



Hochschule Reutlingen
Reutlingen University



Uwe Kloos, Natividad Martinez, Gabriela Tullius (Hrsg.)

19 Informatics
Inside

Tagungsband

Hochschule Reutlingen



infoinside.reutlingen-university.de
infoinside@reutlingen-university.de





Hochschule Reutlingen
Reutlingen University



Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Informatics Inside experience(IT);

Informatik-Konferenz an der Hochschule Reutlingen
8. Mai 2019

Impressum

Anschrift:

Hochschule Reutlingen / Reutlingen University
Fakultät Informatik
Human-Centered Computing
Alteburgstraße 150
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002

Telefax: +49 7121 / 271-4042

E-Mail: infoinside@reutlingen-university.de

Internet: <http://infoinside.reutlingen-university.de>

Organisationskomitee:

Prof. Dr. Gabriela Tullius, Hochschule Reutlingen

Prof. Dr. Natividad Martínez, Hochschule Reutlingen

Prof. Dr. Uwe Kloos, Hochschule Reutlingen

Tanja Brodbeck

Markus Gromer

Julian Hennige

Tobias Lampprecht

Markus Rein

David Salb

Panagiotis Taxidis

Tim Wiedemer

Johannes Timotheus Zillig

Copyright: © Hochschule Reutlingen, Reutlingen 2019
Herstellung und Verlag: Hochschule Reutlingen



Hochschule Reutlinge
Reutlingen University

Vorwort

Sehr geehrte Leserinnen und Leser,
liebe TeilnehmerInnen der Informatics Inside,

bereits zum elften Mal findet nun die Studierendenkonferenz Informatics Inside statt. Als Teil des Masterstudiengangs Human-Centered Computing organisieren Masterstudierende selbständig eine vollumfängliche wissenschaftliche Konferenz. Dazu gehört neben der eigentlichen Veranstaltung auch der Auswahl- und Review-Prozess der eingereichten Beiträge, um für die nötige Qualitätssicherung zu sorgen. Die Informatik ist nach wie vor ständigem Wandel unterworfen. Unsere Studierenden tragen diesem Wandel bei, indem sie in Ihrer wissenschaftlichen Vertiefung aktuelle Problemstellungen durch innovative Konzepte lösen. Inzwischen ist die Informatik aber auch nicht mehr immer sofort sichtbar: Nicht überall, wo Informatik drinsteckt, steht auch Informatik drauf! Das merken wir immer dann, wenn etwas nicht wie vorgesehen funktioniert. Aktuelle Beispiele finden sich in Softwarefehlern in der Steuerung von Passagierflugzeugen, die bereits zu Abstürzen führten; in Erpressungstrojanern, die immer noch Ausfälle von Systemen und finanzielle Verluste fordern; oder in illegaler Software zur Manipulation der Abgaswerte von Fahrzeugen. Das diesjährige Motto der Informatics Inside ist *experience (IT);*, verdeckt als Funktionsaufruf :). Lassen Sie uns also erfahren, was aktuelle Themen in der Informatik sind! Gleich zwei spannende Fragestellungen beschäftigen sich mit dem hochaktuellen Gebiet der künstlichen Intelligenz. In einem Beitrag geht es um ein Verfahren zur Beschreibung von Bilddaten auf Basis eines Machine-Learning Ansatzes und in einem weiteren um die Vorhersage von Erdbeben durch die Verwendung neuronaler Netze, die dem menschlichen Gehirn nachgebildet sind. Auch das dritte Thema, bei dem Ideen aus der Computerspiel-Szene verwendet werden, um einen Maschinen-Service interessanter zu machen, ist hochaktuell. Ausgefeilte Spiele, die auf virtuellen Welten basieren, sind nach wie vor auf Erfolgskurs. Warum also sollte man nicht solche Ansätze viel stärker in sonst eher trockenen Aufgabenbereichen einsetzen?

Ich bin begeistert, wie es unsere Studierenden schaffen, bekannte Konzepte in immer neue Anwendungskontexte zu übertragen und so neue sowie spannende Lösungsansätze schaffen! Diese Leistung ist ausgezeichnet und gehört für mich zu dem, was unsere Hochschule ausmacht: Innovation, Mut, Ergebnisoffenheit.

An die LeserInnen und die AutorInnen des Tagungsbandes habe ich jeweils einen Wunsch: Viel Vergnügen bei der Entdeckung neuer Ideen aus den Artikeln, und weiterhin ein erfolgreiches Studium sowie einen guten Start in ihr Berufsleben!

Es grüßt Sie herzlich aus Reutlingen,
Christian Kücherer

Inhaltsverzeichnis

Steven Cybinski

Semi-automated image data labelling using AprilTags as a pre-processing step for machine learning.....1

Julian Hennige

Zeitliche Vorhersage von Erdbeben durch Klassifizierung mittels Convolutional Neural Networks.....11

Tanja Brodbeck

Anforderungen an ein Gamification-Konzept zur Motivationssteigerung der Anwender eines KI-Service zur Maschinoptimierung..... 21

Semi-automated image data labelling using AprilTags as a pre-processing step for machine learning*

Steven Cybinski
Reutlingen University
Steven.Cybinski@student.
Reutlingen-University.DE

Abstract

Data labelling is a pre-processing step to prepare data for machine learning. There are many ways to collect and prepare this data, but these are usually associated with a greater effort. This paper presents an approach to semi-automated image data labelling using AprilTags. The AprilTags attached to the object, which contain a unique ID, make it possible to link the object surfaces to a particular class. This approach will be implemented and used to label data of a stackable box. The data is evaluated by training a You Only Look Once (YOLO) net, with a subsequent evaluation of the detection results. These results show that the semi-automatically collected and labelled data can certainly be used for machine learning. However, if concise features of an object surface are covered by the AprilTag, there is a risk that the concerned class will not be recognized. It can be assumed that the labelled data can not only be used for YOLO, but also for other machine learning approaches.

Keywords

Semi-automatic Data Labelling, AprilTags, ArUco, Pre-Processing, Machine Learning, You Only Look Once (YOLO)

*

Betreuer Hochschule: Prof. Dr.-Ing. Cristóbal Curio
Hochschule Reutlingen
Cristobal.Curio@Reutlingen-
University.de

Informatics Inside 2019
Wissenschaftliche Vertiefungskonferenz
08. Mai 2019, Hochschule Reutlingen
Copyright 2019 Steven Cybinski

CR-Categories

D.2 [ACM]: SOFTWARE ENGINEERING;
I.2 [ACM]: ARTIFICIAL INTELLIGENCE;
I.4 [ACM]: IMAGE PROCESSING AND
COMPUTER VISION

1 Introduction

Data labelling is a pre-processing step for the preparation of training data to train a neural network. The compilation of labelled image data is a time-consuming activity. The main reason for this is the need for a larger dataset consisting of many image data with corresponding labels. For each image, the required label information must be recorded, such as the position of the object(s). The formatting and storing of the label information depends on the later used framework for machine learning.

Many datasets can be found on the Internet. Some of them have been created automatically and therefore contain potentially incorrect information. The dataset may have to be formatted accordingly to the used framework understands it. But if it is a special object, the dataset itself must be created. In the context of a project at the university, for example, stackable boxes and the visible sides must be detected and classified. There is no suitable dataset for these stackable boxes on the Internet.

If a dataset needs to be created where the object has essentially no optical variation, this can be done with a single physically available object. In this case, the preparation of the object, for example with AprilTags, is not

a work-intensive task. The AprilTags, which will be explained later, can be detected using basic image processing techniques. If they are attached to an object, it is possible to link the object surfaces to a particular class. Thus objects that have such a marker can be localized and classified relatively easy. This approach enables the creation of datasets without the need for time-consuming manual input. With the help of camera calibration, additional information such as a 3D pose and the distance to the camera can be estimated.

1.1 Motivation and Objectives

As previously mentioned, AprilTags can form the basis to create a dataset. If it is well done, it will not matter if it is a stackable box or another object. The aim of this work is to develop a tool for creating a training dataset using AprilTags. So that the first step to machine learning can be completed as easy as possible. The tool should support the connection of Microsoft Kinect and Intel RealSense, but should also be easily expandable by further camera types. Through a dynamic configuration it is possible to set marker ID related classes as needed, so that the tool can be used as flexibly as possible. After configuration and preparation, the object can be recorded via a user interface and stored in a simple and framework-independent format. The subsequent evaluation of the generated data will be performed for the previously mentioned university project scenario. The generated dataset through the tool will be used for YOLO training. This will prove whether the dataset is usable for this task and whether YOLO itself can be trained. However, the prerequisite for this is that the data record is converted to the appropriate format. The general aim is that the tool will support future projects in creating a machine learning solution.

1.2 Structure of the Paper

At the beginning of this paper the state of the art will be discussed. This includes technologies which were used for this project and one project which also dealing with semi-

automated data labelling. Subsequently, the solution approach of labelling with AprilTags is explained. This includes the basic idea and process of labelling, on which a tool was implemented.

Then the result is validated in practice by training an object detection framework with the generated dataset. Finally, a conclusion is given based on the findings.

2 State of the art

This section explains the used technologies which already exist. Afterwards related papers with almost the same aim but another way to solve the labelling challenge are named.

2.1 Used technologies

In this subsection, the important technologies which will be used are briefly presented. This includes the aforementioned AprilTags with the library ArUco for detection of square fiducial markers. Then the real-time object detection system YOLO and the neural network framework Darknet is explained. YOLO and Darknet are only briefly mentioned here, as these technologies were only used for evaluation reasons and are not part of the actual project.

2.1.1 AprilTags

AprilTags are 2D barcodes developed and described in [1] by Edwin Olson for Robotics. At the first look they resemble the widely used QR codes. Unlike QR codes, they do not have more complex embedded information, such as web addresses, but only a single ID. Depending on the family, a certain number of unique IDs can be used. In [1] the authors recommended the family *36h11*, therefore this family was used for this project. In addition, the developed AprilTag detector is robust against illumination variations and occlusions and allows an exact localization of the markers [1].

There are also markers, such as the ARTags. But for this project the AprilTags were used because they delivered a good detection result. Moreover, a comparison between the dif-

ferent markers is not part of the work. In addition, the tests from [1] showed that April markers can be detected from a greater distance and at a greater angle to the camera. However, the configuration and use of ArUco allows many more markers to be used in this project, if they are supported by ArUco library.

2.2 ArUco

ArUco Marker and the detection library was developed by Rafael Muñoz and Sergio Garrido and described in [2]. One main point of the project described in the paper is the camera pose estimation under consideration of occlusion. The development was primarily focused on a solution for robotics and augmented and virtual reality applications.

The markers of ArUco look very similar to the AprilTags and also contain only a single ID. More interesting for this project is the library of ArUco, which can detect different square fiducial markers [2]. This includes the whole palette of AprilTags, ARTags, ArUco markers and many more. In [2] the authors show, that augmented reality work with AprilTags very well. If the marker and object size is known, it is possible to give the synthetic object a real size. For example, you can specify that a object should be exactly five centimetres high in the real world. To make this possible it is assumed that the camera has been calibrated beforehand.

2.2.1 You Only Look Once and Darknet

YOLO is a real-time object detection system developed by authors from [3]. YOLO, unlike other detection methods, looks at the image only once, this is where the name comes from. Due to this, YOLO is the fastest detection method, with up to 155 frames per second, and without much loss of quality [3]. This is possible because this method follows a different approach than the other methods. YOLO goes “straight from image pixels to bounding box coordinates and class probabilities” [3].

Darknet is a framework for neural networks

which was developed by Joseph Redmon [3]. The author does not write a paper for Darknet, but you can see on his webpage¹ that it was developed especially for YOLO. In summary this framework makes it possible to use YOLO.

2.3 Related work

In [4], the authors Daniel Lopresti and George Nagy dealt with the topic of interactive labelling, classifier training, and automated labelling. There are several publications on this approach which were referenced in [4]. In their work, the authors have looked at how this approach can be improved by optimal data partitioning.

One system explicitly mentioned in the paper was the Computer Aided Visual Interactive Classification (CAVIAR) system. The idea is, that a classifier is trained interactively and iteratively with the help of a human operator. In [4] the procedure was explained as follows. A sequence of N patterns is manually labelled by an operator. After the sequence, a classifier is trained with this data and then applied to a new and slightly larger sequence of patterns. The operator now checks the result and makes corrections, such as removing false positives or adding new labels to false negatives. After this sequence, the classifier is retrained and the result is applied to a new sequence. This process is repeated until a certain abort criterion is met, e.g. a low error rate.

This approach allows a fast and comfortable training and could be extended by the functionality, that the labelled data can be stored separately and used for other models. However, it can be assumed that an untrained model initially requires a lot of work because its classification is insufficient. If an already sufficiently trained model is used, the quality can be improved relatively easy by this approach.

3 Labelling with AprilTags

This chapter will explain the semi-automatic labelling approach of this paper. As already

¹See <https://pjreddie.com/darknet/>

mentioned in chapter 2.2, a 3D pose can be estimated from the detected markers, like in augmented reality. If there is also a definition of the ID and class relation as well as the size proportions, then the required information for semi-automatic labelling is given.

The labelling process basically consists of four steps. Each frame reads in the first step passes through the entire process as long as the tool is running. In summary, image resources are retrieved and the markers are detected. If markers have been found and the marker ID is known, the marker information and the configuration will be used to create labels. In order to not record too much similar data, the operator must indicate if he wants to record this frame and label. This assumes that at least one object was detected and labelled.

The following subchapters first explain the camera calibration and configuration. Subsequently, the four previously mentioned steps are explained in more detail.

