FOM Hochschule für Oekonomie & Management

Study centre Stuttgart

Bachelor-Thesis

in the study programme Business Information Systems

to obtain the degree of

Bachelor of Science (B.Sc.)

on the topic

Development of a computer vision prototype to measure soccer player skills

by

Raffael Rizzo

Matriculation number:	497861
First assessor:	Hamid, Maher
Submission date:	04.11.2021

Table of contents

Index of figures

Index of tables

Index of abbreviations

AI	Artificial intelligence
AJSE	Arabian Journal for Science and Engineering
ANN	Artificial neural network
CNN	Convolutional neural network
CV	Computer vision
DFB	Deutscher Fußball-Bund
DL	Deep learning
GUI	Graphical user interface
hfv-info	Hamburger Fußball-Verband Info
IEEE	IEEE Transactions on Visualization and Computer Graphics
IJETAE	International Journal of Emerging Technology and Advanced Engineering
ML	Machine learning

Source code list

1 Introduction

1.1 Problem and research question

"It is particularly important to us to find out how we can use technology in a meaningful way and thus create a state-of-the-art environment for the development of our young talents."[1] This quoted blog entry by Dr. Peter Goerlich, the former TSG 1899 Hoffenheim managing director, illustrates soccer's increasing digitization. In his statement, he refers to the Bundesliga soccer club TSG Hoffenheim, which in SAP has the Market leader for enterprise software as a cooperation partner for innovation projects.[2] But what about the entire soccer industry? Grand View Research estimated the global market revenue for sports technologies at $ 11.70 billion in 2020.[3] By 2028, the market is expected to grow at an average annual growth rate of 16.8%.[4] The soccer segment led this division with over 15% revenue share.[5] This revenue is predominantly attributable to the professional sector, considering that the record price paid for the soccer player Neymar is 222 million €.[6]

A review of the business reports of FC Bayern and Borussia Dortmund confirms that compared to amateur teams, professional soccer teams have more of everything: money, time, personnel, and know-how.[7] Despite the Corona pandemic, FC Bayern could achieve revenues of 698 million €[8] and Borussia Dortmund of 370.2 million €[9] in the 2019-2020 season. After all deductions, FC Bayern had a net profit of €9.8 million[10] while Borussia Dortmund had a loss of 44 million €[11], which shows how high the expenses of these clubs are. One specific investment sector is sports data analysis, which has gained enormous importance.[12] A prominent example of data collection through modern software is Cristiano Ronaldo's holistic performance

---

[1] *Brecht*, *C*., Technologies, 2020, no page number.
[2] Cf. *SAP SE*, SAP, 2021, no page number.
[3] Cf. *Grand View Research*, *Inc*, Sports Segment, 2021, no page number.
[4] Cf. *ibid*., no page number.
[5] Cf. *ibid*., no page number.
[6] Cf. *Zeppenfeld*, *B*., Neymar, 2021, no page number.
[7] Cf. *Borussia Dortmund GmbH & Co. KGaA,* Geschäftsbericht, 2021, p. 8.; Cf. *FC Bayern München AG,* Jahresabschluss, 2020, no page number.
[8] Cf. *FC Bayern München AG,* Jahresabschluss, 2020, no page number.
[9] Cf. *Borussia Dortmund GmbH & Co. KGaA,* Geschäftsbericht, 2021, p. 8.
[10] Cf. *FC Bayern München AG,* Jahresabschluss, 2020, no page number.
[11] Cf. *Borussia Dortmund GmbH & Co. KGaA,* Geschäftsbericht, 2021, p. 8.
[12] Cf. *Stein*, *M*. *et al*., Sport Analysis, 2018, p. 13.

diagnostics from 2016.[13] On this occasion, the staff of a Madrid sports laboratory assessed the football star's physical strength, mental ability, technique, and skills.[14] But also, amateur clubs greatly need digitization, as the 3rd DFB Amateur Football Congress 2019 points out.[15] Compared to professional clubs, amateur clubs make decisions on an emotional level for the most part. Meanwhile on the other side, there are ambitious clubs, coaches, and players willing to spend money.[16] In any case, they have two options to create an objective basis for decision-making.

On the one hand, they can perform tests manually, as the Deutsche Fußball-Bund, hereafter abbreviated DFB, does.[17] As a part of a talent development program, the DFB offers additional weekly training for the most talented players in Germany.[18] This training includes a biannual performance test for these players.[19] The test is conducted manually and illustrates that even the DFB has by no means digitized all processes.[20]

On the other hand, they may utilize charged tracking solutions from companies like STATSports or TRACKTICS.[21] Their products have tiny sensors that record conditional abilities like running distances, sprints, and maximum speeds during exercise.[22] Significant shortcomings of these solutions arise as they do not measure technical skills.[23] However, besides the conditional abilities, a promising footballer stands out for his technical skills. [24] Technical mistakes lead to ball losses, goals and are decisive for the game.[25] Therefore the question arises, isn't it possible to use software for automated measurement of technical soccer skills?

In short: Professional teams like Borussia Dortmund and FC Bayern have the necessary resources to realize fully comprehensive analyses, which are essential to

[13] Cf. *McDowall*, *M*., Ronaldo, 2011, no page number.
[14] Cf. *ibid*., no page number.
[15] Cf. *Byernetzki*. *C*., Amateur-Kongress, 2019, p. 3.
[16] Cf. *ibid*., p. 3.
[17] Cf. *Lottermann*, *S*. *et al*., Leistungsdiagnostik, 2003, p. 91–105.
[18] Cf. *ibid*., p. 91–105.
[19] Cf. *ibid*., p. 91–105.
[20] Cf. *ibid*., p. 91–105.
[21] Cf. *STATSports Group Limited*, STATSports, 2020, no page number.; Cf. *TRACKTICS GmbH*, TRACKTICS, 2021, no page number.
[22] Cf. *STATSports Group Limited*, STATSports, 2020, no page number.; Cf. *TRACKTICS GmbH*, TRACKTICS, 2021, no page number.
[23] Cf. *STATSports Group Limited*, STATSports, 2020, no page number.; Cf. *TRACKTICS GmbH*, TRACKTICS, 2021, no page number.
[24] Cf. *Ruck*, *R*., *Schmidbauer*, *H*., Technical Skills, 2013, p. 72.
[25] Cf. *Ruck*, *R*., *Schmidbauer*, *H*., Technical Skills, 2013, p. 72.

their success.[26] By contrast, most amateur clubs use their resources elsewhere.[27] As a result, ambitious amateur- clubs, coaches, and players have two options. They either conduct performance tests manually by themselves or finance existing solutions out of their own pockets.[28] These solutions provide information about the conditional abilities of a player.[29] But, a comprehensive performance analysis also includes other performance factors, especially technical skills.[30]

1.2 Objective and organization of the research

The problem and question described in section 1.1 lead to the following research question: Is it possible to measure the technical skill of ball juggling in soccer using computer vision?

The aim is to create a proof of concept for the automated evaluation of ball juggling in soccer through prototypical development. The goal is achieved by replacing manual counting through prototypical development.

The development of a prototype is of great importance. The prototype's positive proof of feasibility can answer the in-depth research question conclusively and thus can lead to the achievement of the bachelor thesis goal.

In the next chapter 2, a knowledge base is made by defining the terms prototype, artificial intelligence, abbreviated AI, computer vision, and a soccer player's requirement profile. Subsequently, comparing two possible machine learning approaches to prototypical development concludes the theoretical basis.

After that, chapter 3 explains the scientific methodology and defines requirements that the prototype should fulfill. These requirements are then integrated into a mockup.

The mockup depicts the various interfaces of the prototype and serves as the basis for the subsequent source code implementation in chapter 4.

---

[26] Cf. *Borussia Dortmund GmbH & Co. KGaA,* Geschäftsbericht, 2021, p. 8.; Cf. *FC Bayern München AG,* Jahresabschluss, 2020, no page number.
[27] Cf. *Byernetzki*. *C*., Amateur-Kongress, 2019, p. 3.
[28] Cf. *Lottermann*, *S. et al*., Leistungsdiagnostik, 2003, p. 91–105.; Cf. *STATSports Group Limited*, STATSports, 2020, no page number.; Cf. *TRACKTICS GmbH*, TRACKTICS, 2021, no page number.
[29] Cf. *STATSports Group Limited*, STATSports, 2020, no page number.; Cf. *TRACKTICS GmbH*, TRACKTICS, 2021, no page number.
[30] Cf. *Ruck*, *R*., *Schmidbauer*, *H*., Technical Skills, 2013, p. 72.

Afterward, a prototype simulation, a test group, and a questionnaire help evaluate the prototype in chapter 5. Thus, positive aspects and challenges of the prototype are to be derived. In this context, whether the prototype can already be made available to the public arises.

Finally, a collection of ideas on the further development of the prototype follows before an outlook finalizes the bachelor thesis.

2 Theoretical foundation

2.1 Prototype

A prototype is a general term that is defined differently depending on the area of application.[31] In general, a prototype can be subsumed under a slim design of the desired result.[32] This result can be a physical product, a software system, a model, or even a process.[33] A simple sketch, a design draft, and a usable functionality are examples of prototypes with ascending complexity.[34] In this bachelor thesis, both a design sketch and the implementation of an entirely usable functionality are envisaged. More simply, two prototypes are designed that are interrelated.

2.2 Artificial intelligence

Unfortunately, a generally valuable definition for AI does not exist. The term AI is subject to constant further development.[35] In the past, it was about programming machines to behave intelligently, e.g., to play chess.[36] Today, it is more about teaching machines to learn.[37] Furthermore, in the future, the focus might be somewhere else again.

Stanford professor John McCarthy defined AI as early as 1955 as "the science and technology of making intelligent machines." [38] This abstract definition is still used today but raises further questions.[39] For example, to what extent the term intelligence can be defined. None of the directly related sciences can answer this question precisely.[40] However, this topic is not the subject of this bachelor's thesis. Instead, the focus is on the sub-areas of ML, short for machine learning[41] and computer vision hereafter abbreviated CV.[42] The following examples are intended to shed more light on the subject area of AI.

