 1.0 approved

Prepared by Gäbler, Patek, Pressler

HTL Spengergasse

26.02.2022

IEEE 830-1998

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Lukas Gäbler	2022-02-20	keywords	0.1
Lukas Gäbler	2022-03-21	Various changes to all topics	0.2
Tobias Pressler	2022-02-26	Changes on topic 3.3 and 3.4	0.3
Lukas Gäbler	2022-02-26	Revise document	1.0
Lukas Gäbler	2022-03-15	Various changes	1.1

1. Introduction

1.1 Purpose

This document is the software requirement specification for the project Babyelefant. It gives a detailed view into the software implementation of the project.

1.2 Intended Audience and Reading Suggestions

This software requirement specification is intended for developers, project managers, testers and project overseers. For navigation and orientation refer to the table of contents.

1.3 Product Scope

The main goal is to provide a system which can measure the distance between people in order to control if they keep the minimum required distance because of COVID-19. The project also includes a website which displays the collected distance data in over-time-charts.

2. Overall Description

2.1 Product Perspective

Our product is a new self-contained product and, as far as we know, the only product that exists with these features.

2.2 Product Functions

The main function is to measure the distance between people on images or videos. Each person will be anonymized in order to protect these people and keep their identities hidden. Details will be provided in sections 3 and 4.

2.3 User Classes and Characteristics

Our user classes are mainly people who own a business where many people are present so the risk of getting infected with COVID – 19 is higher as it is not sure whether the safety measures can be implemented as well as developers who are interested in computer vision.

2.4 Operating Environment

The system is based on a Debian LXC Container. The hardware includes an Intel Celeron J4125 processor, 8GB of RAM and roughly 100GB of disk space. In order to start the system, python needs to be installed. OpenVPN Server is also necessary for the system to function.

2.5 Assumptions and Dependencies

The project depends on a VPN connection between the server and the client in order to provide a safe connection where the video feeds are sent. The software on the server is responsible for the processing of the video feeds from the client. During this process, the distance is measured, masks will be detected, and the data will be saved in the database. The server runs on hardware hosted by us.

3. External Interface Requirements

3.1 User Interfaces

The website is the main user and admin interface to manage the system. The website will change due to feedback by partners. The following screenshot should provide an impression on how the website looks like.

3.2 Software Interfaces

A REST – API is used to process the data from the database and send it to our website. The specification for our REST – API will constantly change, as we do not know the full feature extend due to our agile approach.

3.3 Communications Interfaces

To send the images from the camera's location to our server, the ZeroMQ protocol is used. The traffic is secured via a OpenVPN VPN tunnel. Websockets are used to get updates when the camera configuration changes while the client is running.

4. System Features

As we are using a combination between waterfall and agile project management our system features are constantly changing. The core features will be described in this document.

4.1 Automatic distance measuring

4.1.1 Description and Priority

The automatic distance measuring is the most important feature of the project (priority: 10). It contains the person detection, view conversion and the algorithm to measure the distance. The collected distance data will be stored in the database so the user can access them and get an overview about how the distance which the people keep on the event.

4.1.2 Functional Requirements

REQ-1: The server receives the camera streams.
REQ-2: The python backend is working.
REQ-3: The AI on the server is running.
REQ-4: The data is written into the database.
REQ-5: The database server is running.

4.2 Video Feeds

4.2.1 Description and Priority

For the user to see what is going on at the event, he or she can see the video feeds from the cameras with the people anonymized on the website. The overlay for the people which are keeping their distance and wear a mask, is green. The overlay for people who are not keeping distance, but wear a mask, is orange. The overlay for the people who are not wearing a mask and do not keep the distance is red.

4.2.2 Functional Requirements

REQ-1: The person detection is working.
REQ-2: The python backend is working.
REQ-3: The Vue frontend is working.
REQ-4: The webserver is reachable from any web browser.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

As we are handling data which is best for the user to know about in real time, it is necessary to minimize the handling time on the server between receiving the image from the camera and returning the new image with the overlay and the distance data written into the database. As far as our tests have shown, this depends on the network quality from the client.

5.2 Safety Requirements

As we store no personal data, users do not have any risk of their data being stolen by using our product. Nevertheless, there is a risk of the attendees of the event from the user getting infected with COVID-19, as our product does not prevent an infection. Besides that, keeping distance and wearing a mask are not the only factors from which can prevent an COVID-19 infection.

5.3 Security Requirements

As we are transferring images of different people over the internet it is important that these images, as they contain personal data (images of people), cannot be stolen. We use a VPN to secure the connection and the safety of the images. After the images get to the server the people will be anonymized and will not be stored, so there should be no further concern about the safety.

5.4 Software Quality Attributes

The user interface (website) will be intuitive so there are no unclear parts where the user does not know how something works or how to get somewhere.
The correctness of the measured distance is the most important quality attribute. If the measurements are not correct, the data we are collecting is not reliable.