3.1 Camera calibration and dynamic configuration

In this subchapter, two important points are explained which must be given as a precondition for the labelling. This includes camera calibration and configuration.

3.1.1 Camera calibration

The camera calibration is an essential step for this project. The labelling can only run properly with a good calibration. An projection of three-dimensional coordinates onto a planar image can be carried out by the so-called pinhole camera model, which requires a camera calibration [5]. In this project the calibration was done with the AprilTags. There are also alternative methods, such as calibration using a chessboard.

The calibration is performed by recording a chessboard or a board with AprilTags from different perspectives. A batch is created from the sum of the detected features, which is later used to calculate two matrices using the ArUco library. One resulting matrix is the camera matrix or the pinhole camera model,

which is responsible for mapping 3D points on an image plane. “However, using the ideal pinhole model in isolation will poorly capture the dynamics of most real world lenses, especially those with a wide field of view” [5]. Therefore, the model is extended by another distortion matrix that is used to correct the lens distortion [5].

3.1.2 Dynamic configuration

The configuration of the tool defines important basic information for the application and labelling. In order to not only provide a solution for the stackable boxes scenario, it was defined as a requirement for the tool that the labelling should be as dynamically configurable as possible.

First, the marker family to be detected is determined. Below is a list with all objects that should be labelled by the tool. The objects contain a list of classes, such as front, back, left and right. Each class gets unique marker ID as well as the marker size and the size of the white marker border. In addition, the objects dimensions will be defined from the respective point of view. Since the markers cannot always be placed in the centre of the object, it is possible to set offsets to adjust the distance from the center. To keep the same objects apart, they should be defined with different marker IDs as often as needed. This allows the tool to determine which marker belongs to which object, even if it is of the same type, e.g. box.

3.2 Read frame from interface

In the developed tool, an interface was used that allows relatively simple integration of additional cameras. The interface provides functions for fetching RGB and depth frames. The initialization and connection of the camera source is defined in the constructor. Another predefined function is the one that specifies how to disconnect from the camera source. Therefore, with a few definitions, almost any camera can be supported by the tool, regardless of the resolution or frame rate of the camera.

3.3 Detect and interpret markers

Marker detection is the basic part of labelling, as it is used to derive further information for labelling. As mentioned before, the ArUco library was used for marker detection.

The detection is explained by [2] in four essential steps, which are explained for illustration in relation to figure 1. In (a) you can see the original image for which a marker detection is performed. The first step is the image segmentation. Here the most important contours from the greyscale image are segmented by means of local adaptive thresholding (b) [2]. In the second step the contours are extracted and filtered. The Suzuki and Abe algorithms will be used to extract the contours from the segmented image, resulting in a set of image contours (c) [2]. By applying the Douglas-Peucker algorithm, a polygonal approximation is performed, causing the contours that are not attached to four vertex-polygons to be discarded [2]. After simplifying the contours by keeping only the outer contours, the result will look like in (d) [2]. The third step deals with marker code extraction. A homography matrix is calculated and the respective tag is normalized (e) [2]. The optimal threshold value is then determined using the Otsu's method and applied to the image [2]. This results in a binary image that is divided into a grid where each cell is assigned to the value 0 or 1, depending on the majority of pixels (e,f) [2]. If not all border regions have the value 0, an additional analysis is aborted and the marker candidate is discarded [2]. In the fourth step the marker identification and error correction are performed. The aim is to check which marker candidates belong to the dictionary and which are just part of the world [2]. The dictionary is converted into a binary tree and searched for the concatenated bits of the marker [2]. A marker candidate is valid if it was found in the dictionary.

The interpretation of the marker IDs is performed using the configuration from chapter 3.1.2. A list of visible objects is created from the object definitions of the configu-

ration and the detected markers. All further steps are performed on the basis of this list. If the multiple IDs of the same object are visible, such as front and left, an object with four pages is created in this step. If several IDs of the same object are visible, such as front and left, this is registered in the particular object. Therefore, even if the objects are of the same type, they should have unique IDs, as mentioned in chapter 3.1.2. Up to this point, the labelling provides information about which objects are visible with which classes and where exactly their reference points (markers) are.

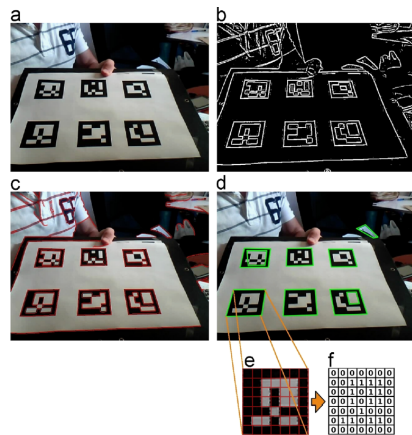


Figure 1: “(a) Original image. (b) Result of applying local thresholding. (c) Contour detection. (d) Polygonal approximation and removal of irrelevant contours. (e) Example of marker after perspective transformation. (f) Bit assignment for each cell.” [2]

3.4 Generate label information

Through the previous step a list of visible objects was created from the detected markers. This chapter explains how further labelling information is generated from these objects. The generation of label information is performed with four additional steps, which will be explained in the following subsections.

3.4.1 Define and inpainting

Since the markers do not be included in the target dataset, inpainting is used to remove the markers. Inpainting enables the reconstruction of an image area using the surrounding image environment. Inpainting was implemented in the tool with the Open Source Computer Vision Library (OpenCV). For all detected markers, the corner points were first expanded by the white border defined in the configuration. This was specified larger, so that the newly defined area is slightly larger than the original marker, see 2 (b). These areas were then drawn on a mask (c), that specifies where a reconstruction should be made. When choosing a range of equal size, the white border is taken into account by the algorithm. In this case this region looks like a bright spot. OpenCV provides a few inpainting algorithms, from which the Telea method was chosen because of its performance and quality. The Telea method was developed by Alexandru Telea and described in [6]. It is based on a Fast Marching Method and tries to reconstruct relatively fast with consideration of gradient direction and photometry [6].

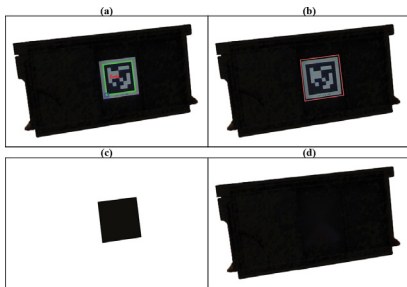


Figure 2: Illustration of the inpainting process. (a) Detection of the marker. (b) Resizing the detection area. (c) Inpainting mask. (d) Result of the inpainting.

Once the inpainting has been applied, the result should look like figure 2 (d). In this step no relevant labelling information was created, but preparation of the image material was made, so it can be used for machine learning.

Now the markers do not falsify the result of the detector to be trained. In order to not waste too much computing power, the inpainting is only used if the corresponding preview option has been activated via the GUI or a recording is performed.

Unfortunately, this step has a weakness. If only a small part of a marker is visible and therefore it is not detected, no inpainting will be performed for this region. As a result, individual recordings may occur that falsify the dataset. Further information on this can be found in chapter 4.

3.4.2 Estimate 3D pose

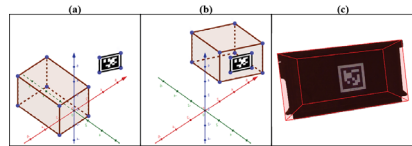


Figure 3: 3D pose estimation by projection. (a) Definition of the 3D object and the position of the detected marker. (b) Transformation of the 3D object to the marker. (c) Result based on a real recording.

Based on the previously generated list of objects and detected markers, the 3D pose of the object is determined. The object is first defined in the origin using the configured size, as shown in figure 3 (a). In this case, the origin corresponds to the centre of the marker. Therefore, the points of the front side seen from the marker are defined with the Z coordinate 0 and the points of the back side with a corresponding negative Z coordinate, since the object is located behind the marker. The centre of the front must be the origin itself, if it is assumed that the marker is centred on the object. If an offset is defined in the configuration, the entire object is transformed accordingly. Then, as shown in b, the defined 3D object must be transformed to the position of the marker and then assigned to the image plane. For the transformation and the mapping OpenCV offers a function called *projectPoints*. As input param-

ters the function needs the points of the 3D object, the output rotation vector (*rvec*) and output translation vector (*tvec*) of the respective marker as well as the camera and distortion matrix. Where *rvec* and *tvec* are used for the object transformation and the camera and distortion matrix for the projection. The resulting image points can now be drawn to the image plane, as (*c*) illustrates.

After this step we know exactly where the detected objects are and which pose they have. The obtained image points are also stored in the respective object label and used in the next step.

3.4.3 Extract detection windows

In this step, a detection window is extracted from the previously generated image points. To define the detection window the first step is to iterate through the eight image points to find the minimum and maximum values of the coordinates. Then the four corner points of the detection window *DW* are defined as follows.

$$DW = \{[x_{min}, y_{min}], [x_{max}, y_{min}], [x_{max}, y_{max}], [x_{min}, y_{max}]\} \quad (1)$$

Figure 4 shows the recognition window determined in this way, whereby the projected 3D pose is essentially framed.

This step is uncomplicated, because the positions of the objects is already known. The definition of a detection window was nevertheless done, because many training datasets only aim at a pure object detection and no 3D pose detection.

3.4.4 Estimate distance

The distance between the camera and the respective object is determined by the Euclidean distance. The camera position, which in this case is the origin $o = [0, 0, 0]$ itself, and the *tvec* of the marker are required for this calculation. Then the Euclidean distance is calculated as follows.

$$distance = \sqrt{(tvec_0 - o_0)^2 + (tvec_1 - o_1)^2 + (tvec_2 - o_2)^2} \quad (2)$$

If the markers and object size are specified in millimetres in the configuration, the result must also be interpreted in millimetres. The distance to the objects is also stored in the label information of the respective object.

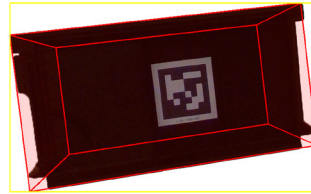


Figure 4: Determination of a detection window using a known 3D pose.

3.5 Create a record

Records are triggered by the operator. The creation of a record is done by generating a JavaScript Object Notation (JSON) object from the label information and applying in-painting on the RGB image. Recording is only possible if at least one object has been recognized, so that the label contains at least one object. Each object has a unique ID for this recording, as well as the previously generated label information. Among other things, each object contains at least one visible side that contains a class name and a distance. If an object is positioned so that two markers are recognized, both visible sides could be listed here.

Subsequently, the JSON object, depth and RGB image is stored with the same name, but with the corresponding prefix, in the configured path. The three data types are organized in three separate subfolders, so that they are not mixed. A framework specific preparation as well as composition of the data was intentionally left out, so that the datasets can be used in a wide range.

4 Project result evaluation

This chapter evaluates the previously presented project. First a look is taken at the generated label information and then a practical test is performed.

4.1 Label information

In this subchapter the 3D pose estimation and the derived detection window quality are considered. Subsequently, a small experiment will show how accurate the distance estimation is.

4.1.1 3D pose estimation and detection window

The quality of the 3D pose estimation and the detection window cannot be measured directly, because there is no reference data for comparison. Therefore, the position is taken in relation to the visual impression. Table 1 shows a ranking that was performed in the context of this work. Essentially it was looked at how the camera resolution and marker size affect the quality of the 3D pose estimation. It is assumed that the configuration of the markers and object size as well as the offsets were set correctly and accurately during this evaluation. It is also assumed that the calibration of the camera was done properly. Moreover, not all variations were examined, since the individual estimations already allow a good estimation of the results. The ratings of the table have the meaning *dash* for bad results, *circle* for acceptable results and *plus* for good results.

As shown in the table, the resolution of the camera has the greatest effect on the result. At a low resolution, an acceptable result can be achieved with a large marker, while at a higher resolution, slightly smaller markers allow a good result. The best result, with an optically very well-fitting 3D pose estimation, was achieved with the resolution $1280px \times 720px$ and a marker size of $50mm \times 50mm$. Therefore, this constellation was used for the dataset evaluation in chapter 4.2. Since the detection window is derived from the projected 3D pose and is therefore in a qualitative relation, no separate evaluation was performed for this.

4.1.2 Distance estimation

This chapter examines the accuracy of the previously mentioned distance estimation is.

The proposed setup from chapter 4.1.1 was used for this purpose. For the experimental setup, the camera was set to the height of the marker, so that the distance is not distorted by a vertical shift. To avoid distortion of the results by a horizontal shift, the marker was always aligned so that the center of the marker corresponds to the center of the camera image. The marker was then measured in 100 mm increments, as shown in table 2 *Real distance (mm)*. It is assumed that a proper camera calibration was performed.

During the experiment, it was observed that the distance estimation did not show a fixed value. Rather, this value varied from frame to frame, so that the maximum and minimum estimated distance were recorded. Even the minimum and maximum values show that the estimate can be distorted by about one centimetre.

In summary, the distance estimation is quite good, even if the label information is not strictly suitable for critical applications that need to accurately determine the distance. If necessary, an exact determination of the distance should be done by a more robust solution anyway. For simpler scenarios such an estimate is quite enough.

Table 1: 3D position estimation in the context of camera resolution and marker size

Cam. resolution	Marker size	Ranking
640x480 px	15x15 mm	-
640x480 px	25x25 mm	-
640x480 px	50x50 mm	o
1280x720 px	15x15 mm	-
1280x720 px	25x25 mm	o
1280x720 px	50x50 mm	+

4.2 Evaluation of the generated dataset

In this chapter, a dataset generated by the tool is tested in practice. This includes the preparation of the dataset, the training of a YOLO detector and the discussion of the results.