---

[31] Cf. *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 1–2.
[32] Cf. *ibid*., p. 1–2.
[33] Cf. *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 1–2.
[34] Cf. *ibid*., p. 1–2.
[35] Cf. *Neander*, *J*., Chess, 1996, no page number.
[36] Cf. *ibid*., no page number.
[37] Cf. *Khanna*, *R*., *Awad*, *M*., Machine Learning, 2015 p.1–3.
[38] *Manning*, *C*., AI & ML Definition, 2020, no page number.
[39] Cf. *ibid*., no page number.
[40] Cf. *Kirste*, *M*., *Schürholz*, *S*., AI Meaning, 2018, p. 21.
[41] Cf. Khanna, R., Awad, M., Machine Learning, 2015 p. 1.
[42] Cf. *Russell*, *S*. *et al*., AI, 2020, p. 32.

In 1996, IBM's Deep Blue computer beat the world chess champion, Garry Kasparov.[43] In March 2016, Lee Sedol, the international winner of the Japanese board game Go, lost against Deep Mind's AlphaGo.[44] In 2018, Google's Duplex system showed an AI system acting as a customer on the phone to book a hair appointment and reserve a table at a restaurant.[45] That same year, the podcast "Sheldon County" appeared, written and voiced entirely by an AI.[46]

What do these events have in common? On some level, correlations with activities that generally require human intelligence are recognizable. For another, the versatile use of AI becomes clear.

The areas of application can be traced back to the Turing test proposed in 1950.[47] The purpose of this test was to find out whether a computer could pass a human-oriented test.[48] In this process, capabilities were worked out that a computer must possess.[49]

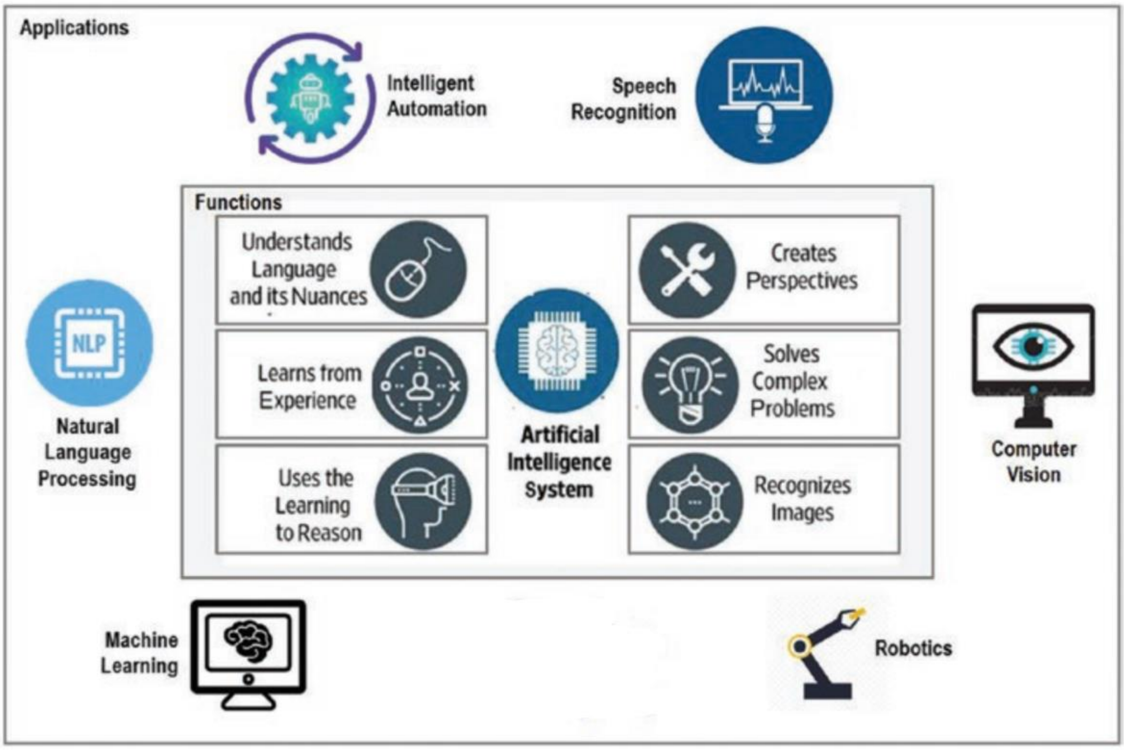
These include Natural language processing to communicate with people.[50] Knowledge representation for storing recorded knowledge.[51] Automated reasoning as a means of answering questions and drawing new conclusions.[52] ML, as a way of adapting to particular circumstances and recognizing patterns.[53] CV and speech recognition to perceive the environment.[54] Moreover, finally, robotics as a way to physically move around.[55]

These capabilities form the so-called sub-areas of AI, which are visible in the figure below.[56]

---

[43] Cf. *Neander*, *J*., Chess, 1996, no page number.
[44] Cf. *Wittpahl*, *V*., Go, 2018, p. 7.
[45] Cf. *Leviathan*, *Y*., Matias, Y., Google Duplex, 2018, no page number.
[46] Cf. *Schwan*, *B*., Podcast, 2018, no page number.
[47] Cf. *Russell*, *S*. *et al*., AI, 2020, p. 32.
[48] Cf. *ibid*., p. 32.
[49] Cf. *ibid*., p. 32.
[50] Cf. *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 7.
[51] Cf. *Russell*, *S*. *et al*., AI, 2020, p. 32.
[52] Cf. *ibid*., p. 32.
[53] Cf. Khanna, R., Awad, M., Machine Learning, 2015 p. 1.
[54] Cf. *Russell*, *S*. *et al*., AI, 2020, p. 32.
[55] Cf. *ibid*., p. 32–33.
[56] Cf. *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 7.

Figure 1: AI subfields



Source: *Gollapudi*, *S*., AI Subfields, 2019, p. 6

The example of the Turing test shows that the sub-areas are interconnected.[57] Also, the juggling prototype later described in this thesis, implements the two sub-areas ML and CV, which shows that real-world application examples usually consist of several subfields.

2.3 The performance profile of a soccer player

Before the topics of CV and ML are looked at in more detail, the exercise to be implemented here, ball juggling, is assigned to the performance profile of a soccer player.

As mentioned in the introduction, even the DFB, the governing body in German soccer, manually measures technical skills in soccer.[58] Therefore, to better understand what is meant by technical skills and how they fit into the requirement profile of a soccer player, the performance factors are explained.[59] The performance

[57] Cf. *ibid*., p. 7.
[58] Cf. *Lottermann*, *S*. *et al*., Leistungsdiagnostik, 2003, p. 91–105.
[59] Cf. *Roth*, *K*., Performance Factors, 2014, no page number.

factors are multifaceted, consist of numerous subcategories, and are described differently in various sports science sources.[60] However, according to the scientifically prestigious Institute for Sport and Sport Science Heidelberg, the following figure shows the most critical performance factors.[61]

Figure 2: Performance factors

Direct performance factors			
Physique	Athletics	Tactic	Technique
Weight	Strength	Playing intelligence	Finish
Size	Stamina	Playing creativity	Dribbling
BMI	Speed	Speed of action	Pass
Biological maturity	Mobility		Ball handling
	Coordination		

Indirect performance factors		
Environment	Sport Experience	Personality
Support	Club History	Motivation
Distance to the club	Number of matches	Will
Compatibility between school and soccer	Hobbys	Self-perception
Sports careers of the environment	Compensatory sport	Motivation
Relationship to the trainer		

Source: Modelled after *Roth*, *K.*, Performance Factors, 2014, no page number.

The performance of a soccer player depends on direct and indirect performance factors.[62] Indirect performance factors cover the environment, the sporting experience, and the personality of a soccer player.[63] In contrast, direct performance factors result from physique, athleticism, tactics, and technique.[64] The physique encompasses height, weight, Body Mass Index, and biological age.[65] The latter indicates the general condition of a player at a certain point in his calendrical age, characterized by physical, psychological, and social features.[66] Athletics includes the conditional skills of strength, stamina, speed, coordination, and mobility.[67] Tactics are made up of game intelligence, game creativity, and speed of action.[68]

Last but not least, technical skills consist of finishing, dribbling, passing, and ball handling.[69] Even the most athletic and tactically skilled player will not be highly

[60] Cf. *ibid*., no page number.
[61] Cf. *ibid*., no page number.
[62] Cf. *ibid*., no page number.
[63] Cf. *Roth*, *K*., Performance Factors, 2014, no page number.
[64] Cf. *ibid*., no page number.
[65] Cf. *ibid*., no page number.
[66] Cf. *Karasik et al*., Biological Age, 2005, p. 575.
[67] Cf. *Reinhold*, *T*., Athletic, 2008, p. 4.
[68] Cf. *Roth*, *K*., Performance Factors, 2014, no page number.
[69] Cf. *Bisanz*, *G*., *Gerisch*, *G*., Technique, 2007, p. 321.

effective if he lacks the necessary technique.[70] Therefore technique is one of the central performance factors.[71] Especially ball juggling is a popular technical exercise that counts towards ball handling.[72] It is part of the DFB test battery[73] and much more complex than other technical exercises such as passing.[74] Therefore it has a high significance for the technical ability of a soccer player.[75] Furthermore, juggling is possible in a small space, which facilitates the deployment of the prototype.[76]

During the exercise, a player tries to pick up the soccer ball from the ground with his foot. [77] Then the ball is held in the air with all body parts except the hand for the longest time possible.[78] Once the ball falls to the ground, the number of ball contacts during the juggling is written down.[79]

2.4 Computer vision

Humanity's vision is a complex affair.[80] Everything began millions of years ago when a mutation developed in microorganisms that enabled them to see.[81] Today, many living things have a visual system with similar functionality. [82] Here, the eyes catch the light, then the light is absorbed in the brain and processed by the visual cortex.[83]