The stackable box problem was chosen as the experimental scenario. A total of 449 la-

Table 2: Evaluation of distance estimation

Real distance (mm)	200	300	400	500	600	700	800	900	1000
Min. estimated distance (mm)	197	298	399	498	597	698	789	890	991
Max. estimated distance (mm)	203	302	401	501	600	703	810	908	1008
Average difference (mm)	0	0	0	0,5	1,5	0,5	0,5	1	0,5

belled images were recorded for this experiment using the setup suggested in chapter 4.1.1. This preparatory work took a little over an hour. This step includes the mounting of AprilTags, configuration of the tool and the recording itself. This dataset contains the four classes mentioned in chapter 3.1.2.

4.2.1 Prepare dataset

As mentioned before, the dataset is stored in a way that is independent of the framework. Therefore, in this step the dataset must be transferred to the structure that YOLO can understand. The basic definition of the training is a fundamental task and is not relevant for this paper. More interesting is the transformation of the dataset itself, YOLO needs the data in a folder with image and label pairs each. The image can be easily adopted, but would have to be resized depending on the configuration. However, YOLO cannot handle the label information from the developed tool, since this framework only needs the detection windows with the respective classes.

The required information is available in the generated dataset, so it only needs to be parsed. As part of this experiment, a parser was built that needed about three to five seconds to parse the 449 labelled images. The preparation of the dataset was therefore quick and easy, since the JSON format is simpler to process and the expected YOLO format does not require any special things.

4.2.2 Training with the dataset

After the training configuration and the dataset preparation, the training could be done with Darknet. With the *YOLOv2*² model and

²The YOLO models are presented in <https://pjreddie.com/darknet/yolo/>.

the 449 labelled data, the training has reached an average loss of 0,28 after about six hours of training with an *Intel Core i7-6700K* and *Nvidia GeForce GTX 1080 Ti*, which is a pretty good result. Darknet itself aborts the training as soon as the average loss dropped below 0,3.

4.2.3 Training results

After the training was completed, YOLO was able to detect the four classes, which are shown in figure 5. In addition, an attempt was made to illustrate how well *YOLOv2* was trained. Of course, the quality also depends on the framework itself. Since the developer's paper has already shown that the detection of YOLO can provide pretty good results, it is interesting to find out how YOLO performs with the self-generated dataset.

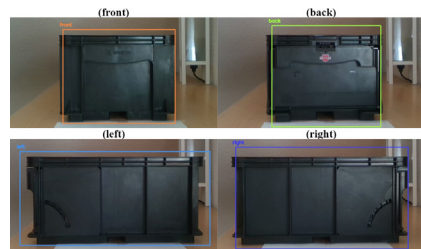


Figure 5: Label definition in YOLO.

A total of 2000 frames, with 500 frames per class, were processed with YOLO for the experiment. The object was rotated, placed further or closer to the camera and exposed to changing light reflections during the experiment. Table 3 shows the result of the experiment. The detection of the front and back sides has achieved a pretty good detection rate. However, as soon as the light conditions became too poor, causing the object to be parti-

Table 3: Detection results of YOLOv2

Frames with	Correct	Wrong	Not
Front class	92,4%	2%	5,6%
Back class	93,8%	0%	6,2%
Left class	63,4%	13,88%	22,72%
Right class	50,4%	31%	18,6%

ally overexposed or underexposed, the object could not be detected. Whereby light conditions are meant, under which the human eye already gets problems to determine the object. The false detections occurred as soon as the essential features of the class, the raised font on the front side and the red logo on the back side, were not recognizable. The detection of the left and right side did not perform this well, as these two classes are more susceptible to light conditions. The main reasons for this are the weakly recognizable features and indentation on the left and right side, see figure 5 (*left*) and (*right*). As soon as the object was slightly over- or underexposed, this feature could not be detected, so that the object was partly not or incorrectly detected.

5 Conclusion

In this paper an approach for a semi-automated labelling tool was presented, which was implemented in the context of this work. The tool achieved a very accurate labelling result with a camera resolution of $1280px \times 720px$ and a marker size of $50mm \times 50mm$.

The practical test using a dataset of 449 labelled images of stackable boxes to train a YOLO network resulted in a fairly good result. The evaluation revealed that the marker had to be attached in such a way that it did not cover any important features of the class or object. The whole procedure for creating a detector, without an existing dataset, took about seven to eight hours along with the training. For this purpose, a small parser was developed, which converted the independent dataset structure into a structure understandable by YOLO.

In the future the tool could be extended by the approach of [4]. A detector could be trained

in parallel, which also performs detections during labelling. The markers could be used to identify false defects and simultaneously create a dataset with negative samples.

In summary, it can be said that the use of AprilTags is suitable for semi-automated labelling. Scenarios such as the identification of living creatures or other objects that cannot be prepared with markers are not suitable for this approach. For scenarios such as the detection of visible sides of stackable boxes, this approach is very well suited.

Literatur

- [1] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.
- [2] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. volume 47, pages 2280–2292. Elsevier BV, 2014.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.
- [4] D. Lopresti and G. Nagy. Optimal data partition for semi-automated labeling. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 286–289, 2012.
- [5] A. Richardson, J. Strom, and E. Olson. Aprilcal: Assisted and repeatable camera calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.
- [6] A. Telea. An image inpainting technique based on the fast marching method. volume 9, pages 23–34. Informa UK Limited, 2004.

Zeitliche Vorhersage von Erdbeben durch Klassifikation mittels Convolutional Neural Networks

Julian Hennige
Reutlingen University
Julian.Hennige@student.
Reutlingen-University.DE

Abstract

In dieser Ausarbeitung wird eine zeitliche Vorhersage von Erdbeben getroffen. Hierfür werden mit einem Datensatz aus Labor-Erdbeben Convolutional Neural Networks (CNN) trainiert. Die trainierten Netzwerke geben Vorhersagen, indem sie einen Input an seismischen Daten klassifizieren. Durch das Klassifizieren kann das CNN die zeitliche Entfernung zum nächsten Erdbeben vorher-sagen. Es werden hierfür zwei Ansätze miteinander verglichen. Beim ersten Ansatz werden die Originaldaten in ein CNN gegeben. Beim zweiten Ansatz wird vor dem CNN eine Vorverarbeitung der Daten mit den Mel Frequency Cepstral Coefficients (MFCC) durchgeführt. Es zeigt sich, dass mit beiden Ansätzen eine gute Klassifikation möglich ist. Die Kombination aus MFCC und CNN liefert die besseren quantitativen Ergebnisse. Hierbei konnte eine Genauigkeit von 65 % erreicht werden.

Betreuer Hochschule: Prof. Dr. Benjamin Himpel
Hochschule Reutlingen
Benjamin.Himpel@Reutlingen-
University.de

Informatics Inside
Wissenschaftliche Vertiefungskonferenz
08. Mai 2019, Hochschule Reutlingen
Copyright 2019 Julian Hennige

Schlüsselwörter

Erdbebenvorhersage, Convolutional Neural Network, Machine Learning, MFCC

CR-Kategorien

Computing methodologies → Machine learning → Machine learning approaches → Neural networks

1 Einleitung

Aufgrund der verheerenden Folgen von Erdbeben ist ihre Vorhersage eine der wichtigsten Herausforderungen der Geowissenschaften. Forschungen zur Prognose von Erdbeben konzentrieren sich auf drei Kernfragen: *Wann*, *Wo*, und *Wie stark* wird das Erdbeben sein? Diese wissenschaftliche Ausarbeitung konzentriert sich auf das *Wann*. Mittels CNNs soll vorausgesagt werden, wie weit das nächste Erdbeben zeitlich noch entfernt ist. Bei den verwendeten Daten handelt es sich um Labor-Erdbeben. Ansätze des Machine Learning zur Vorhersage von Erdbeben bieten eine Möglichkeit, welche bei richtiger Anwendung Menschenleben und Infrastruktur retten können [10].

Folgende Beispiele zeigen die Folgen von Erdbeben. Das größte Erdbeben der letzten Jahre passierte 2011 an der japanischen Küste, das einen wirtschaftlichen Schaden von 210 Milliarden USD verursacht hat. Dieses Erdbeben führte zu einem verheerenden Tsunami, der für die hohen Schäden verantwortlich war [17]. Beim Erdbeben 2004 in Thailand starben 220.000 Menschen. Auch hier löste das Erdbeben einen Tsunami

aus [18]. Die Fakten zeigen was ein Erdbeben für Folgen haben kann und wie wichtig eine qualitativ hochwertige Vorhersage von Erdbeben ist. Eine falsche Vorhersage kann im Falle eines Erdbebens zu unnötiger Panik oder im Falle eines bald auftretenden Erdbebens zu falscher Sicherheit führen.

2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es, aus den seismischen Signalen den zeitlichen Abstand bis zum nächsten Laborerdbeben vorherzusagen. Hierfür sollen CNNs trainiert werden. Die CNNs bekommen als Input das seismische Signal und treffen durch Klassifizierung des Signals eine Vorhersage. Die verschiedenen Klassen spiegeln die zeitliche Entfernung zum nächsten Erdbeben wieder. Entscheidend sind hierbei der Input und der Aufbau des Netzwerks, um einen bestmöglichen Output zu bekommen. Die vorhandenen Daten sollen in Trainings-, Validierungs- und Testdaten aufgeteilt werden. Die trainierten und validierten CNN-Modelle sollen für die Testdaten eine Zuteilung in eine Klasse treffen. Für jedes Testsegment wird eine Vorhersage in eine Klasse getroffen. Diese werden mit den wahren Klassen verglichen und in einer Confusion Matrix dargestellt.

Das Projekt soll in Python in einem Jupyter Notebook programmiert und dokumentiert werden. Die CNNs werden mit Keras und TensorFlow im Backend aufgebaut.

3 Stand der Technik

Der Stand der Technik ist aufgeteilt in Erdbebenvorhersage und Convolutional Neural Networks. Im ersten Teil sind verwandte Arbeiten genannt, die mit Machine Learning Methoden eine zeitliche Vorhersage von Erdbeben treffen. Im zweiten Teil werden Arbeiten gelistet, die CNNs verwenden, um aus akustischen Signalen Vorhersagen und Klassifikationen zu treffen.

3.1 Erdbebenvorhersage

Ein Großteil der Erdbebenvorhersagen mit Machine Learning Methoden beziehen sich auf das *Wo* und *Wie stark* ein Erdbeben auftritt. Eine zeitliche Vorhersage ist in der

Literatur gering vorhanden. Die Arbeit von Rouet-Leduc et al. kommt diesem Projekt sehr nahe. Sie verwenden Machine Learning Methoden zur Vorhersage von Erdbeben. Das Team entscheidet sich bewusst gegen neuronale Netzwerke, da sie diese als eine Blackbox sehen. Sie verwenden stattdessen Random Forest, Decision Tree und verwandte Verfahren, wie z.B. Gradient Boosted Trees [4] [15]. Mit ihren Forschungen konnte das Team sehr gute Ergebnisse zur zeitlichen Erdbebenvorhersage erzielen. Aus den Daten werden statistische Features definiert, bevor diese in ein Modell gegeben werden. Das Extrahieren qualitativ hochwertiger Features ist ein entscheidender Schritt in ihrer Vorgehensweise [15].

3.2 Convolutional Neural Networks

In diesem Projekt werden verschiedene CNNs auf den Datensatz angewendet. Es soll dabei herausgefunden werden, welche Vorgehensweise am besten geeignet ist. Zuerst wird unterschieden, was dem Netzwerk als Input übergeben wird. Hier können sowohl unverarbeitete Daten, als auch vorverarbeitete Daten ins Netzwerk gegeben werden. Ein Beispiel für eine Arbeit, die unverarbeitete, akustische Daten in ein Netzwerk gibt, stammt von Dai et al. [1]. Hier werden Deep-CNN verwendet, um Informationen aus rohen Daten zu ziehen. Wichtig in der Arbeit ist die Reduktion der Parameter durch Pooling-Layer. Ein Großteil der Ansätze verwendet vorverarbeitete Daten, speziell bei großen Datenmengen.

Eine der häufigen Vorverarbeitungen bei akustischen Daten ist die Fourier-Transformation. Für diese sind verschiedene Anwendungen möglich. Im Bereich der akustischen Vorverarbeitung sind die Kurzzeit-Fourier-Transformation (STFT) [6] [16] und die Mel Frequency in Form von MFCCs [7] [13] von Bedeutung. Diese beiden Ansätze werden auf ihre Kompatibilität mit den hier verwendeten Daten geprüft. Die aufgelistete Literatur bezieht sich ausschließlich auf CNNs. Diese haben große Vorteile in

Bezug auf Zeitreihendaten und Bilddaten. Warum in diesem Projekt CNNs das Mittel der Wahl sind, wird in Kapitel 6 genauer erläutert.

4 Erdbeben-Datensatz

In diesem Kapitel wird aufgezeigt, um was für Daten es sich handelt. Die Daten stammen aus einem klassischen Labor-Erdbebenmodell (siehe Abbildung 1). Die zwei Gesteinsschichten werden gleichzeitig geschert, während sie einer konstanten Normalbelastung und Schergeschwindigkeit ausgesetzt sind. Durch die Verschiebung der Gesteinsschichten wird durch den Haft-Gleit-Effekt der Zyklus von Belastung und Versagen bei tektonischen Platten nachgeahmt. Obwohl das Experiment wesentlich einfacher ist als eine Verschiebung der tektonischen Platten, weist es viele physikalische Eigenschaften davon auf [9].