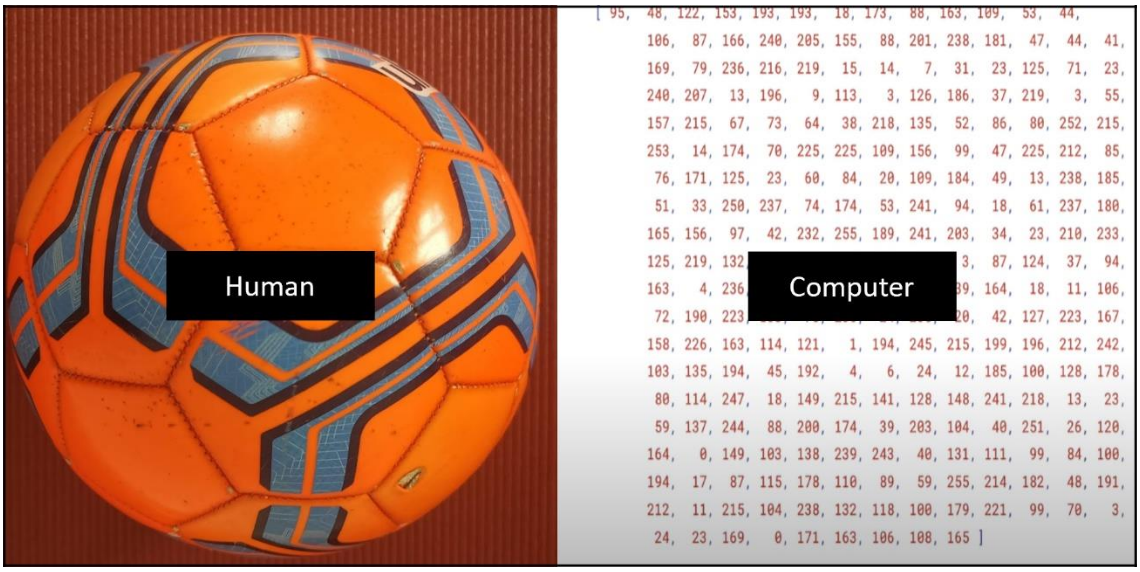
The question arises, how do cameras process images? The first type of the photo camera was invented around 1826.[84] For a picture, a silver chloride-covered paper was required.[85] The paper was in a box with a closure.[86] As soon as the closure was opened, the silver chloride darkened where it was exposed to light.[87] This created a black and white image.[88] Today, two centuries later, camera systems can process

[70] Cf. *Drobisch*, *D*., *Stratmann*, *A*., Soccer Technique, 2021, p. 62.
[71] Cf. *ibid*., p. 62.
[72] Cf. *Lottermann*, *S*. *et al*., Leistungsdiagnostik, 2003, p. 91–105.
[73] Cf. *ibid*., p. 91–105.
[74] Cf. *Schreiner*, *P*., Ball Juggling, 2009, p. 28 et seqq.
[75] Cf. *ibid*., p. 28 et seqq.
[76] Cf. *ibid*., p. 28 et seqq.
[77] Cf. *Schreiner*, *P*., Ball Juggling, 2009, p. 28.
[78] Cf. *ibid*., p. 28.
[79] Cf. *ibid*., p. 28.
[80] Cf. *Harari*, *Y*., Light, 2015, p.16.
[81] Cf. *ibid*., p. 16.
[82] Cf. *Frings*, *S*., *Müller*, *F*., Eye, p. 125.
[83] Cf. *ibid*., p. 125.
[84] Cf. *Sehnbruch*, *L*., Camera, 2017, p. 275.
[85] Cf. *ibid*., p. 275.
[86] Cf. *ibid*., p. 275.
[87] Cf. *ibid*., p. 275.
[88] Cf. *ibid*., p. 275.

color and light just like the human eye.[89] As a result, images can be taken directly in digital form.[90]

However, this does not mean that a computer can easily understand the content of the picture.[91] While humans have the content passed down from generation to generation,[92] computers only have numbers as input information.[93] Figure 3 contrasts what an image looks like to a human and a computer.

Figure 3: Human vs. Computer



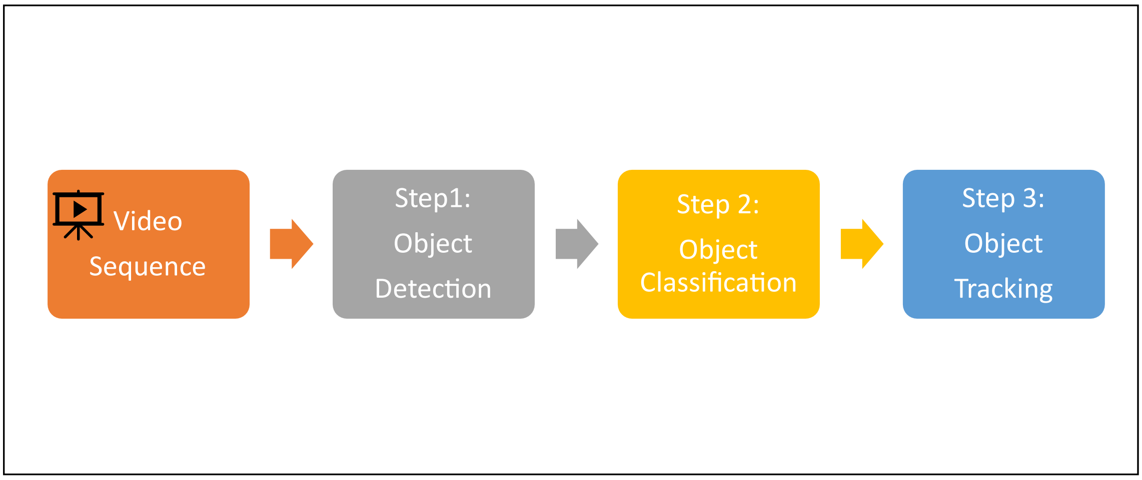
Source: Own presentation

A human can directly recognize that it is a soccer ball based on the information passed down from generations.[94] For a computer, on the other hand, the image is initially just an array of numbers that represent a color spectrum.[95] Therefore, the content information is crucial to classify images like the human brain.[96] Here comes the task of CV into its stride. CV can be classified "as a field of AI that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that

[89] Cf. *Sehnbruch*, *L*., Camera, 2017, p. 11.
[90] Cf. *ibid*., p. 11.
[91] Cf. *Rheinboldt*, *W*., *Rosenfeld*, *A*., Picture Array, 2014, p. 1–3.
[92] Cf. *Antz*, *J*. *et al*., Information, 2009, p. 26.
[93] Cf. *Rheinboldt*, *W*., *Rosenfeld*, *A*., Picture Array, 2014, p. 1–3.
[94] Cf. *Antz*, *J*. *et al*., Information, 2009, p. 26.
[95] Cf. *Rheinboldt*, *W*., *Rosenfeld*, *A*., Picture Array, 2014, p. 1–3.
[96] Cf. *ibid*., p. 1–3.

information."[97] As part of this bachelor's thesis, an object, more precisely a soccer ball, will be recognized on a video, then classified and finally tracked. This process can be divided into three superordinate steps.[98] The following Figure 4 summarises the steps once again.

Figure 4: Soccer ball tracking



Source: Modelled after *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 120

Two widely used approaches exist for soccer ball tracking that uses different ML techniques and algorithms. In the following section, 2.5, these two approaches are defined and compared.

2.5 Machine learning and deep learning

In the following three subsections, two approaches for prototype development will be explained and compared.

2.5.1 Traditional ML approach

ML is a subfield of AI[99] based on finding specific patterns or rules in given data sets in an automated way.[100] This information can be applied to further data with a similar structure to achieve the desired result with high accuracy.[101] In simpler terms, "Machine learning is the technique of helping a machine to think so it can perform

[97] *IBM*, CV, n.d., no page number.
[98] Cf. *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 120.
[99] Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
[100] Cf. *Bostrom*, *N*., ML & AI, 2014, p. 46.
[101] Cf. *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 11.

actions on its own."[102] Most CV programs use ML algorithms.[103] This prototype will also be implemented using ML. The goal is that the program receives image data in the form of a video source. After that, ML is used to teach the program context to the image data, which it uses to recognize and track the soccer ball.

The first approach considered for the prototype is color and contour-based ball recognition.[104] This traditional approach is based on the use of established ML techniques.[105] Here, the CV engineer tries to extract interesting features, in this case, the ball's color and shape, from images and assign them to an object class.[106] A four-step workflow is followed to recognize the ball by its shape and color.[107] The first step is to ensure that the prototype receives data relevant to the evaluation of the desired target.[108] Accordingly, the prototype requires video data with the exercise execution of ball juggling.[109] This data is pre-processed in the second step to minimize the error rate, which means that ML functions remove irrelevant data, such as noise in an image.[110]

Finally, the third step is to develop the ML model that recognizes, classifies and tracks the ball. "Which means this model takes the data as input, performs computations on it, then produces some output from it."[111] Usually, ML functions blackout all pixels except those of the desired object. As a result, the desired object remains, whose position can be tracked.[112] An evaluation of the ML model is the last step to find out the accuracy.[113] However, if the evaluation reveals inaccuracies, the CV engineer must manually retrain the model.[114]

2.5.2 Deep learning approach

An approach using Deep learning, hereafter abbreviated DL is the second approach. It is a sub-field of ML and uses algorithms that resemble the way the human brain

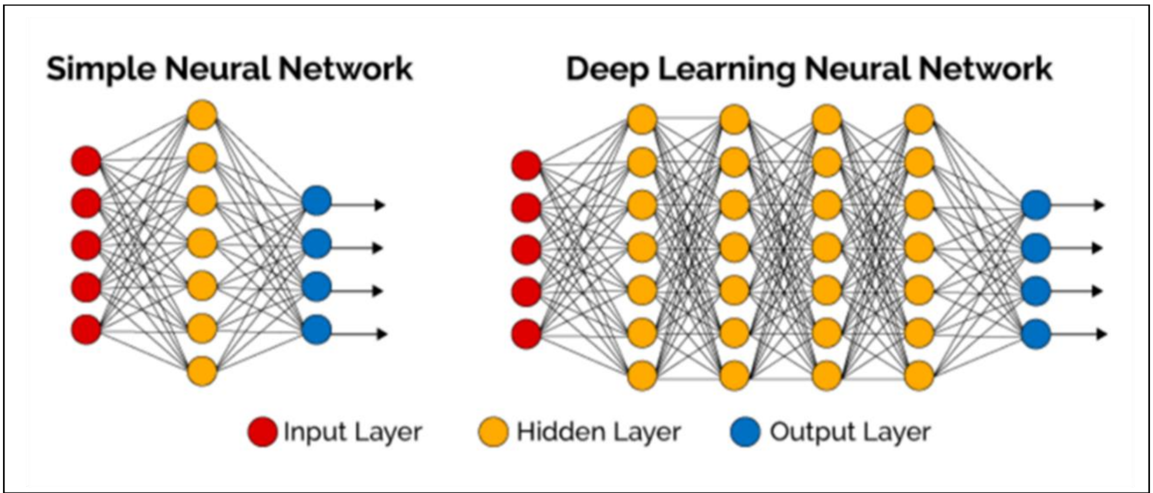
---

[102] *Silaparasetty*, *N*., ML Workflow, 2020, p. 57.
[103] Cf. *Russell*, *S. et al*., AI, 2020, p.1206.
[104] Cf. *Kadouf*, *H. H.*, *Mustafah*, *Y. M.*, Traditional Approach, 2013, p. 7.
[105] Cf. *ibid*., p. 7.
[106] Cf. *Kadouf*, *H. H.*, *Mustafah*, *Y. M.*, Traditional Approach, 2013, p. 2–5.
[107] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 22.
[108] Cf. *ibid*., p. 22.
[109] Cf. *OpenCV*, GaussianBlur, 2021, no page number.
[110] Cf. *ibid*., p. no page number.
[111] *Silaparasetty*, *N*., ML Workflow, 2020, p. 22–23.
[112] Cf. *OpenCV*, Object Detection, 2021, no page number
[113] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 23.
[114] Cf. *ibid*., p. 23.

works.[115] More precisely, the human biological neural network is mimicked by an artificial neural network, short ANN.[116] This ANN consists of many neurons that are interconnected on at least three layers.[117] The first layer is the input layer, which consists of input signals that are further transmitted into the ANN.[118]

The hidden layer forms the second layer. This is where all calculations are executed with the inputs and forwarded to the next layer.[119] If an ANN consists of several hidden layers, it is called a deep neural network or deep learning neural network, implying DL.[120] Finally, the last layer is the output layer, which contains the results of all calculations.[121] Figure 5 opposes a simple neural network to a deep learning neural network.