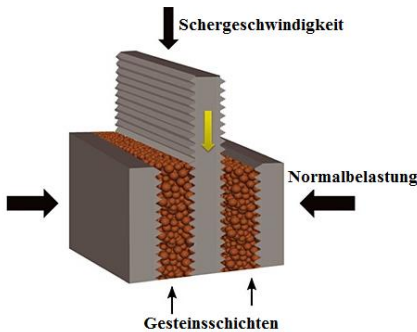


Abbildung 1: Labor-Erdbebenmodell [9]

Die seismischen Daten des Labor-Erdbebens werden mit einem piezokeramischen Sensor erfasst, der bei Verformung durch seismische Schwingungen eine Spannung ausgibt. Die verwendeten Daten sind die aufgezeichneten Spannungen [14]. Der Datensatz stammt von der Plattform *Kaggle.com* und ist Teil eines Wettbewerbs, ausgetragen vom Los Alamos National Laboratory [11].

5 Datenanalyse

In diesem Kapitel wird anhand einer Analyse aufgezeigt, wie diese Daten aussehen. Es handelt sich bei dem Datensatz um eine ca.

10 Gigabyte große CSV-Datei. Diese Datei besteht aus 629.145.481 Zeilen und 2 Spalten. In der ersten Spalte steht das aufgenommene seismische Signal in ganzen Zahlen. Hierbei handelt es sich um die Spannungen, welche der piezokeramische Sensor ausgibt. In der zweiten Spalte sind die Zeitangaben in Sekunden, bis ein Erdbeben im Datensatz auftritt. Diese sind als Kommazahlen mit zweistelligen Nachkommastellen eingetragen. Die Zeit dient als Label für das seismische Signal.

Im kompletten Datensatz sind 16 Labor-Erdbeben aufgezeichnet, welche in Abbildung 2 dargestellt sind. Die x-Achse stellt die Zeitachse dar. Der obere Graph zeigt die Zeit bis zum nächsten Erdbeben. Der untere Graph stellt das seismische Signal dar. Wenn der obere Graph parallel zur y-Achse steht, ist ein Erdbeben im Datensatz aufgetreten. Im unteren Graph zeigen sich an diesen Stellen starke Peaks. Wohingegen nicht jeder Peak im unteren Graph automatisch ein Erdbeben ist. An den Maxima des oberen Graphen beginnt die Zeit wieder von vorne zu zählen, bis wieder ein Erdbeben auftritt. Das seismische Signal hat visuell die Form eines akustischen Signals.

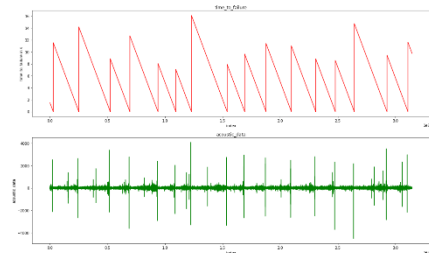


Abbildung 2: Gesamtübersicht Datensatz

Der kürzeste zeitliche Abstand zwischen zwei Erdbeben ist ca. 7,1 s. Der längste Abstand beträgt ca. 16,1 s. Es handelt sich um einen sehr hochaufgelösten Datensatz. Eine Sekunde enthält ca. 3,85 Mio. Zeilen im Datensatz und jede Zeile enthält eine seismische sowie eine zeitliche Information. Durch die hohe Auflösung des Datensatzes wird angenommen, dass die eigentlich sehr kurze Zeit

des kompletten Datensatz keinen negativen Einfluss auf die Ergebnisse hat.

Bei näherer Betrachtung lassen sich Auffälligkeiten feststellen. Der Graph mit dem zeitlichen Label sollte direkt nach Eintreten eines Erdbebens wieder einen hohen Wert annehmen, der die Zeit bis zum nächsten Erdbeben angibt. Es zeigt sich, dass eine Zeitverzögerung zwischen den zwei Signalen von ca. 0,3 s vorliegt (siehe Abbildung 3). Der Bereich zwischen Erdbeben und vertikalem Sprung im Zeitsignal müsste eigentlich zur Vorhersage des nächsten Erdbebens mitgenutzt werden. Somit ist dieser Bereich falsch gelabelt. Eine negative Auswirkung hiervon muss ausgeschlossen werden.

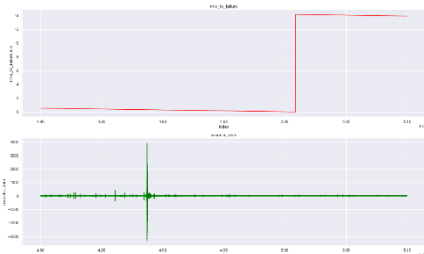


Abbildung 3: Zeitverzögerung Signale

Das seismische Signal ist ein kontinuierliches Signal und sieht auch bei näherer Betrachtung aus wie ein akustisches Signal (siehe Abbildung 4). Es wird geprüft, ob das Signal wie ein akustisches Signal behandelt werden kann. Sowohl seismische Signale, als auch akustische Signale sind Schwingungen.

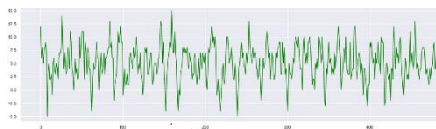


Abbildung 4: Seismisches Signal

Der Abbildung 5 ist zu entnehmen, dass sich die Zeit bis zum Erdbeben schrittweise verringert. Der Graph ist nicht linear, wie es noch auf Abbildung 2 aussah. Eine „Stufe“ im Graph entspricht ca. 5 ms.

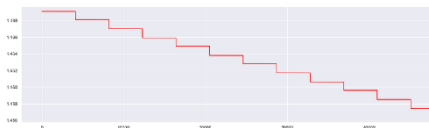


Abbildung 5: Zeit bis Erdbeben

6 Umsetzung

Beide Ansätze werden mit einem CNN umgesetzt. CNNs werden zur Datenverarbeitung verwendet, die eine bekannte, gitterartige Topologie und dadurch ein räumliches Muster haben. Das sind zum einen Zeitreihendaten und zum anderen Bilddaten. Bei den Originaldaten handelt es sich um Zeitreihendaten, die vorverarbeiteten Daten werden als Bilddaten betrachtet, da die Vorverarbeitung Spektrogramme ausgibt. Für beide Anwendungsmöglichkeiten konnten mit CNNs in den letzten Jahren sehr gute Ergebnisse erzielt werden [3].

Entscheidend sind der Input und der Aufbau des CNNs. Beide hier beschriebenen Ansätze haben die gleiche Vorbereitung der Daten, die folgend beschrieben wird.

6.1 Allgemeine Vorbereitung

Für eine schnellere Verarbeitung der Daten wird der 10 GB große Datensatz zuerst komprimiert. Hierfür werden die seismischen Daten im komprimierten NPZ-Format gespeichert und von den Zeitdaten nur noch die Maxima und Minima (siehe Abbildung 2) statt der einzelnen Zeitdaten. Beim Aufruf der Daten werden lineare Funktionen zwischen den Maxima und den Minima generiert. Der größte Unterschied zwischen der Approximation und den Originaldaten liegt bei 1,15 ms, was vernachlässigbar ist. Dadurch konnten die Daten auf ca. 400 MB komprimiert werden und dadurch über 20 Mal so schnell geladen werden [2].

Nachdem der Datensatz geladen wurde, wird dieser in Abschnitte unterteilt. Jeder Abschnitt enthält ein Erdbeben. An jeder Stelle an dem das Zeitsignal ein Minimum hat, wird der Datensatz getrennt. Folgend sind 16 Abschnitte vorhanden mit jeweils einem

Erdbeben. Die Abschnitte sind unterschiedlich groß, wie in Abbildung 6 zu sehen ist. Auf der y-Achse ist die Anzahl der Daten pro Abschnitt zu sehen. Wichtig ist zu beachten, dass die Zahlenwerte der y-Achse mit 10 Mio. multipliziert werden müssen.

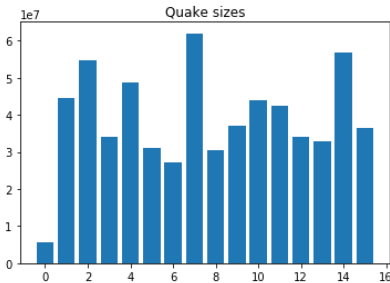


Abbildung 6: Erdbeben-Abschnitte

Anschließend werden die Abschnitte in Trainings-, Validierungs- und Testdaten unterteilt (siehe Tabelle 1).

Tabelle 1: Verteilung Abschnitte

Verwendung	Gesamtanteil
Training	66 %
Validierung	17 %
Test	17 %

Alle Daten werden in einzelne Batches gespalten, die jeweils aus 15.000 Reihen bestehen. Auf diese Batches werden die beiden folgenden Verfahren angewendet. Jeder Batch wird mit einer einzelnen Zeit gelabelt. Hierfür wird die Zeit der letzten Reihe im Batch genommen.

6.2 Originaldaten

Beim ersten Ansatz werden die Originaldaten in das aufgebaute CNN gegeben. Dieses muss alle Features aus den Daten selbst lernen. Alle Daten werden in zwei Kategorien aus X-Daten und y-Daten unterteilt. Die X-Daten enthalten die seismischen Daten, der Input des CNN. Die y-Daten enthalten die Zieldaten/Label-Daten, in diesem Fall die Zeit bis zum nächsten Erdbeben. Die X-Daten werden zu einem 3D Tensor konvertiert. In diesem Stecken die Anzahl der Batches, die Größe der einzelnen Batches und die

Channels. Der 3D Tensor für das Training hat das Format: (19493, 15000, 1). Da es sich um ein 1D-Zeitsignal handelt, hat der Tensor nur einen Channel. Die Zieldaten werden in Klassen eingeteilt, für die das Netzwerk Vorhersagen treffen soll. Diese haben für das Training das Format: (19493, 6). Die Anzahl der Batches ist dieselbe und die 6 gibt die Anzahl der Klassen an. Die Validierungs- und Testdaten haben dasselbe Format, einzig die Anzahl der Batches ist geringer. Mit den korrekt formatierten Daten kann folgend das CNN trainiert werden.

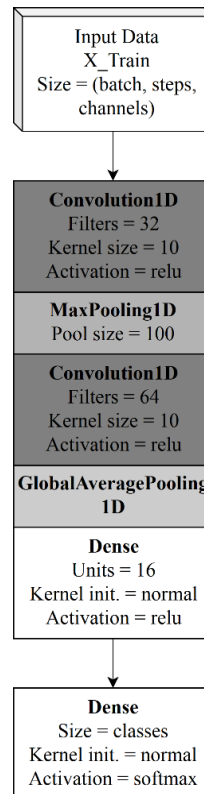


Abbildung 7: CNN Originaldaten

In Abbildung 7 ist das CNN dargestellt. Hervorzuheben ist, dass alle Faltungs- und Pooling-Layer eindimensional rechnen. Der Input Layer ist ein 1D-Faltungs-Layer. Dieser enthält 32 Filter über welche gefaltet

wird. Die Kernel size beträgt 10 und gibt die Länge des Faltungsfensters an. Der erste Hidden Layer ist ein 1D-MaxPooling-Layer, welcher aus 100 Elementen das Maximum nimmt. Dies verringert die Parameter des Outputs um den Faktor 100. Es folgen ein 1D-Faltungs-Layer und ein 1D-GlobalAveragePooling-Layer, welcher wieder die Parameter im Modell reduziert und zusätzlich das Overfitting minimiert. Abschließend folgen zwei Dense-Layer, welche gewöhnlichen Fully-Connected-Layer entsprechen. Diese haben eine normalverteilte Kernel Initialisierung. Hierdurch wird die Art und Weise definiert, wie die anfänglichen Zufallsgewichte des Layers eingestellt werden. Als Aktivierungsfunktion werden relu und beim letzten Layer softmax verwendet. Der zweite Dense-Layer ordnet die Daten in die Zielklassen zu [5].

6.3 Vorverarbeitung

Beim zweiten Ansatz wird der Input des Netzwerks vorverarbeitet. Jedes akustische Signal kann mit einer Fourier-Transformation in seine einzelnen Frequenzen transformiert werden. Es wird eine Frequenzanalyse durchgeführt, die das Signal in seinen einzelnen Frequenzen beschreibt [12]. Dadurch können wertvolle Informationen aus dem Signal gewonnen werden. Wie im Kapitel 3 *Stand der Technik* beschrieben wird, sind häufige Anwendungen der Fourier-Transformation die Kurzzeit-Fourier-Transformation und die Mel Frequency. Durch eine Reihe an Tests hat sich herausgestellt, dass die Mel Frequency Cepstral Coefficients (MFCC) für diese Anwendung die besten Ergebnisse liefert. Auf jeden Batch (15.000 Reihen) werden die MFCCs angewendet. Die Berechnung der MFCCs besteht aus den folgenden fünf Hauptschritten:

1. Signal in Abschnitte unterteilen
2. Amplitudenspektrum erzeugen
3. Logarithmieren
4. In Mel Spektrum umwandeln
5. Diskrete Fourier Transformation [8]

Durch die Berechnung ergeben sich für jeden Batch ein MFCC-Spektrogramm (siehe Abbildung 8). Zur besseren Anschaulichkeit wird das MFCC Spektrogramm in ein dB-skaliertes Spektrogramm umgewandelt.

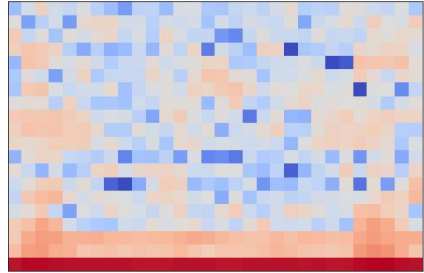


Abbildung 8: MFCC Spektrogramm

Jedes Spektrogramm wird mit einer Zeit gelabelt. Die X-Daten werden dieses Mal zu einem 4D Tensor konvertiert. Dieser besteht aus der Anzahl der Batches, den Reihen, den Spalten und den Channels. Die Anzahl der Batches und der Channels bleibt gleich, wie im vorherigen Kapitel. Das Spektrogramm beinhaltet auf dem einen Channel die Intensitäten, welche auf Abbildung 8 farblich dargestellt werden. Die zusätzliche Dimension ergibt sich durch die Transformierung des 1D-Zeitsignals in ein 2D-Spektrogramm. Für das Training ergibt sich ein Tensor vom Format: (19493, 20, 30, 1). Die Zieldaten werden in Klassen eingeteilt, was wieder ein Format von (19493, 6) ergibt. Mit diesen vorverarbeiteten Daten wird das CNN trainiert.

Der Aufbau des CNN ist in Abbildung 9 gezeigt. Es werden 2D-Faltungs- und Pooling-Layer verwendet. Nach zwei Faltungs-Layer folgt ein MaxPooling-Layer. Die Faltungs-Layer haben jeweils 32 Filter, eine Kernel size von 3x3 und eine relu-Aktivierungsfunktion. Der erste Faltungs-Layer hat zusätzlich ein Padding. Dieses sorgt dafür, dass Output und Input die gleiche Länge haben. Der darauffolgende Dropout-Layer setzt bei jedem Durchlauf des Netzwerks 25 % zufällig gewählte Einheiten auf null, um ein Overfitting zu verhindern.

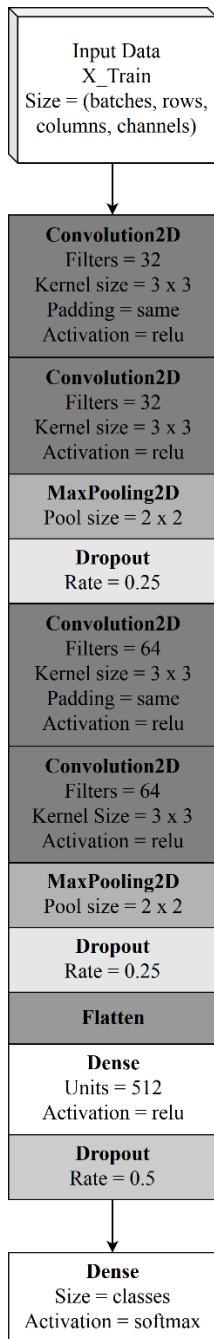


Abbildung 9: CNN MFCC

Darauf folgt die gleiche Abfolge von Layern noch einmal. Ausschließlich die Filteranzahl der Faltungs-Layer wird verdoppelt. Danach folgt ein Flatten-Layer, welcher den Input ebnet. Der 3D Input des Layers wird zu einem 1D Output. Abschließend folgen ein Dense-Layer, ein erhöhter Dropout und noch ein Dense-Layer, welcher den Input in die endgültigen Klassen einteilt [5].

6.4 Training

Nachdem die CNNs aufgebaut wurden, können diese trainiert werden. Der Output der CNNs beinhaltet die Wahrscheinlichkeiten, mit welcher der Input in der jeweiligen Klasse liegt. Beide beschriebenen Vorgehensweisen werden auf dieselbe Weise kompiliert und gefittet.

Für das Kompilieren mit Keras muss ein Optimierer, eine Loss-Funktion und eine Metrik angegeben werden. Als Optimierer wird Adam gewählt, ein verbreiteter, stochastischer Optimierer. Als Loss-Funktion wird die kategorische Kreuzentropie gewählt, welche für Klassifikationsaufgaben empfohlen wird. Das Ziel des Modells ist es, die Loss-Funktion zu minimieren. Abschließend wird die Metrik des Trainings auf die Genauigkeit gesetzt, was ebenfalls für Klassifikationsaufgaben empfohlen wird.

Beim Fitten wird angegeben, mit welchen Daten trainiert und validiert wird. Zusätzlich werden eine Batch size und die Epochen des Trainings angegeben. Wichtig beim Fitten ist das Einfügen von Callbacks. Der erste Callback sorgt dafür, dass das Training abbricht, wenn sich der Loss der Validierungsdaten (val_loss) nach 10 Epochen nicht verbessert hat. Der zweite Callback speichert das Modell mit dem geringsten val_loss ab. Der letzte Callback verringert die Lernrate um ein Zehntel, wenn sich der val_loss nach 5 Epochen nicht verringert hat [5].

7 Ergebnisse und Diskussion

Bei beiden Verfahren wurde der seismische Input in sechs Klassen eingeteilt. Die Klassen können als Gefahrenstufen gesehen werden. Je niedriger die Zahl der Klasse, desto höher

die Gefahr. In Klasse 0 weist das seismische Signal bereits ein Erdbeben auf oder es steht kurz bevor. Die besten quantitativen Ergebnisse werden als Diagramme für die Genauigkeit und den Loss der Trainings- und Validierungsdaten dargestellt. Hier konnte mit dem CNN in Kombination mit den MFCCs die besten Ergebnisse erzielt werden. Aufschlussreicher ist die Darstellung in einer Confusion Matrix, welche anzeigt in welche Klasse die Batches einsortiert wurden (predicted label) und in welche sie eigentlich hätten einsortiert werden sollen (true label). Die Confusion Matrizen zeigen anschaulich, wie die CNNs abgeschnitten haben.

7.1 Originaldaten

Das CNN mit den Originaldaten konnte höchstens eine Validierungs-Genauigkeit von ca. 40 % erreichen. Wohingegen mit den Testdaten gute Ergebnisse in der Confusion Matrix erzielt werden konnte (siehe Abbildung 10). Bei einer perfekten Klassifikation liegen die Batches alle auf der Diagonalen von links oben nach rechts unten. Eine perfekte Klassifikation liegt nicht vor, doch es zeigt sich ein klarer Trend von links oben nach rechts unten.

Es lassen sich drei Beobachtungen in der Confusion Matrix machen. Erstens werden die Batches eher näher an das Erdbeben klassifiziert, als sie in Wirklichkeit sind. Dies führt zu dem breiten Streifen unter der Diagonalen. Für den Anwendungsfall dieses Projektes ist das von Vorteil, da eine „vorsichtiger“ Vorhersage getroffen werden kann, die eher früher eine Erdbebenwarnung ausgibt. Zweitens werden generell keine Vorhersagen in hohe Klassen getroffen, was sich in den leeren Spalten vier und fünf zeigt. Sowohl die erste, als auch die zweite Beobachtung könnten ähnliche Ursachen haben. Bei der Klassifikation wird nach der Übereinstimmung von Merkmalen der Test- und Trainingsdaten in Klassen eingeteilt. Es wird davon ausgegangen, dass näher am Erdbeben liegende Batches markantere Merkmale haben. Werden Testdaten in das Netzwerk gegeben, werden diese möglicherweise in

Klassen eingeteilt, die näher am Erdbeben liegen, da hier bestimmte Merkmale gefunden wurden. Ein weiterer wichtiger Faktor für eine Einstufung in niedrigere Klassen ist die vergrößerte Anzahl an Trainingsdaten in diesem Bereich. Die Erdbeben-Abschnitte sind unterschiedlich lang (siehe Abbildung 6), d.h. für Bereiche, die weiter weg vom Erdbeben sind, gibt es weniger Daten. Für diesen Bereich ist das Training des Netzwerks ungenügend. Die dritte Beobachtung zeigt sich in der Zeile 0. Hier wurden zu viele Batches in zu hohe Klassen eingeteilt. Dies liegt zu einem großen Teil an dem falschen Label/dem Delay zwischen den Signalen (siehe Kapitel 5). Hier werden die Batches direkt nach dem Erdbeben als Batches vor dem Erdbeben gelabelt, obwohl das Erdbeben bereits vorbei ist.

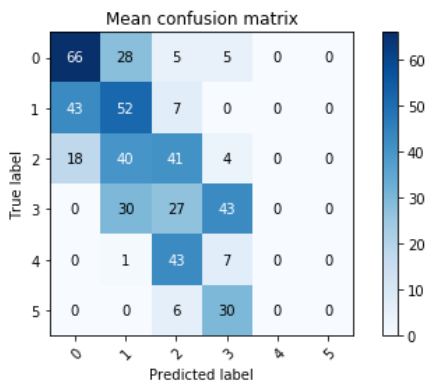


Abbildung 10: Originaldaten

Bei beiden Confusion Matrizen handelt es sich um eine Mean Confusion Matrix. Durch diesen Ansatz konnten bessere Ergebnisse erzielt werden als ohne einen Mean zu berechnen. Die Netzwerke werden mit kleineren Batches trainiert. Nach der Klassifikation der Batches wird jeweils der Mean-Wert aus 10 Batches berechnet. Dadurch können Ausreiser eliminiert und die Ergebnisse normalisiert werden. Der Ansatz liefert robustere Ergebnisse. Es wird mit Batches aus 15.000 Reihen trainiert und durch

die Berechnung des Mean-Werts eine Vorhersage für jeden Batch aus 150.000 Reihen getroffen. Dies entspricht ca. 39 ms.

7.2 Vorverarbeitung

Das CNN mit der MFCC-Vorverarbeitung konnte eine höhere Validierungs-Genauigkeit und einen geringeren Validierungs-Loss erzielen.

In Abbildung 11 zeigt sich, dass nach knapp 200 Epochen eine Validierungs-Genauigkeit von ca. 65 % erreicht wurde. Der Validierungs-Loss hat sich bei ca. 1,1 nicht mehr verbessert, weswegen das Training durch den Callback abgebrochen wurde (siehe Abbildung 12). Bei beiden Diagrammen zeigt sich am Ende des Trainings ein Plateau, ab welchem sich die Ergebnisse nicht mehr verbessern. Die gestrichelte Linie zeigt die Validierungsdaten, während die durchgezogene Linie die Trainingsdaten darstellt.

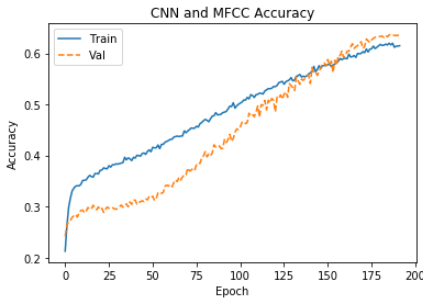


Abbildung 11: CNN und MFCC Modell Genauigkeit

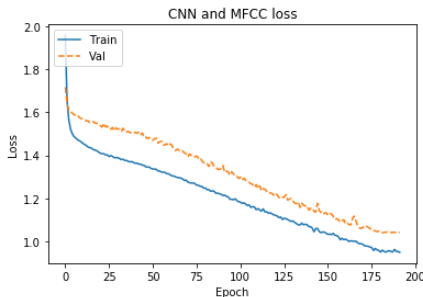


Abbildung 12: CNN und MFCC Modell Loss

Die Confusion Matrix weist eine sehr ähnliche Verteilung wie bei den Originaldaten auf (siehe Abbildung 13). Auch hier zeigen sich die im vorherigen Kapitel beschriebenen Beobachtungen. Die Ergebnisse in der Confusion Matrix unterscheiden sich jeweils nur um wenige Batches.

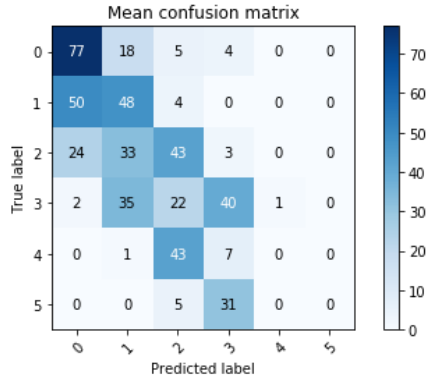


Abbildung 13: MFCC

8 Fazit und Ausblick

Die entwickelten Ansätze haben das Potenzial für eine zeitliche Vorhersage von Erdbeben und sind eine neue Herangehensweise an das Problem. Es zeigt sich, dass die seismischen Signale wie akustische Signale behandelt werden können. Mit beiden Ansätzen konnten bereits gute Ergebnisse erzielt werden. Verbesserungen könnten durch ändern der Parameter, hinzufügen von Layern oder Anpassung der Daten erreicht werden. Das Projekt wurde im Rahmen der Wissenschaftlichen Vertiefung an der Hochschule Reutlingen umgesetzt.

Als Ausblick wäre eine Anwendung der trainierten Modelle auf echte Erdbeben Daten interessant. Hier würde sich zeigen, wie gut die Modelle und die Daten der Labor-Erdbeben sind. Zusätzlich wäre ein Vergleich mit anderen Machine Learning Methoden sinnvoll, wie z.B. Random Forest oder Decision Tree.