Figure 5: Neural networks



Source: *Vázquez*, *F*., Neural Network, 2017, no page number.

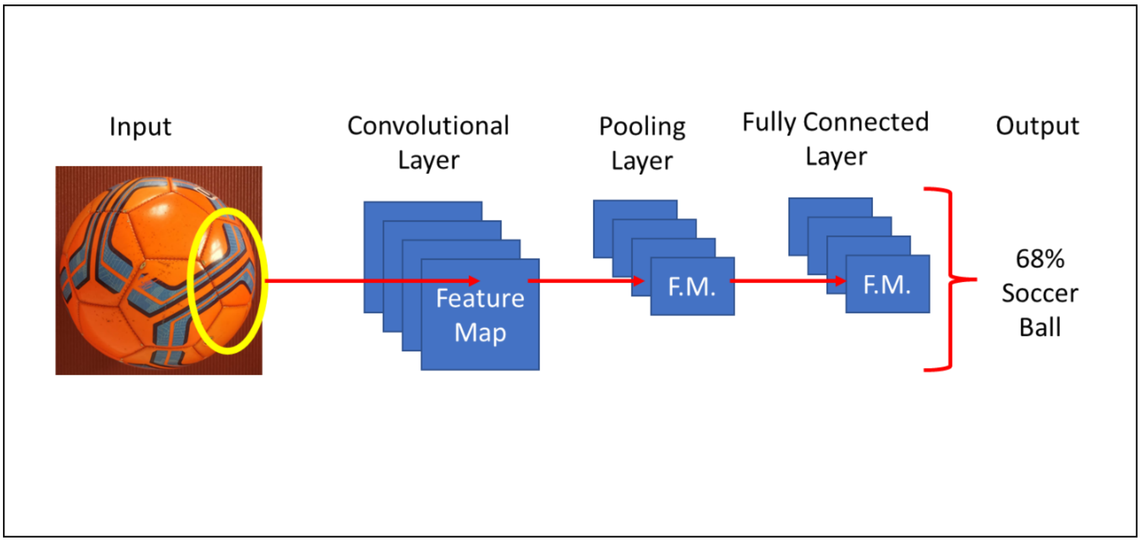
The DL method Faster R-CNN allows ball recognition in real-time and is a better performing development of a convolutional neural network, hereafter abbreviated CNN.[122] A CNN is an expression of an ANN whose principles are used for object recognition of many DL approaches.[123] Therefore the following three layers, which

---

[115] Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
[116] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 57.
[117] Cf. *Gollapudi*, *S*., Artificial Intelligence, 2019, p. 58–59.
[118] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 49.
[119] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 49–50.
[120] Cf. *Kirste*, *M*., *Schürholz*, *S*., AI Meaning, 2018, p. 30–31.
[121] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 50.
[122] Cf. *Ren et al*., Deep Learning, 2017, p. 1136 et seqq.
[123] Cf. *ibid*., p. 1137.

form the building blocks of a CNN architecture, are demonstrated in Figure 6 using a soccer ball.

Figure 6: Process of a CNN



Source: Own presentation

The convolution layer is the first layer of a CNN and recognizes concise and relevant features such as corners and curves.[124] That means a frame with the soccer ball is divided into small pixel components in this layer.[125] Afterward, the pixels are compared with pixels in a particular pattern for matches.[126] The resulting frame is called a "feature map".[127]

The pooling layer follows this as the second layer.[128] Therein the size of the feature maps is compressed, e.g., by the max-pooling operation.[129] Lastly, there is the Fully-Connected Layer.[130] Here the feature maps are classified to return an output.[131] After passing through the layers, the CNN can predict an object, though initially inaccurate.[132] Therefore a failure function is used after each prediction to match the prediction with the actual input object.[133] Thereupon, the CNN optimizes the filter

---

[124] Cf. *Michelucci*, *U.*, CNN, 2019, p. 105.
[125] Cf. *ibid.*, p. 105.
[126] Cf. *Gollapudi*, *S.*, Artificial Intelligence, 2019, p. 65.
[127] Cf. *Michelucci*, *U.*, CNN, 2019, p. 179.
[128] Cf. *Silaparasetty*, *N.*, ML Workflow, 2020, p. 54.
[129] Cf. *ibid.*, p. 54.
[130] Cf. *Gollapudi*, *S.*, Artificial Intelligence, 2019, p. 65.
[131] Cf. *ibid.*, p. 65.
[132] Cf. *Kirste*, *M.*, *Schürholz*, *S.*, AI Meaning, 2018, p. 32.
[133] Cf. *ibid.*, p. 32.

values and restarts the process.[134] This process is called backpropagation and takes place until a reasonable prediction is made.[135]

In the following subsection, the advantages and disadvantages of both approaches are compared. Thereafter, it will be decided which approach will be used for the implementation of the prototype.

2.5.3 ML approach vs. DL approach

Based on the previous two subsections, it is clear that ML is a subfield of AI[136] and enables machines to perform actions in an automated manner.[137] In turn, DL is a subset of ML[138] that enables machines to perform actions automatically through ANNs.[139] Both the traditional ML approach and the DL approach can be considered for prototype development. Based on six characteristics, a comparison of both approaches follows in Table 1. It is worth mentioning that algorithms vary depending on the use case, and exceptions are not considered in the comparison.

[134] Cf. *Kirste*, *M*., *Schürholz*, *S*., AI Meaning, 2018, p. 32-33.
[135] Cf. *ibid*., p. 32-33.
[136] Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
[137] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 57.
[138] Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
[139] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 57.

Table 1: Traditional ML vs. DL approach

	Traditional ML Approach	DL Approach
Accuracy	There is a smaller database to learn. Moreover, it exists dependency on feature representation. Thus the accuracy is lower.[140]	Conclusions are drawn from a much more extensive database. Thus the accuracy is higher.[141]
Data Amount	Based on a smaller database. Only data for direct processing is required.[142]	A large amount of data is needed to ensure that without external adjustments, the model is trained.[143]
Computational Power	Less data needs to be analyzed, which means less computational power is required.[144]	More computational power is needed to evaluate and train large data sets.[145]
Cognitive Ability	Lower cognitive ability, which means that if the model has inaccuracies, it must be manually retrained.[146]	Higher cognitive ability, since the system can detect and train inaccuracies on its own.[147]
Hardware Requirements	The algorithms are trainable and executable on most conventional computers.[148]	Conventional hardware reaches its limits with DL algorithms. Therefore, GPUs are usually required to train the processes.[149]
Time Taken	The models consist of fewer parameters and are trainable more quickly.[150]	Training the models takes more time as they consist of more parameters.[151]

Source: Modelled after *Silaparasetty*, *N*., ML Workflow, 2020, p. 61–64.

[140] Cf. *Rani*, *S*., *Kumar*, *P*., Machine-Learning, 2018, p. 3306.; Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[141] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[142] Cf. *Rani*, *S*., *Kumar*, *P*., Machine-Learning, 2018, p. 3306.
[143] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[144] Cf. *Rani*, *S*., *Kumar*, *P*., Machine-Learning, 2018, p. 3306.; Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[145] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[146] Cf. *ibid*., p. 62.
[147] Cf. *ibid*., p. 62.
[148] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 62.
[149] Cf. *ibid*., p. 62.
[150] Cf. *ibid*., p. 63 -64.
[151] Cf. *ibid*., p. 63 -64.

In summary, both approaches can achieve the objective defined in section 1.2. The DL approach has higher accuracy[152] and cognitive ability[153] than the traditional ML approach. However, the problem with this approach lies in the vast amount of data that an algorithm needs to mimic the human eye.[154] Consequently, more computing power[155] and more powerful hardware are required.[156] On the other hand, the traditional ML approach requires less data,[157] computational power,[158] and time to build the model.[159] In addition, the algorithms can be trained and executed on conventional computers.[160] These advantages create an ideal foundation for prototype development. Therefore, the traditional approach will be chosen for the development of the prototype.

[152] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[153] Cf. *ibid*., p. 62.
[154] Cf. *ibid*., p. 61.
[155] Cf. *ibid*., p. 61.
[156] Cf. *ibid*., p. 62.
[157] Cf. *Rani*, *S*., *Kumar*, *P*., Machine-Learning, 2018, p. 3306.
[158] Cf. *ibid*., p. 3306.; Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 61.
[159] Cf. *Silaparasetty*, *N*., ML Workflow, 2020, p. 63 -64.
[160] Cf. *ibid*., p. 62.

3 The prototype - Requirements analysis and mockup

3.1 Research method

This research aims to prove that technical exercises in soccer can be measured in an automated way through CV. Qualitative and quantitative research offers the possibility to bring theoretical solutions to light. Nevertheless, these approaches leave much room for interpretation for the final proof of feasibility. This makes the research question more difficult to answer, so qualitative and quantitative research are not considered. Instead, to establish clear proof of concept, a prototype must be developed that evaluates a technical exercise in soccer in an automated way. This approach also requires a theoretical elaboration of possible solutions that are subsequently implemented and evaluated. Furthermore, the prototype creates a basis that can be used for possible further developments. Therefore, the chosen research method of this thesis is the source code implementation of a prototype.