9 References

- [1] Dai, W., Dai, C., Qu, S., Li, J., and Das, S. 2017. Very Deep Convolutional Neural Networks for raw waveforms. *ICASSP 2017*.
- [2] F., M. 2019. *How to reduce the training data to 400MB*. <https://www.kaggle.com/friedchips/how-to-reduce-the-training-data-to-400mb>. Accessed 16 April 2019.
- [3] Goodfellow, I., Bengio, Y., and Courville, A. 2016. *Deep Learning*. MIT Press.
- [4] Hulbert, C., Rouet-Leduc, B., Johnson, P. A., Ren, C. X., Rivière, J., Bolton, D. C., and Marone, C. 2019. Similarity of fast and slow earthquakes illuminated by machine learning. *Nature Geosci*, 69–74.
- [5] Keras. 2019. *Keras Documentation*. <https://keras.io/>. Accessed 17 April 2019.
- [6] Kong, M., Zhang, J., Liu, W., and Zhang, G. 2018. Radar Emitter Identification Based on Deep Convolutional Neural Network. *ICCAIS 2018*, 309–314.
- [7] Li, S., Kim, H., Lee, S., Gallagher, J. C., Kim, D., Park, S., and Matson, E. T. 2018. Convolutional Neural Networks for Analyzing Unmanned Aerial Vehicles Sound. *ICCAS 2018*.
- [8] Logan, B. 2000. Mel Frequency Cepstral Coefficients for Music Modeling.
- [9] Los Alamos National Laboratory. 2019. *LANL - Additional Information*. <https://www.kaggle.com/c/LANL-Earthquake-Prediction>. Accessed 17 March 2019.
- [10] Los Alamos National Laboratory. 2019. *LANL - Description*. <https://www.kaggle.com/c/LANL-Earthquake-Prediction>. Accessed 17 March 2019.
- [11] Los Alamos National Laboratory. 2019. *LANL Earthquake Prediction*. <https://www.kaggle.com/c/LANL-Earthquake-Prediction>. Accessed 16 April 2019.
- [12] Meyer, M. 2017. *Signalverarbeitung*. Springer Fachmedien Wiesbaden, Wiesbaden.
- [13] Prakruthi, U. S., Kiran, D., and Ramasangu, H. 2018. High performance neural network based acoustic scene classification ICISC 2018, 781–784.
- [14] Rouet-Leduc, B. 2019. *LANL - Discussion (Additional Information)*. <https://www.kaggle.com/c/LANL-Earthquake-Prediction/discussion/77526>. Accessed 17 March 2019.
- [15] Rouet-Leduc, B., Hulbert, C., and Johnson, P. A. 2019. Continuous chatter of the Cascadia subduction zone revealed by machine learning. *Nature Geosci*, 75–79.
- [16] Seo, Y., Jang, B., and Im, S. 2018. Drone Detection Using Convolutional Neural Networks with Acoustic STFT Features.
- [17] Statista. 2019. *The 10 biggest natural disasters worldwide by economic damage since 1980*. <https://www.statista.com/statistics/268126/biggest-natural-disasters-by-economic-damage-since-1980/>. Accessed 17 March 2019.
- [18] Statista. 2019. *The 10 most significant natural disasters worldwide by death toll since 1980*. <https://www.statista.com/statistics/268029/natural-disasters-by-death-toll-since-1980/>. Accessed 17 March 2019.

Anforderungen an ein Gamification-Konzept zur Motivationssteigerung der Anwender eines KI-Service zur Maschinenoptimierung

Tanja Brodbeck

Reutlingen University

Tanja.Brodbeck@Student.Reutlingen-University.DE

Abstract

In Folge der gegenwärtigen Digitalisierung in der produzierenden Industrie werden Anwendungen oder Services mit potentiell positiven Auswirkungen auf Faktoren wie Effektivität und Arbeitsqualität entwickelt. Ein geeigneter Ansatz zur Stärkung motivierender Aspekte im Arbeitskontext kann *Gamification* darstellen. In dieser Arbeit ist die initiale Konzeption und Evaluation eines *Gamification*-Ansatzes für Anwender eines KI-Service zur Maschinenoptimierung dargestellt und möglichen Anforderungen an ein Konzept zur Motivationssteigerung extrahiert.

Schlüsselwörter

Gamification, Motivation, Digitalisierung, Industrie 4.0, Produzierende Industrie, KI, Prozessoptimierung

CR-Kategorien

Human-centered computing~Interaction design process and methods • Human-centered computing~User studies • Applied computing~Industry and manufacturing

Betreuerin Hochschule: Prof. Dr. Tullius
Hochschule Reutlingen
Gabriela.Tullius@Reutlingen-
University.de

Betreuer Institut: Felix Georg Müller
Fraunhofer IPA
felix.mueller@ipa.fraunhofer.de

Informatics Inside
Wissenschaftliche Vertiefungskonferenz
08. Mai 2019, Hochschule Reutlingen
Copyright 2019 Tanja Brodbeck

1 Einleitung

Das behandelte Projekt des Fraunhofer-Instituts für Produktionstechnik und Automatisierung (IPA) betrifft in erster Linie die Overall Equipment Effectiveness (OEE) von hochautomatisierten Einzelmaschinen wie z.B. Spritzgießmaschinen. Das Fraunhofer IPA kooperiert mit Maschinenherstellern und -betreibern, um einen auf Künstlicher Intelligenz (KI) basierten Software-Service zu entwickeln, der den OEE-Wert von Stand-alone-Maschinen in Echtzeit messen und zugleich die technischen Ursachen erklären kann. Die Kennzahl OEE wird von den Faktoren Verfügbarkeit, Leistung und Qualität beeinflusst. Ist diese aggregierte Kennzahl und die einzelnen Verlustursachen, die diese bedingen bekannt, können Optimierungen in den Parametern und Einstellungen der einzelnen Maschinen vorgenommen werden. Ziel ist es Abweichungen zu eliminieren und somit den erlernten idealen Prozess in den realen Maschinen abzubilden.

Zur automatisierten Ursachenfindung von Effizienz-Abweichungen soll die zu entwickelnde Software auch die Subprozesse im Herstellungsprozess von jeder einzelnen Maschine abbilden und diese mit anderen Maschinen vergleichen. So lassen sich die notwendigen Optimierungen der Parameter in kürzerer Zeit und technisch präziser auf einzelne Maschinen und Komponenten zurückführen.

1.1 Stakeholder / Zielgruppe

Als Hauptzielgruppen des Service sind Prozessoptimierer und Maschineneinrichter zu nennen. Sie sind an sämtlichen oder an einem Teilbereich der Daten und den daraus resultierenden Interpretationen und empfohlenen Aktionen des Service interessiert. Weitere Stakeholder stellen Produktions- und Werksleiter dar, die vermehrt Interesse an der Methodik des Software-Produkts und den wirtschaftlichen Effekten des Einsatzes zeigen.

1.2 Arbeitsabläufe von Prozessoptimierern

Die derzeitigen Arbeitsabläufe eines Prozessoptimierers können wie folgt beschrieben werden: Zunächst erfolgt eine manuelle Auswertung der Betriebsdaten einzelner Maschinen, sodass gut laufende Maschinen bestimmt und zur näheren Analyse herangezogen werden können. Zur Erfassung der Zykluszeit wird die Zeit zwischen Start und darauffolgendem erneuten Start eines Prozesses gemessen. Auf diese Weise wird eine Rangfolge der Maschinen bzgl. Optimierungspotential auf einer technisch detaillierten Ebene gebildet. Zusammen mit einem Steuerungstechniker eruiert der Prozessoptimierer mögliche Gründe für die Abweichungen und vergleicht manuell die Parameter über das Interface der speicherprogrammierbaren Steuerung an einzelnen Maschinen. Daraufhin erfolgt eine Anpassung der Parameter.

Die möglichen, zukünftigen Arbeitsschritte bei Nutzung des KI-Service zur Maschinenoptimierung sind wie folgt zu beschreiben: Die Berechnung der Zykluszeit erfolgt automatisiert, kontinuierlich bei allen Maschinen. Große Diskrepanzen werden innerhalb des Service ausgewiesen und direkt technischen Ursachen zugeordnet, eine manuelle Erfassung ist nicht mehr notwendig. Zudem ist ein zusätzliches Detail-Level vorhanden, bei dem diejenigen Subschritte mit den stärksten Abweichungen einsehbar sind. Diese Informationen können dem Steuerungstechniker zur Plausibilitätsprüfung vorgelegt werden. Daraufhin erfolgt die Anpassung der Parameter pro Maschine.

1.3 Problemstellung

Unter Verwendung des KI-Service zur Maschinenoptimierung erfolgen einige zuvor manuell durchgeführten Arbeitsschritte automatisiert. Anhand von Nutzerbefragungen kann davon ausgegangen werden, dass diese Erleichterung der Arbeit einen essentiellen Motivationsfaktor für die Anwender darstellt. In zukünftigen Entwicklungsstadien des KI-Service zur Maschinenoptimierung sollen ähnliche Produktionen automatisiert miteinander verglichen werden, sodass durch lernende Algorithmen mögliche Störquellen erkannt und dazu passende Maßnahmen empfohlen werden können. Hierfür ist regelmäßiger Input seitens der Anwender des KI-Service zur Maschinenoptimierung essentiell, um anhand realer Industriedaten den Service kontinuierlich weiterlernen zu lassen. Vor diesem Hintergrund ist die tägliche Nutzung essentiell, allerdings ließ sich aus den ersten Nutzerbefragungen die Tendenz zu einer weniger häufigen Nutzungsbereitschaft erkennen. Die mit Hilfe eines *Gamification*-Ansatzes zu optimierende Problemstellung stellt die Motivation zur täglichen bzw. häufigen Nutzung des KI-Service zur Maschinenoptimierung dar.

2 Stand der Forschung

2.1 Gamification

Der Begriff *Gamification* beschreibt die Verwendung von Design-Elementen, die charakteristisch für (Video-)Spiele sind, außerhalb des Kontexts von Spielen, um die Motivation und Aktivität von Nutzern einer Anwendung zu steigern [4].

In [4] wurden aus der Forschungsliteratur im Bereich *Gamification* Definitionen für Spielelemente, –mechaniken und –prinzipien extrahiert. Die Verwendung dieser Begriffe in den folgenden Abschnitten orientiert sich an den nachstehenden Erkenntnissen aus [4]:

Interface Design Pattern aus Spielen: Erfolgreiche Interaktionskomponenten und Designlösungen für bekannte Probleme, z.B. Badges oder Auszeichnungen, Leaderboard, Level. Im Folgenden als *Elemente* bezeichnet.

Spielmechaniken: Häufige, wiederkehrende Elemente des Designs eines Spiels, die das Gameplay betreffen, z.B. Zeitdruck und begrenzte Ressourcen.

2.2 Motivationspsychologie

Als das hauptsächliche Anliegen von *Gamification* kann die Unterstützung bzw. das Hervorrufen von Motivation genannt werden. Innerhalb der Literatur dieses Bereichs wird weitgehend zwischen intrinsischer und extrinsischer Motivation unterschieden. In [8] wird beschrieben, dass diese dichotome Unterscheidung nicht ausreicht, um alle Schnittstellen zu beschreiben, an denen *Gamification*-Ansätze motivieren können. Das in [8] beschriebene, ausführliche Modell ist im Folgenden skizziert:

Interne Prozessmotivation: Die Freude an der Handlung selbst.

Internes Selbstverständnis: Streben und Handeln nach eigenem Idealbild, gebildet aufgrund eigener Werte und Ansichten.

Instrumentelle Motivation: Ausführen einer Handlung mit Aussicht auf eine Belohnung (Bezahlung, Beförderung oder andere Boni).

Externes Selbstverständnis: Handeln entsprechend der Erwartungen anderer Personen oder Gruppen.

Internalisierung von Zielen: Die Übernahme von Zielen einer Gruppe und das Streben nach diesem gemeinsamen Ziel, insofern diese mit dem eigenen Wertesystem übereinstimmen.

2.3 Spielertypen

Ein wichtiger Ansatz zur Wahl der passenden Spielelemente für die Nutzer bzw. Zielgruppe stellen psychologische Aspekte der Persönlichkeit (während des Spielens) dar.

In der Literatur im Bereich *Gamification* wird insbesondere die Klassifizierung von Spielertypen nach Bartle [1] genannt. Hierbei wird zwischen vier verschiedenen Typen unterschieden, die jeweils auf verschiedene Spielmechaniken und Motivationsfaktoren ansprechen [8]:

- (1) *Achiever* möchten möglichst viel in einem Spiel erreichen. Sie sprechen auf Elemente wie Punkte, Auszeichnungen und sammelbare Gegenstände an (*instrumentellen Motivation* [8]).
- (2) *Explorer* möchten das Spiel und dessen Spielwelt erkunden/entdecken, u.U. können sie sich Anerkennung für Entdeckungen erhoffen (*externes Selbstverständnis* [8]).
- (3) *Socializer* legen den Fokus auf soziale Interaktion mit anderen Spielern und können z.B. durch den Status in einer Gruppe beeinflusst werden (*externes Selbstverständnis* [8]).
- (4) *Killer* möchten andere Spieler oder Charaktere besiegen und Bewunderung / Respekt erhalten (*externes Selbstverständnis* [8]).

2.4 Wirkungsweisen von Gamification

Das Potential von *Gamification* in unterschiedlichen Anwendungsgebieten wird nach [21] zukünftig vermehrt diskutiert werden, Publikationen im Kontext der (produzierenden) Industrie stellen vergleichsweise jedoch eine Minderheit dar. Dennoch ließ sich innerhalb der letzten fünf Jahre eine ansteigende Relevanz dieses Anwendungsgebiets erkennen, z.B. anhand von Publikationen wie [13] und [14]. Diese Entwicklung führen die Autoren von [21] auf die Arbeitsaktivitäten in der produzierenden Industrie oder Logistik zurück, die u.U. monoton aufgrund ihrer stark standardisierten und repetitiven Natur wirken können. *Gamification* könnte potentiell die von Angestellten erfahrene Monotonie aufbrechen, die Motivation und damit die Produktivität stärken. Zudem bestehen Problempunkte wie die Kosteneffizienz von Automatisierungen, bei denen *Gamification*-Ansätze einen (Teil-)Aspekt zur Lösung darstellen können [12]. Mit der zunehmenden Digitalisierung in Industriebetrieben im letzten Jahrzehnt und der vermehrten Verwen-

derung von Sensoren sind Daten für *Gamification*-Konzepte zudem zugänglicher geworden [21] [22].