To make the source code easier to understand for third parties, Robert Martin's book Clean Code from 2008 serves as the basis for the source code implementation.[161] This book provides coding conventions that are recognized as a worldwide standard for every programming language.[162]

According to this, for source code implementation, particular attention is paid to ensuring that the name of a class, variable, and function indicates its purpose, task, and use.[163] Nevertheless, not all recommendations are taken into account. Although Martin claims, "The proper use of comments is to compensate for our failure to express ourselves in code"[164] and recommends using comments sparingly[165], many comments are used for the following reasons. Even if naming conventions are self-explanatory from the programmer's point of view, this does not automatically apply to third parties. An informative comment coupled with deliberately chosen naming conventions increases the likelihood that third parties will interpret the source code correctly.

---

[161] Cf. *Martin*, *R*., Clean Code, 2008, p. 1 et seqq.
[162] Cf. *ibid*., p. 1 et seqq.
[163] Cf. *ibid*., p. 18.
[164] *Ibid*., p. 54.
[165] Cf. *ibid*., p. 54.

Finally, the question, which programming language is the best for the prototype comes up. As already explained, the usage of ML and CV algorithms is required for prototype development. Both ML and CV belong to the sub-areas of AI.[166] The most common programming language for AI applications worldwide is Python, which means that numerous research, libraries, and documentation are based on Python.[167] Due to this, using Python is best suited for prototyping in this thesis.[168] Only for defining, editing, and querying data structures, the Prototype uses SQL, which is not further discussed in this Bachelor's thesis.

3.2 Prototype procedure

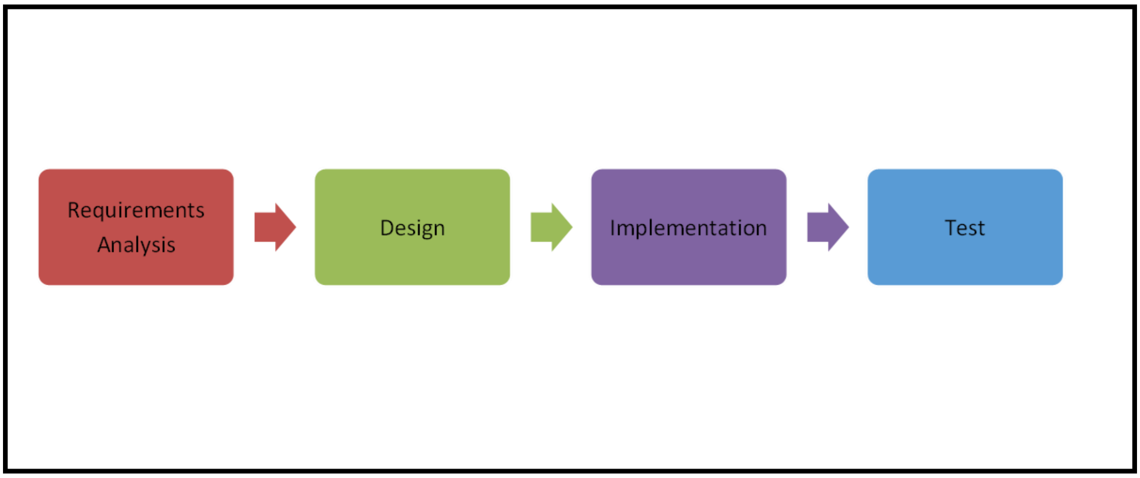
Each developed software system goes through a so-called software life cycle.[169] This is comparable to the life cycle of a human being.[170] The software system is born with the start of development and dies with shutdown, replacement, or further development.[171] Just the same applies to software prototypes.[172] For instance, if the prototype is removed after development, it is called a throw-away prototype.[173] However, if the prototype serves as the basis for the final software, the term evolutionary prototype is used.[174] In this case, the prototype to be created in this Bachelor's thesis should aim for the latter. Full-scale software development and software prototyping hardly differ from each other.[175] Only the integration to existing systems and subsequent start-up and maintenance that is part of full-scale software development is not required for a prototype.[176] Therefore, four phases of the waterfall model are adopted for the prototype procedure.[177] The waterfall model was developed by Dr. Winston Royce in 1970 and is the oldest process model in software development.[178] Its "waterfall" stands for the representation of the model.[179]

---

[166] Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
[167] Cf. *Bhasin*, *H*., Python Language, 2018, p. 16.
[168] Cf. *ibid*., p. 16.
[169] Cf. *Balzert*, *H*., Software Lifecycle, 2011, p. 1.
[170] Cf. *ibid*., p. 1.
[171] Cf. *ibid*., p. 1.
[172] Cf. *ibid*., p. 1.
[173] Cf. *Bell*, *D*., Prototype, 2005, p. 305.
[174] Cf. *ibid*., p. 307.
[175] Cf. *Kneuper*, *R*., Software Processes, 2018, p. 87–88.
[176] Cf. *ibid*., p. 87–88.
[177] Cf. *Aditya*, *M*. *P*., Waterfall, 2013, p. 70.
[178] Cf. *Kneuper*, *R*., Software Processes, 2018, p. 81–82.
[179] Cf. *Aditya*, *M*. *P*., Waterfall, 2013, p. 70.

The individual project phases are arranged as a cascade.[180] Thereby, the phases are processed sequentially from top left to bottom right. That means that there is no loop back to a previous phase.[181] Since loopbacks are necessary for error revision, the classic waterfall model is hardly used today.[182] Nevertheless, it illustrates the individual phases of a software life cycle quite well and thus serves as a reference point.[183] Four adopted phases from the classic Royce waterfall model are shown in the following Figure 7.

Figure 7: Prototype process model



Source: Own presentation

The four phases seen in the figure above are run through hereafter and used for the development of the prototype.

3.3 Prototype requirements

This section defines the requirements for the prototype to be developed. It uses a template that clearly presents the requirements for small software projects. More precisely, the "Schablone für das Lastenheft, Pflichtenheft und Glossar" by Balzert acts as a reference point.[184] Hence, the requirements consist of five categories. The first letters followed by numbers serve as abbreviations:[185]

/V1.0/ Vision of the project: What is the long-term goal of the project?

[180] Cf. *Aditya*, *M. P.*, Waterfall, 2013, p. 70.
[181] Cf. *Kneuper*, *R.*, Software Processes, 2018, p. 81–82.
[182] Cf. *ibid.*, p. 81–82.
[183] Cf. *Aditya*, *M. P.*, Waterfall, 2013, p. 70.
[184] Cf. *Balzert*, *H. et al.*, Requirements, 2010, p. 429 et seqq.
[185] Cf. *ibid.*, p. 429 et seqq.

/G1.0/ Goal of the project: What can be derived and operationalized from the vision for the project?

/L1.0/ Limitations of the project: What are the organizational and technical limitations of the prototype?

/F1.0/ Functional requirements: What functions are expected from the prototype?

/NF1.0/ Nonfunctional requirements: What other requirements are expected from the prototype?

Optional requirements are not implemented within the scope of this bachelor thesis. However, for the sake of completeness, they are listed in the requirements analysis marked with an "O". In addition, the optional requirements can also be taken into account for the further development of the prototype.[186] In the following, the requirements for the prototype are defined. Afterward, they serve as orientation for the development.

Project Vision:

/V1.0/ The prototype must serve as an aid for the automated measurement of technical skills in soccer. This must be demonstrated using the example of juggling.

/V2.0/ The prototype must be intuitive enough that any user can operate it on the first try.

/V3.0/ The prototype must create an objective basis, especially comparability for users.

/OV4.0/ The further development of the prototype must be used by soccer associations and clubs.

Project Goal:

/G1.0/ The prototype must evaluate ball juggling.

/G2.0/ The design must be ensured by a mockup.

/G3.0/ The prototype must ensure comparability through a ranking list.

/G4.0/ The prototype must be provided to test users as an EXE file.

[186] Cf. *Balzert*, *H*. *et al*., Requirements, 2010, p. 429 et seqq.

Project Limits:

/L1.0/ The prototype is limited to ball juggling in this bachelor thesis.

/OL1.1/ Further development of the prototype is to be focused on two more technical drills, e.g., passing and dribbling.

/L2.0/ The prototype must be designed for a specific soccer ball by default.

/L3.0/ The prototype ranking must be limited to five places for the sake of clarity.

Functional Requirements:

/F1.0/ The prototype must have a protected area and a public area.

/F1.1/ New users should be able to set their username and password right away through a registration function.

/F1.2/ Access is to be protected by a login function for registered users.

/F2.0/ The prototype must provide the user with a way to view and select all cameras connected to the device to perform the exercise.

/F2.1/ The prototype must allow users to upload previously recorded videos.

/F3.0/ The prototype must provide two modes for ball juggling.

/F3.1/ The first mode must be a free mode. In this mode, the application must measure the number of juggled ball contacts without any time limit.

/F3.2/ In the second mode, the number of contacts must be counted during ball juggling within a time window of thirty seconds. The timer must be resettable via a button.

/F4.0/ Every user must understand that the ground level is the boundary between the instep of the foot and the ground. It represents the ground and is essential for counting the ball contacts during juggling.

/F4.1/ To enable a user to test the prototype on different devices, in different environments, and with different camera perspectives, the ground level must be set manually.

/F4.2/ As soon as the soccer ball reaches the ground level, it means that the ball hit the ground, and the score must be transferred to the database and set to zero.

/F4.3/ The ground level must be highlighted by a transparent bar while juggling the ball.

/OF4.4/ The ground level should be adjusted automatically for the further development of the prototype. Thereby, the user's shoe should be recognized and serve as an upper limit.

/OF.5.0/ The application is intended for a specific soccer ball and preset for it by default. Nevertheless, it should be possible to use other balls for the application. For this reason, it must be possible to adjust the colors of the ball via the screen. However, this functionality will not be discussed in this thesis.

Nonfunctional Requirements:

/NF1.0/ The prototype must display the results in near real time.

/NF2.0/ The default language of the prototype must be English.

/ONF2.1/ For the further development of the prototype, users should be able to choose between English and German.

3.4 Mockup as design

The requirements for the prototype can be derived from the previous section. These are to be integrated into a design prototype in the following.

3.4.1 Design tool

A design prototype represents the graphical user interface, hereafter abbreviated GUI, of the desired result.[187]

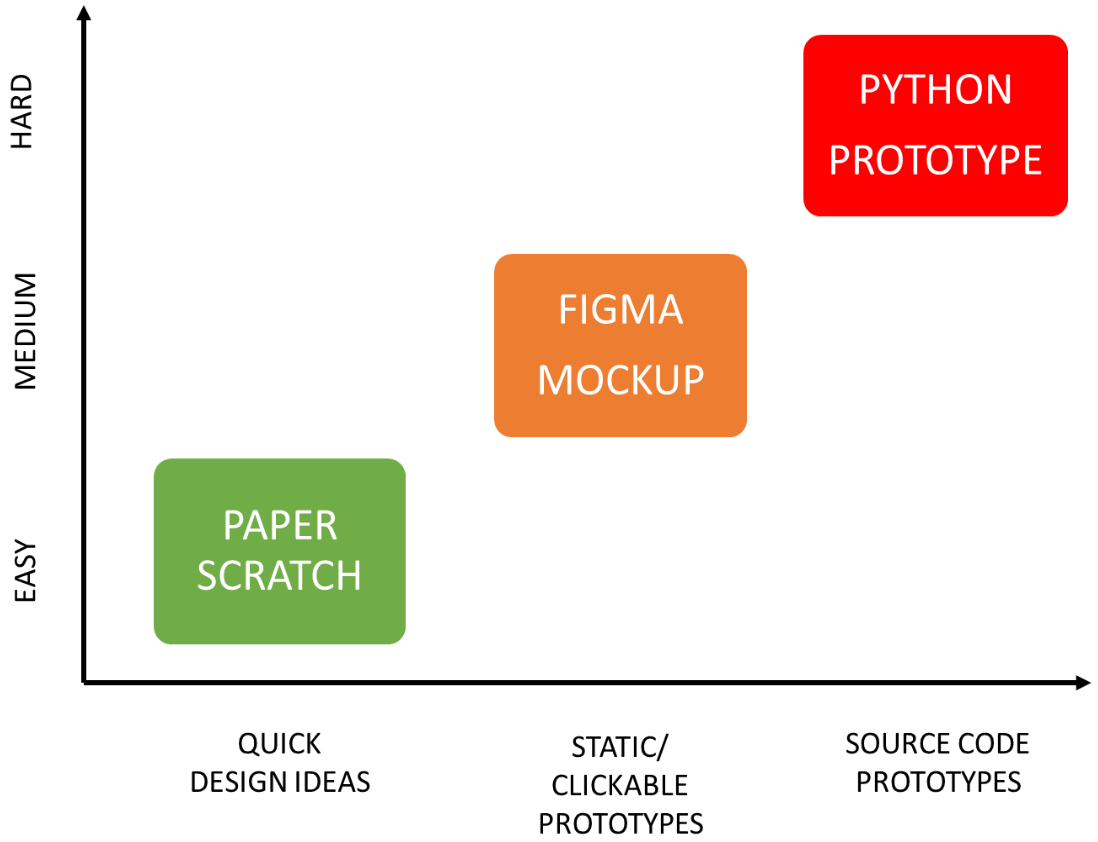
In principle, a distinction is made between three different types of design prototypes.[188] The simplest and fastest option is a paper sketch.[189] Followed by static and clickable design drafts, also summarized under the term mockup.[190] Finally, up to the most complex possibilities in the form of programmed frontend prototypes, for instance, in the programming language Python.[191]

---

[187] Cf. *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 52.
[188] Cf. *ibid*., p. 52–53.
[189] Cf. *ibid*., p. 52–53.
[190] Cf. *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 52 – 53.
[191] Cf. *ibid*., p. 52–53.

In this section, a clickable mockup is created as a design draft. The Figma software is used for this purpose. Figma enables the creation, editing, and sharing of user-defined designs in a web-based solution. Thereby, linking static designs and enabling user interactions is possible.[192] Figure 8 illustrates the design possibilities concerning complexity.

Figure 8: Design options



Source: Modelled after *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 52.

Another advantage of the Figma mockup is that the previously defined requirements can be directly converted into a UI design and quickly modified.[193] In addition, it offers the possibility to illustrate ideas to third parties visually and to interact with them.[194] This saves costs and time.[195] The mockup designs are presented in the following

[192] Cf. *Figma Inc*., UX, 2021, no page number.
[193] Cf. *ibid*., no page number.
[194] Cf. *Coleman*, *B*., *Goodwin*, *D*., Prototype, 2017, p. 3–4.
[195] Cf. *Camburn*, *B*. *et al*., Design, 2017, p. 1.

subsection 3.4.2. They serve as the foundation for the Python prototype implemented in Chapter 4.

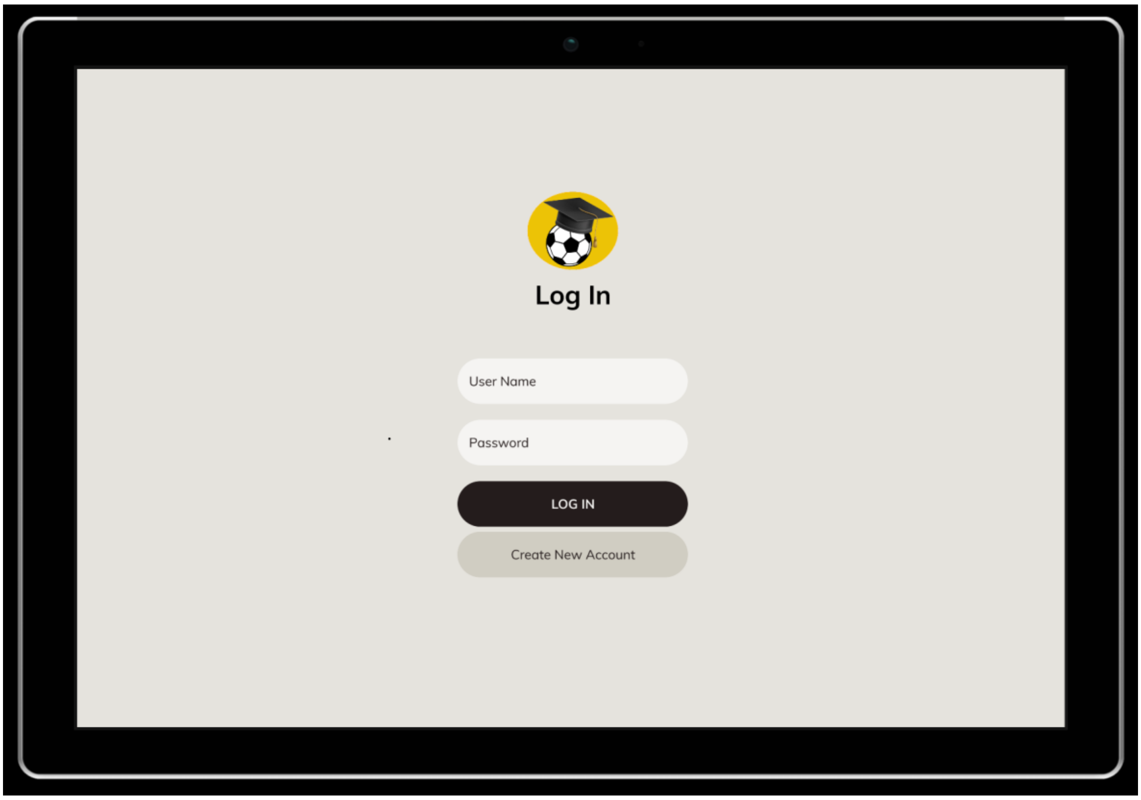
3.4.2 Interface structure

The following three subsections demonstrate the prototypes interface structure.

3.4.2.1 Login-and Register screen

Figure 9 and Figure 10 result from the requirements F1.0 - F1.3. As soon as a user opens the application, the following login screen should be displayed.

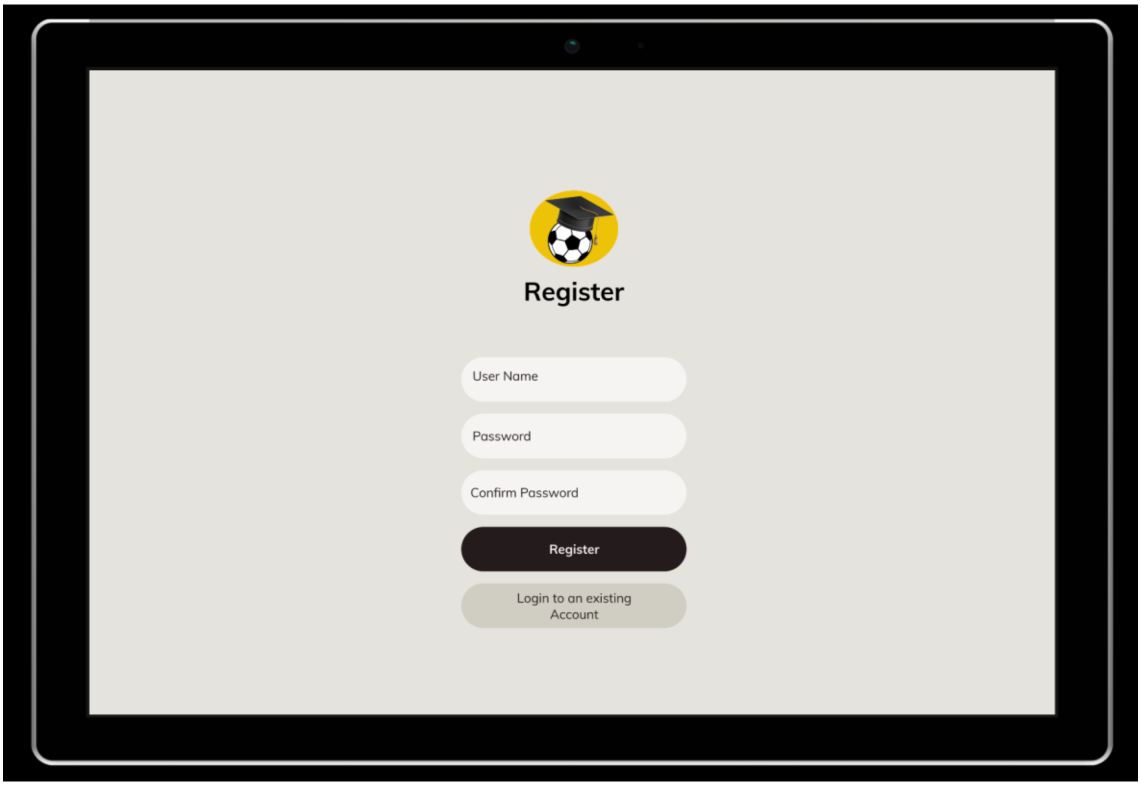
Figure 9: Login screen



Source: Own presentation

Through the Login screen, Registered users can access the protected area with their username and password. If a user has not registered yet, the register screen from Figure 10 can be accessed via the "Create New Account" button.