Gegenwärtig existieren erfolgreiche Ansätze zur Anwendung von *Gamification*, anhand von [2] kann eine hauptsächlich positive Auswirkung von *Gamification* seitens Forschung jedoch nicht bestätigt werden. Dies wird auf anfängliche, hohe Erwartungen zurückgeführt und der anschließenden Enttäuschung aufgrund häufig fehlkonzipierter Ansätze, die für den intendierten Zweck unzureichend sind [2]. Die Entwicklung von *Gamification*-Ansätzen ist einigen Fehlerquellen ausgesetzt, bspw. dem fehlenden Wissen über die Nutzer und deren Motivationsfaktoren, u.a. aufgrund der in [10] evaluierten Einflussfaktoren von Alter und Geschlecht in Bezug auf die wahrgenommenen Vorteile von *Gamification*, oder auch der Integration von Elementen wie Punkte und Level, obwohl diese für lediglich einen Spielertyp (*Achiever*) positiv fungieren [2].

3 Methoden und iterativer Design-Prozess

Zur Minderung der möglichen Fehlerquellen wie in 2.1.4. beschrieben orientierte sich die Konzeptionsphase an Methoden des Player-Centered Design bzw. dem Gamification Design Model insbesondere nach [6], die bei der Erfassung der Bedürfnisse und Motivationsfaktoren der Nutzer unterstützen.

3.1 Analyse-Phase

3.1.1 Mechanikkarten

Die Methode *Mechanikkarten* nach [7] soll einen initialen Zugang zur Zielgruppe und deren Anforderungen an einen *Gamification*-Ansatz zur Steigerung der Motivation ermöglichen. Die von *Playful Interactions Concepts* bereitgestellten *Mechanikkarten* [7] stellen die Basis eines Interviews mit einem Projektmitarbeiter dar, der zuvor als Prozessoptimierer arbeitete. Der Einsatz der *Mechanikkarten* fand wie seitens *Playful Interactions Concepts* empfohlen in der Analyse-Phase zur Ideenfindung statt.

3.1.2 Workshop: Nutzungskontext und Zielgruppe

Ein initialer Workshop fand zusammen mit externen UX-Designern und einem Projektmitarbeiter mit Praxiserfahrung als Prozessoptimierer statt. Ziel war es hierbei den Nutzungskontext des Service zu eruieren und einen vorläufigen Spielertyp zu definieren, der als Basis für erste *Gamification*-Konzepte dienen soll. Im Rahmen des Workshops ließ sich der vorläufige Spielertyp *Explorer* nach Bartle [1] extrahieren.

3.1.3 Player Personas

Basierend auf [15] ist eine sogenannte *Player Persona* für den Spielertyp *Explorer* entwickelt worden. Auf diese Weise sollte die Empathie mit der Zielgruppe und deren Bedürfnisse gefördert werden.

3.2 Konzeptions-Phasen

3.2.1 Erste Iteration

Die erste Iteration orientierte sich zunächst an den Spielertyp *Explorer* nach Bartle, der im Rahmen des initialen Workshops (3.3.2) als vorläufige Basis für *Player Persona* (3.3.3.) und das *Gamification*-Konzept extrahiert werden konnte.

Nach [5] stellten u.a. die folgenden Spielmechaniken die am geeignetsten für den Spielertyp *Explorer* nach Bartle [1] dar, oder für andere Typen weiterer Autoren mit nahezu identischen Eigenschaften: *Story/Narrative*, *Aesthetics* und *World Building*. Einige initiale Ideen zur Integration eines *Gamification*-Ansatzes, die auf den genannten Mechaniken basierten, standen im Projektteam zur Diskussion. Die Projektinternen wiesen auf ihre bisherigen Erfahrungen und Eindrücke der Zielgruppe Prozessoptimierer hin und stellten heraus, dass die nahtlose Integration der *Gamification*-Elemente innerhalb der Arbeitssoftware-Umgebung essentiell sei. Die Hypothese der Befragten stellte hierbei dar, dass Ansätze von visuell ansprechenden oder herausstehenden Elementen bzw. die Einbin-

die eine übergreifende Narration ungeeignete, ablenkende Ansätze im Arbeitskontext darstellen könnten.

Die erwünschte Zielwirkung des zu entwickelnden *Gamification*-Konzepts ist die häufige, im besten Fall tägliche Nutzung des KI-Service zur Maschinenoptimierung. Zusammen mit der zuvor beschriebenen Einschätzung des Projektteams bzgl. klassischer Mechaniken des Typus *Explorer* konzentrierte sich das überarbeitete Konzept auf ein im Hintergrund laufendes Punkte-System, das zur Erreichung dreier Level dient. Die Frequenz und Art der Interaktionen mit dem KI-Service zur Maschinenoptimierung sollten erfasst und einem dreistufigen Levelsystem (Bronze, Silber, Gold) zugeordnet werden:

(1) *Brave Firefighter (Bronze)*

Aktion: dringende Probleme mit dem größten Potential lösen, *Nutzung:* sporadisch (mehrmals pro Monat).

(2) *Efficient Multiplier (Silber)*

Aktion: Maßnahme zur Optimierung auf mehrere Maschinen und Parameter anwenden, *Nutzung:* häufig (mehrmals pro Woche).

(3) *Shopfloor General (Gold)*

Aktionen: sämtliche Maßnahmen, die Einfluss auf das insgesamt eingesparte Potential besitzen, *Nutzung:* täglich.

Das dreistufige Levelsystem soll die Nutzungsfrequenz ständig erfassen, sodass bei einem Abfall der Nutzungshäufigkeit innerhalb eines vordefinierten Zeitraums eine Herabstufung im Level erfolgt.

Die Wahl von Levels als Spielelement zusammen mit Auszeichnungen ist einerseits auf die Auswertung des Interviews mit Mechanikarten, andererseits auf eine Erkenntnis von [9] zurückzuführen: Die Wahl der Elemente und der Kontext, in dem diese im Design verwendet werden, beeinflusst die Mechanik selbst und damit den gesamten Prozess [9]. Daraus lässt sich die Notwendigkeit des Einbezugs des (Arbeits-)Kontexts bei der Auswahl von Spielmechaniken für *Gamifica-*

tion-Ansätze ableiten, was die zuvor formulierte Hypothese der Projektinternen untermauert: Es sollten in den Arbeitskontext integrierbare Elemente, die sich von denen des klassischen Modells nach Bartle [1] unterscheiden, verwendet werden. Unter Einbezug von [9] sind Levels und Auszeichnung integriert worden, da diese als geeignete Elemente zur Ansprache des Belohnungs-Mechanismus für den Typus des *Explorer* benannt wurden.

Im Folgenden sind die im Konzept der ersten Iteration verwendeten Spielmechaniken mit den zugehörigen Elementen genannt:

- (1) *Mechanik:* Leistung & Errungenschaften [7], *Element:* Level
- (2) *Mechanik:* Leistung & Errungenschaften [7], *Element:* Auszeichnungen
- (3) *Mechanik:* Fortschritt & Orientierung [4], *Element:* Feedback-Elemente: Was tun für das nächste Level?
- (4) *Mechanik:* Aktionsbasierte Belohnungen [4], *Element:* Punkte im Hintergrund
- (5) *Mechanik:* General [17], *Element:* Verlust von Level und Punkten bei geringer Nutzung (*Loss Aversion*)
- (6) *Mechanik:* Stolz [7], *Element:* Auszeichnungen und Level-Status

3.2.2 Zweite Iteration

Die zweite Iteration konzentrierte sich nach einem weiteren Workshop innerhalb des Projektteams auf den Ansatz dem Anwender eine Top 10-Liste derjenigen Prozessschritte bereit zu stellen, die als Top-Potentiale kategorisiert wurden – unter Berücksichtigung mehrerer Faktoren wie Optimierungspotential, gesamte Laufzeit und wirtschaftliche Kennzahlen. Das vorab definierte Ziel (Motivation zur täglichen Nutzung des KI-Service zur Maschinenoptimierung) sollte durch die Schaffung eines neuen Anreizes zur täglichen Nutzung der App, z.B. als Einstiegspunkt die Bearbeitung der täglich aktualisierten „Top10-des-Tages“-Liste, verwirklicht werden.

Ausgehend von diesem Element und unter Einbezug der Auswertung des Gamification *User Type Test* zweier Probanden mit geringer Streuung (siehe Abschnitt 5.2) ist das entwickelte Punktesystem mit Auszeichnungen, Boostern und der Rangliste verknüpft, um mehrere Spielertypen anzusprechen. Auf Basis von [17] fanden Mechaniken Verwendung, die unabhängig vom Spielertyp sind. Hierzu zählen Feedback zum Fortschritt und Zeitdruck. Das in der Literatur [15] bekannte Risiko einer Demotivation bei der Verwendung einer Rangliste bei Verbleib auf den unteren Rängen sollte durch die Integration von Herausforderungen minimiert werden: Bei Vollendung ist ein Booster aktiv, meint eine Potenzierung der gesammelten Punkte innerhalb eines Zeitraums. Auf diese Weise können Anwender, die in den unteren Rängen des Leaderboards zu finden sind, durch Teilnahme an Herausforderungen einen schnelleren Aufstieg erfahren. Bei der Vollendung von Herausforderungen können ebenso Auszeichnungen gewonnen werden, die einerseits ebenso eine Booster-Wirkung aufweisen, andererseits insbesondere für den Spielertyp *Player* (z.B. nach [19]) als alleinstehendes Element motivierend wirken können.

Im Folgenden sind die im Konzept der ersten Iteration verwendeten Spielmechaniken mit den Elementen und dem Spielertyp genannt:

- (1) *Mechanik*: Fortschritt & Orientierung [7], *Element*: Anzahl der (noch zu) bearbeiteten „Top10-des-Tages“ (*Feedback* [17]), *Spielertyp*: General [17]
- (2) *Mechanik*: General [17], *Element*: Bearbeitungszeit von einem Tag für „Top10-des-Tages“ (*Time Pressure* [17]), *Spielertyp*: General [17]
- (3) *Mechanik*: Leistung & Errungenschaften [7], *Element*: Auszeichnungen (*Badges* [17]), *Spielertyp*: Player [17]
- (4) *Mechanik*: Aktionsbasierte Belohnungen [7], *Element*: Punkte bei Aktionen (*Points, Fixed Reward* [10]), Booster bei Herausforderungen (*Challenges* [10]), *Spielertyp*: Player, Achiever [10]

(5) *Mechanik*: Stolz [7], *Element*: Rang, Auszeichnungen (*Social Status* [10]), *Spielertyp*: Socialiser [17]

(6) *Mechanik*: Ranglisten [7], *Element*: Leaderboard [10], *Spielertyp*: Player [17]

3.3 Evaluations-Phasen

Die Projektmitarbeiter des Fraunhofer IPA konnten erste Einsichten in die Zielgruppe liefern, mit der sie seit mehreren Jahren in unterschiedlichen Projekten zusammenarbeiten. Diskussionen und Brainstorming Workshops stellten erste Orientierungspunkte zur Verfügung und dienten als Hilfestellung zur iterativen Adaption des *Gamification*-Konzepts an die Bedürfnisse der Nutzer.

Die Teilnehmer der ersten Evaluation bestanden aus zwei (externen) UX-Designern und zwei Endanwendern des KI-Software-Service. Schwerpunkt der Befragung stellte die intuitive Verständlichkeit des Konzepts und die dazugehörigen visuellen Elemente dar. Weiterhin wurden Einschätzungen zu motivierenden Aspekten in der Arbeitswelt und während der Nutzung von Software im Arbeitskontext mit den Probanden diskutiert.

Im Anschluss daran bearbeiteten die beiden Probanden den Fragebogen zu *Marczewski's Gamification User Type Test* [16], um im Nachgang einen Spielertyp extrahieren zu können.

Zum Zeitpunkt der Überarbeitung des *Gamification*-Konzepts standen tatsächliche Endanwender des Service bzw. praktizierende Prozessoptimierer nicht zur Verfügung. Die Evaluation fand mit insgesamt zwei internen Mitarbeitern des Fraunhofer IPA statt, die in ihrer Ausbildung und/oder ihrem aktuellen Beruf dem Berufsbild des Prozessoptimierers nahestehen.

4 Ergebnisse

4.1 Analyse-Phase

4.1.1 Mechanikkarten

Im Rahmen des initialen Interviews mit einem internen Mitarbeiter ist eine Wertung

(besonders positiv, positiv, negativ) der in [7] genannten Spielmechaniken vorgenommen worden. In Tabelle 1 sind diese zusammen mit der Wertung des Probanden gelistet:

Tabelle 1: Auswertung der Mechanik-karten, sortiert nach Wertung

Mechanik	Kategorie	
Schwimmweste	Hilfestellung	++
Höhere Ziele	Merkmale	+
Stolz	Merkmale	+
Personalisierung	Interface	+
Avatar	Interface	+
Botschafter	Hilfestellung	+
Einführung	Hilfestellung	+
Grundlagen-aufbau	Hilfestellung	+
Virtuelle Elemente	Gegenstände	+
Fortschritt & Orientierung	Fortschritt	+
Status	Ranglisten	+
Leistung & Errungenschaften	Belohnungen	+
Aktionsbasierte Belohnungen	Belohnungen	+
Easter Eggs	Belohnungen	+
Forscher- & Entdeckertum	Prinzipien	+
Information-häppchen	Hilfestellung	-
Bestrafung	Strafe	-
Ignoranz	Strafe	-
Rangliste	Ranglisten	-
Chats	Soziale Interaktion	-

4.1.2 Workshop: Nutzungskontext und Zielgruppe

Im Rahmen des Workshops ließ sich der vorläufige Spielertyp *Explorer* [1] extrahieren, insbesondere anhand der Erläuterungen des Projektmitarbeiters über positive Erlebnisse im Arbeitsalltag: Am meisten Motivation erhielt dieser bei der Aufdeckung eines Problems oder zu optimierenden Parameters. Der Rückschluss auf den Spielertyp *Explorer* nach Bartle [1] erfolgte u.a. auf Basis dieser Aussage, die sich mit Charakterisierungen der Literatur und Forschung deckt: In [5] wird der Typus *Explorer* als neu- und wissbegierig beschrieben, die *mit* der Welt interagieren und neues Wissen, neue Geheimnisse und Objekte innerhalb dieser aufdecken möchten. Zudem ist womöglich ein Hinweis auf den intrinsischen Motivationsfaktor des eigenen persönlichen Fortschritts und Autonomie zu erkennen, die sich u.a. ebenso bei [3] und [18] finden.