Figure 10: Register screen



Source: Own presentation

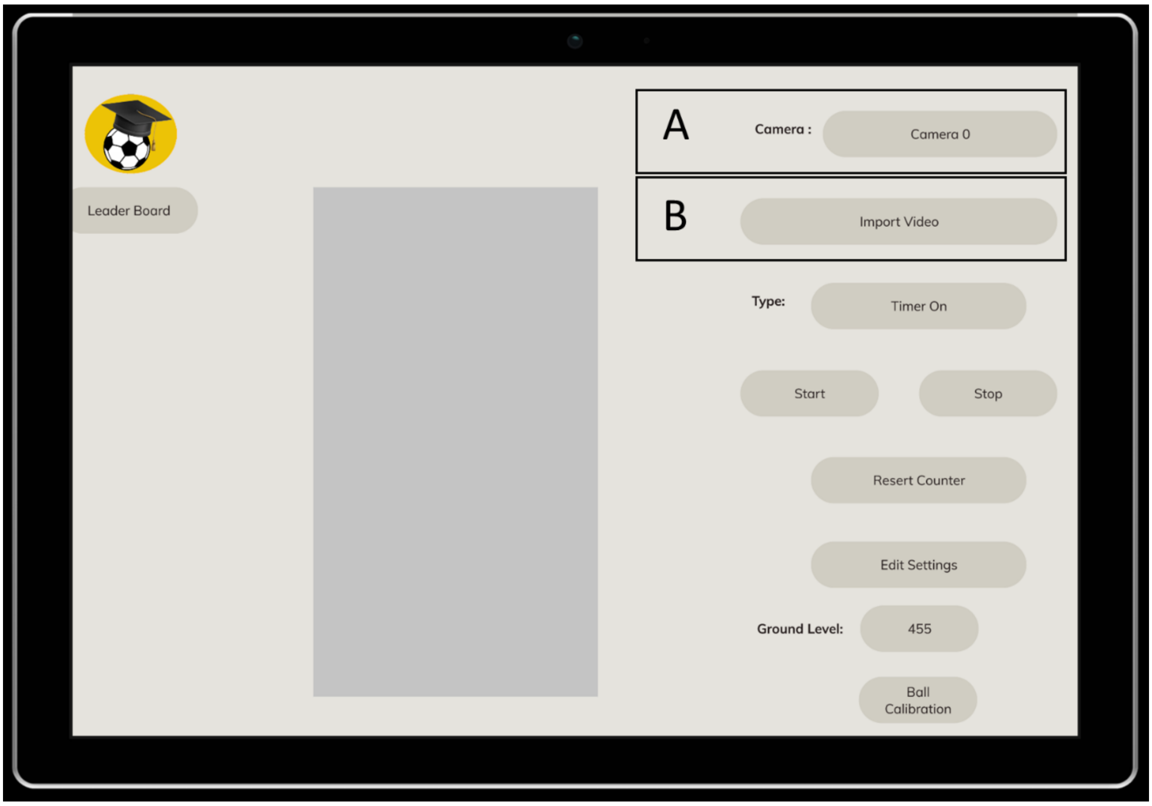
Through the Register screen, a username and password can then be set.

3.4.2.2 Ball juggling

The following figures result from the requirements /F2.0/ - /F4.4/ with the exception of /F3.1.1/ and /F3.2.2/.

Figure 11 shows the first screen of the protected area that becomes visible after logging in.

Figure 11: Home screen

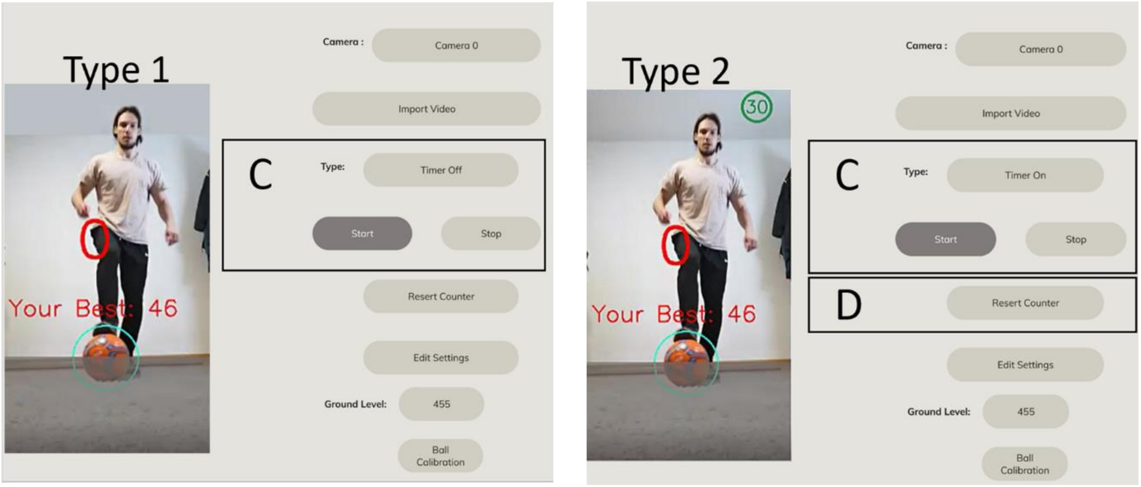


Source: Own presentation

Button A and B refer to the requirements /F2.0/ and /F2.1/. After that, a dropdown menu accessed via button A shows the user all connected camera systems of a device. As default, the application accesses camera 0 during initialization. Finally, button B allows the user to upload previously recorded video.

The requirements /F3.0/, /F3.1/, /F3.1.2/, /F1.1.3/, /F3.2/ and /F3.2.1/ are implemented according to the following Figure 12.

Figure 12: Types and counter

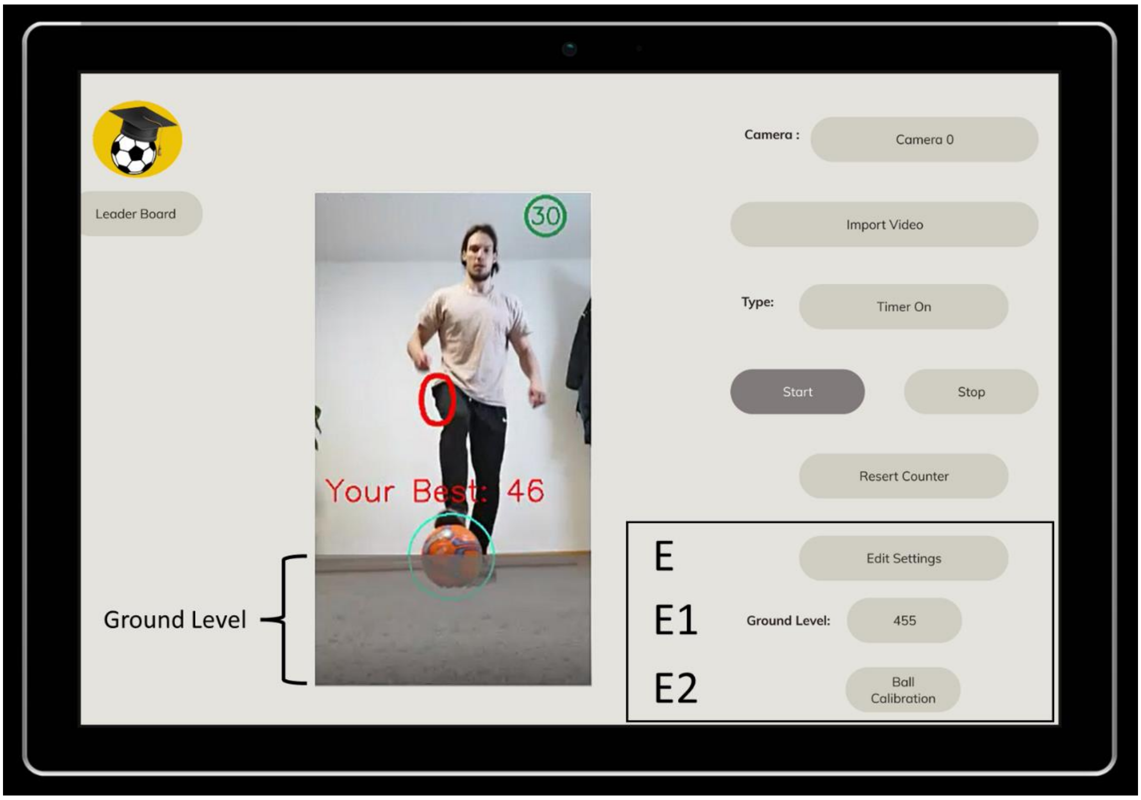


Source: Own presentation

Button C shows the current mode of the application. Here, a distinction between the two types, "Timer off " and "Timer on," is possible. The first type, shown on the left, measures ball juggling without a time limit. A user's best attempt is permanently displayed on the screen and serves as an incentive for personal bests. The type can be paused and started via the "Start" and "Stopp" buttons in area C. The second type, shown on the right, can be selected via a dropdown menu. The mode aims to simulate juggling under a pressure situation. If the user starts the type, a time window of thirty seconds appears, in which the ball contacts are counted. The time is displayed as a countdown in the upper right corner, as shown in the figure. Suppose a user wants to restart a try, for example, due to several failed attempts. In that case, the time is resettable to thirty seconds via button D.

Figure 13 shows the integration of the requirements /F4.0/ to /F4.4/ into the design. Again, especially the relevance of the ground level comes to the fore here.