4.2 Erste Iteration

4.2.1 Evaluation

Die relevantesten Erkenntnisse aus der Evaluation in Bezug sind im Folgenden skizziert:

- (1) Rangbezeichnungen und die Systematik dahinter ohne Erläuterung nicht intuitiv verständlich (Proband A).
- (2) Ränge/Levels als kritisch in Bezug auf eine mögliche, motivierende Wirkung betrachtet (Proband A und B).
- (3) *Gamification* ein für Jüngere attraktiver Aspekt, für Ältere eine Spielerei ohne Mehrwert, die entweder als nicht störend empfunden wird, oder gar eine Ablenkung darstellt (Proband A und B).
- (4) Das Konzept motiviert nicht zu einer häufigeren Nutzung des Service, dies könnte lediglich äußerer Druck auslösen (Proband A und B).

4.2.2 User Type Test

Der Fragebogen zur Erfassung des Spielertypen nach Marczewski [16] basiert auf seinem

intrinsicischen Nutzertypenmodell, einem ähnlichen Modell wie bei Bartle [1] mit einigen Anpassungen und Ergänzungen der dort beschriebenen Typen. Der Typus *Free Spirit (FS)* nach Marczewski ist dem des *Explorer* nach Bartle nachempfunden, wohingegen Bartles *Killer* von Marczewski entschärft als *Player (PL)* beschrieben ist [11]. Zusätzlich sind die Typen *Philanthropist (PH)*, Motiv des höheren Ziels) und *Disruptor (D)*, Streben danach Systeme negativ zu beeinflussen, negativer *Explorer*) im Modell nach Marczewski zu finden [11]. Die Typen *Achiever (A)* und *Socialiser (S)* sind Bartle's Modell direkt entlehnt [11].

Die quantitative Aufschlüsselung der Ergebnisse des *Marczewski's Gamification User Type Tests* beider Probanden ist der nachstehenden Tabelle 2 zu entnehmen:

Tabelle 2: Auswertung des Marczewski's Gamification User Type Tests, Anteile in Prozent

	A	D	FS	PH	PL	S
DE 01	20	15	16	17	14	18
DE 02	20	12	18	18	18	14

Bei beiden Probanden ist keine starke Tendenz zu einem Spielertypen ersichtlich. Die ermittelten prozentualen Werte weisen lediglich eine geringe Streuung auf, sodass eine eindeutige Zuordnung zu einem Spielertyp kaum erfolgen kann.

4.2.3 Fazit der ersten Iteration

Die Evaluation mit zwei Probanden deutet darauf hin, dass das motivierende Potential von *Gamification* nur greifen kann, insofern die verwendeten Spielelemente nahtlos in den Arbeitskontext integriert sind. Zudem ist das Rang- und Levelsystem als alleiniges *Gamification*-Element nicht ausreichend, um eine motivierende Wirkung zu erzeugen.

Dies deckt sich mit der Auswertung des *User Type Tests* nach Marczewski [16]: Beide Probanden stimmen zwar mit dem Spielertyp

Achiever mit 20% am meisten überein, allerdings ist die Streuung der prozentualen Anteile zu gering, sodass keine eindeutige Zuordnung zu diesem Spielertypen vorgenommen werden kann. Dies wird insofern bestätigt, als dass beide Probanden sich kritisch gegenüber dem Level-System äußerten, obwohl dieses Element in der Literatur als ein geeignetes für den Typ *Achiever* bezeichnet wird.

4.3 Zweite Iteration

Die relevanten Erkenntnisse aus der Evaluation der zweiten Iteration sind im Folgenden skizziert:

- (1) Das *Gamification*-Konzept und die Systematik dahinter sind ohne Erläuterung nahezu vollständig intuitiv verständlich (Proband D).
- (2) Das *Gamification*-Konzept und die Systematik dahinter sind ohne Erläuterung nicht intuitiv verständlich (Proband C).
- (3) Bewertungskriterien für Rangliste müssen objektiv, nachvollziehbar und fair sein (Proband C)
- (4) Das Konzept motiviert nicht zu einer häufigeren Nutzung des Service, intrinsische Motivation kann nicht extern beeinflusst werden (Proband C).
- (5) Potential für *Gamification*-Ansätze bei Anwendung auf Team-Ebene, insofern diese nach objektiven Kriterien bewertet werden (Proband C).
- (6) Die Bearbeitung aller „Top10-des-Tages“ stellt nur ein attraktives Ziel dar, wenn ein Bonus oder Incentive daraus resultiert (Proband D).
- (7) Die Rangliste wurde teilweise kritisch wahrgenommen, aber insofern eine faire Vergleichbarkeit gewährleistet ist als potentiell positiv (Probanden C und D).

5 Diskussion

Die Anzahl der befragten Probanden ist nicht aussagekräftig, um valide Aussagen über die Zielgruppe und deren Bedürfnisse zu treffen. Eine Probandenzahl von mindestens sieben

Personen war zu Beginn angestrebt, konnte jedoch aufgrund externer Faktoren nicht umgesetzt werden.

Während der Befragungen mit Probanden fand die Evaluation des Spielertyps über *Marczewski's Gamification User Type Test* statt. Der Laddering-Ansatz nach [20] ist zu Beginn des Projekts erwogen worden, allerdings stellte die Befragungstechnik des *lauten Denkens* für die Probanden eine wenig intuitive und daher wenig zielführende Methode zur Erfassung ihrer Bedürfnisse dar. Das Einüben dieser Technik während der Evaluation hätte das verfügbare Zeitkontingent überschritten, sodass auf die weitere Verwendung des Ansatzes verzichtet wurde.

Die Ergebnisse werden zur Formulierung von Anforderungen an ein *Gamification*-Konzept für die Zielgruppe verwendet:

- (1) Onboarding des Gamification-Ansatz: Erreichte Level und Erfolge werden seitens der Zielgruppe in erster Linie hinterfragt, insofern die Kriterien dahinter unbekannt sind.
- (2) Fairness und Objektivität: Ein auffälliges, mehrfach genanntes Motiv der Befragten. Diese Aspekte sollten im Rahmen des Onboarding kommuniziert und als Basis der Kriterien einer Rangliste verwendet werden: z.B. durch die Bereitstellung mehrerer Ranglisten in Abhängigkeit des Nutzertyps und dessen Arbeitsweise, die sich innerhalb von Prozessoptimierern stark unterscheiden.
- (3) Boni und externer Druck: Laut den Probanden besitzen externe Faktoren einen potentiell motivierenden Einfluss auf ihre Handlungen. Im Arbeitskontext setzen diese gegenwärtig auf die interne Prozessmotivation oder das interne Selbstverständnis. Als zukünftige Ansatzpunkte können die instrumentelle Motivation und das externe Selbstverständnis erwogen werden.
- (4) Auszeichnungen: Ein Proband empfände Auszeichnungen positiv, insofern Andere darauf aufmerksam werden.

Dies kann durch eine prominenter Positionierung in der Anwendung erreicht werden, aber auch durch die Integration derer in die reale Welt als digital beispielbares, an der Kleidung anbringbares Abzeichen mit e-Ink-Technologie.

6 Fazit und Ausblick

Im Rahmen des iterativen Design- und Konzeptionsprozesses ist die Tendenz erkennbar gewesen, dass die Zuordnung von Spielertypen zu geeigneten Spielmechaniken und –elementen nicht automatisch den tatsächlichen Präferenzen der Nutzer entspricht. An dieser Stelle hätte ein Befragungsansatz wie Laddering, der sich vermehrt auf die Bedürfnisse der Nutzer konzentriert, möglicherweise eindeutiger Ergebnisse liefern können. Durch die Diskrepanz von erfasstem Spielertyp und tatsächlicher Wahrnehmung der Spielelemente ist die gewählte Vorgehensweise im Nachhinein weniger geeignet zur finiten Analyse der Zielgruppe gewesen. Dennoch konnten einige Anhaltspunkte bzw. Anforderungen an ein *Gamification*-Konzept mit dem Potential der Motivationssteigerung der Anwender eruiert werden, die als Basis für die zukünftige Weiterentwicklung dieses Ansatzes als Basis dienen können.

7 Literaturverweise

- [1] Bartle, R.A. Hearts, Clubs, Diamonds, Spades: Players Who Suit Muds. *Hearts, Clubs, Diamonds, Spades: Players Who Suit Muds*. <http://mud.co.uk/richard/hcds.htm>. Accessed 23 March 2019.
- [2] Broer, J. 2014. Gamification and the trough of disillusionment. *Mensch & Computer 2014-Workshopband*. (2014).
- [3] Chou, Y. 2013. User and Player Types in Gamified Systems. *Yu-kai Chou: Gamification & Behavioral Design*. <https://yukaichou.com/gamification-study/user-types-gamified-systems/>. Accessed 23 March 2019.
- [4] Deterding, S. et al. 2011. From Game Design Elements to Gamefulness: Defining “Gamification.” *Proceedings of the 15th international academic MindTrek conference*:

Envisioning future media environments (2011), 9–15.

[5] Ferro, L.S. et al. 2013. Towards personalised, gamified systems: an investigation into game design, personality and player typologies. *Proceedings of The 9th Australasian Conference on Interactive Entertainment Matters of Life and Death - IE '13* (Melbourne, Australia, 2013), 1–6.

[6] Hoos, S. 2016. Gamification in UX Design – one way to do it. *UP 2016* (Aachen, 2016).

[7] Hoos, S. 2015. Methode Mechaikkarten. *Playful Interaction Concepts*. <http://playful-interaction-concepts.wordpress.de/methode-mechaikkarten/>. Accessed 24 March 2019.

[8] Kindsmüller, M.C. et al. 2014. Zum Anwendungspotential von Gamification in Unternehmen. *Mensch und Computer 2014 Workshopband*. M. Koch et al., eds. De Gruyter Oldenbourg. 379–387.

[9] Kocadere, S.A. and Çağlar, Ş. 2018. Gamification from Player Type Perspective: A Case Study. *Educational Technology & Society*. 21, 3 (2018), 12–22.

[10] Koivisto, J. and Hamari, J. 2014. Demographic differences in perceived benefits from gamification. *Computers in Human Behavior*. 35, (Jun. 2014), 179–188.

[11] Korbas, S. 2015. Gamification User Types zur Unterstützung der Konzeption in der Softwareentwicklung. *Mensch und Computer 2015–Workshopband*. (2015).

[12] Korn, O. et al. 2012. Assistive Systems in Production Environments: Exploring Motion Recognition and Gamification. *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments* (New York, NY, USA, 2012), 9:1–9:5.

[13] Korn, O. et al. 2015. Design approaches for the gamification of production environments: a study focusing on acceptance. *Pro-*

ceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments (2015), 6.

[14] Korn, O. and Schmidt, A. 2015. Gamification of business processes: Re-designing work in production and service industry. *Procedia Manufacturing*. 3, (2015), 3424–3431.

[15] Kumar, J. 2013. Gamification at Work: Designing Engaging Business Software. *Design, User Experience, and Usability. Health, Learning, Playing, Cultural, and Cross-Cultural User Experience* (2013), 528–537.

[16] Marczewski, A. Gamified UK Gamification User Type HEXAD Test. *Gamified UK Blog*. <https://gamified.uk/UserTypeTest2016/user-type-test.php>. Accessed 23 March 2019.

[17] Marczewski, A. 2017. The Periodic Table of Gamification Elements. (2017). <https://www.gamified.uk/2017/04/03/periodic-table-gamification-elements/>. Accessed 25 March 2019.

[18] Ryan, R.M. et al. 2006. The Motivational Pull of Video Games: A Self-Determination Theory Approach. *Motivation and Emotion*. 30, 4 (2006), 344–360.

[19] Tondello, G.F. et al. 2016. The Gamification User Types Hexad Scale. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, 2016), 229–243.

[20] Vanden Abeele, V. and Zaman, B. 2009. Laddering the User Experience! *User Experience Evaluation Methods in Product Development (UXEM'09) - Workshop* (Aug. 2009).

[21] Warmelink, H. et al. 2018. Gamification of production and logistics operations: Status quo and future directions. *Journal of Business Research*. (2018).

[22] Xu, L.D. et al. 2014. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*. 10, 4 (Nov. 2014), 2233–2243.

Reutlingen University

Fakultät Informatik

Human-Centered Computing

Alteburgstraße 150

72762 Reutlingen

<http://www.huc.reutlingen-university.de>

Telefon: +49 7121 / 271-4002

Telefax: +49 7121 / 271-4042

E-Mail: *infoinside@reutlingen-university.de*

Internet: *<http://infoinside.reutlingen-university.de>*



Hochschule Reutlingen
Reutlingen University