Figure 13: Ground level



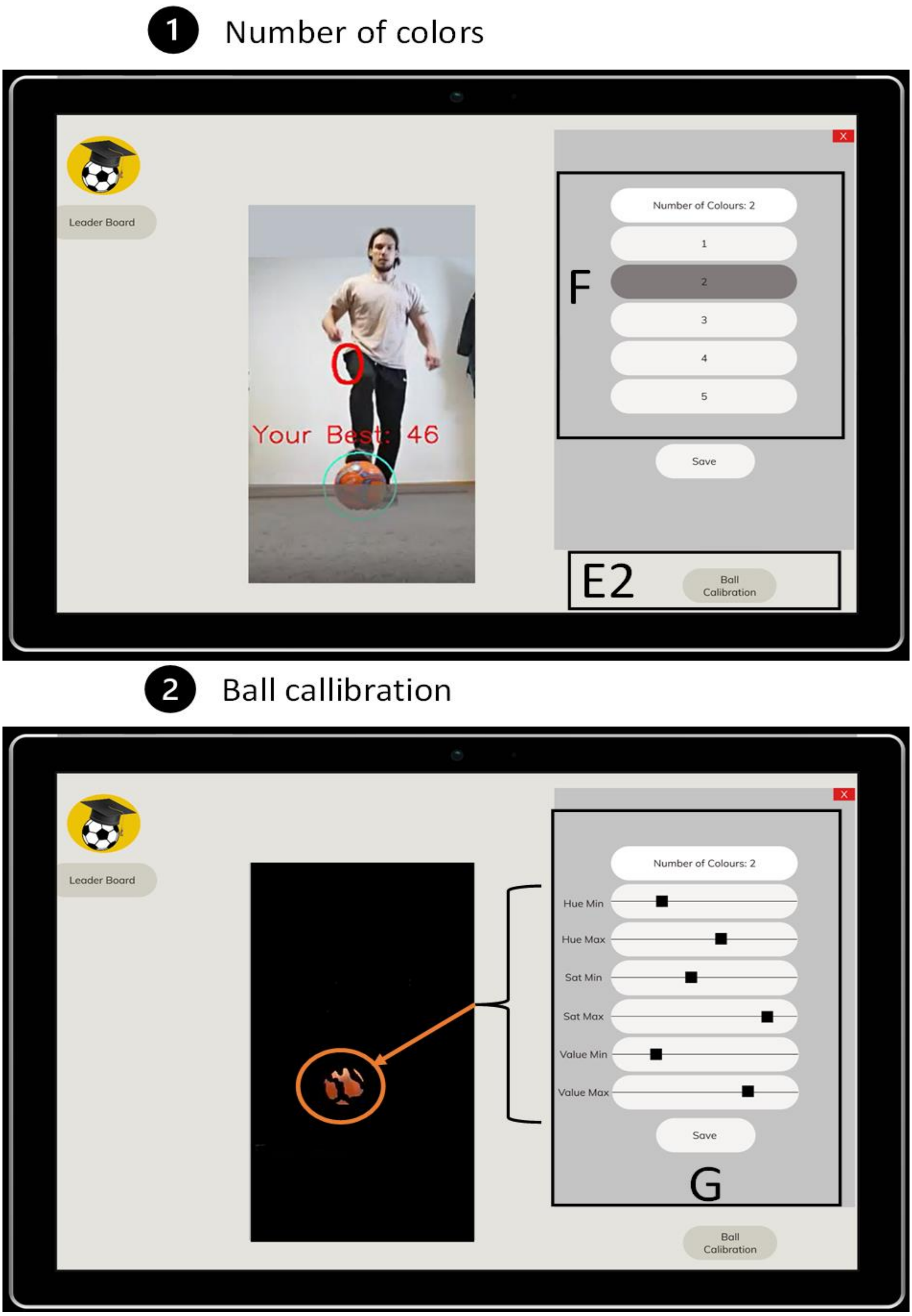
Source: Own presentation

The ground level, visible in gray in the figure, simulates the ground of an environment. If the ball lands within this level, this is equivalent to a failed attempt. In this case, the score must be reset to null. Ideally, the ground level is halfway up the ball. By pressing the button E, the buttons E1 and E2 must be displayed. E1 shows the height of the ground level in pixels. Users with different devices and distances to the camera should be able to change the height manually by clicking in the field E1. In this example, the height of the ground level is set to 455 pixels. E2 is only labeled for completeness and is explained in more detail in the following subsection.

3.4.2.3 Color change of the ball

The scenario in Figure 12 results from the requirement /F.5.0/ and explains the color change of the ball. This is not intended for everyday use and is only relevant for the user in exceptional cases. Examples of exceptional cases include the loss or damage of the ball. Therefore, the corresponding source code implementation is left out.

Figure 14: Color change of the ball



Source: Own presentation

The color change takes place in a two-step process. First, the pop-up field F opens by pressing the "Ball Callibration" button E2. Then, the user can select the number of colors the desired ball has. The fewer colors a ball has, the better results occur during the exercise execution. In this example, the ball consists of two colors, orange and blue. As soon as the user saves the two colors, the field G opens. There a user defines the upper and lower limit of the two ball colors in a HSV color space. HSV stands for the three coordinates hue, saturation, and value in a color model. In this color model, every color value can be represented.[196] "Since the hue channel models the color type, it is very useful in image processing tasks that need to segment objects based on its color."[197] A slider allows the corresponding color tones to be highlighted directly in the image. Figure 14 demonstrates how the slider works, using the orange color of the ball as an example. The HSV values are set and saved for all colors that the ball possesses. Afterward, the prototype is changed to the corresponding ball and usable for juggling.

[196] Cf. *Burger*, *W*., *Burge*, *M*., HSV, 2010, p. 100–101.
[197] *OpenCV*, Object Detection, 2021, no page number.

4 Source code implementation

4.1 Used libraries

The following subsections look at the three most crucial Python libraries used for implementing the prototype.

4.1.1 OpenCV

The Open Source Computer Vision Library, also known as OpenCV, is the core software library of this prototype.[198] It supports the creation of CV and ML applications by providing more than 2500 algorithms.[199] More than 47000 users, including companies such as Google, Microsoft, and Intel, use the library for various projects.[200] Use cases range from monitoring mining equipment in China, to detecting accidents in swimming pools, to facial recognition in Japan.[201]

During the prototype development, the OpenCV algorithms support, in particular, the two most important functionalities. These include the detection and tracking of the ball. Each line of source code indicated by the keyword "cv2." accesses the OpenCV library.

4.1.2 NumPy

The Numeric Python Library is abbreviated as NumPy and provides algorithms that simplify arithmetic with matrices, arrays, and vectors.[202] Explicit application of Numpy occurs in the definition of the color range of the ball.[203] The corresponding source code is indicated with the keyword "np.".[204] Nevertheless, it must not remain unmentioned that the library OpenCV explained before uses NumPy arrays in the background to process image data.[205] Thus NumPy is also an essential part of the prototype, though it is often not indicated.[206]

198 Cf. *OpenCV*, About, 2020, no page number.
199 Cf. *ibid*. no page number.
200 Cf. *ibid*. no page number.
201 Cf. *ibid*. no page number.
202 Cf. *OpenCV*, Numpy, 2021, no page number.
203 Cf. *ibid*. no page number.
204 Cf. *ibid*. no page number.
205 Cf. *ibid*. no page number.
206 Cf. *ibid*. no page number.

4.1.3 Tkinter

Tkinter is a Python library used to create GUIs.[207] All interfaces of the prototype, such as windows, buttons, and text fields, are created with the help of Tkinter.[208] In particular, the tkinter.ttk module, which provides widgets, is frequently used.[209]

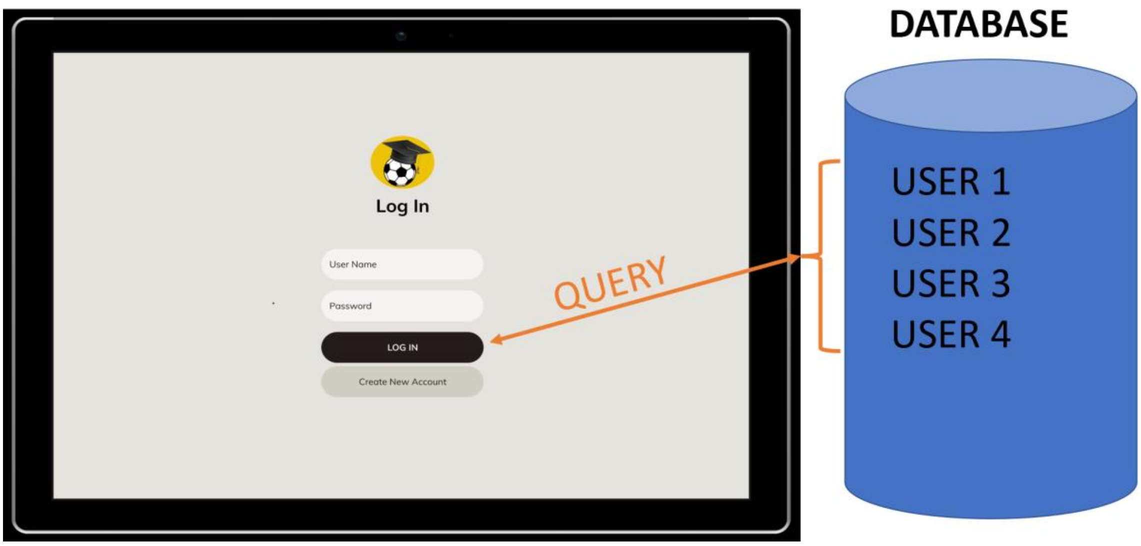
4.2 Source code implementation

The following subsections explain the source code implementation of the login, registration and main application.

4.2.1 Login

This section implements the login screen from Figure 11. On the one hand, a graphical interface with input fields, images, and buttons allows interaction with the user. On the other hand, a database stores relevant data for the user login. These are queried during a login. Figure 15 demonstrates the interaction between the GUI and the database.

Figure 15: GUI and database



Source: Own Representation

The login area is created with Python and MySQL. In order to display elements like buttons, texts, input fields, and images on the interface, the Python GUI library

[207] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[208] Cf. *ibid*. no page number.
[209] Cf. *ibid*. no page number.

Tkinter serves.[210] Finally, for database creation, SQLite 3 is applied, which will not be illustrated in detail.[211] Both are pre-installed with the Python installation.[212]

For the login start screen the three methods, "__init__()", "progstart()" and "register()" are implemented in the class Login. A class allows creating its own data type.[213] Thus, several objects of this class can be created. Associated methods, variables, and attributes are grouped under the class.[214]

4.2.1.1 The __init__() method

The first method ensures the implementation of elements that are visible on the interface. A section of the source code of the first method is visible in Listing 1.

Listing 1

```
# login window class
(1)class Login:
(2)   def __init__(self):     #intialize the GUI window
(3)       self.root = tki.Tk()
(4)       self.root.title("Log In")                # set window title
(5)       self.root.geometry("600x500+300+50")        # set window geometry
(6)       self.root.iconbitmap('Data\Img\Logo_32.ico') # set Icon
(7)       self.root.resizable(0,0)                  # disable resizing
(8)       self.root.configure(background='#4D4D4D')
(9)       s=ttk.Style(self.root)
(10)      s.theme_use('xpnative')                    # set the window theme
```

Source: Own presentation

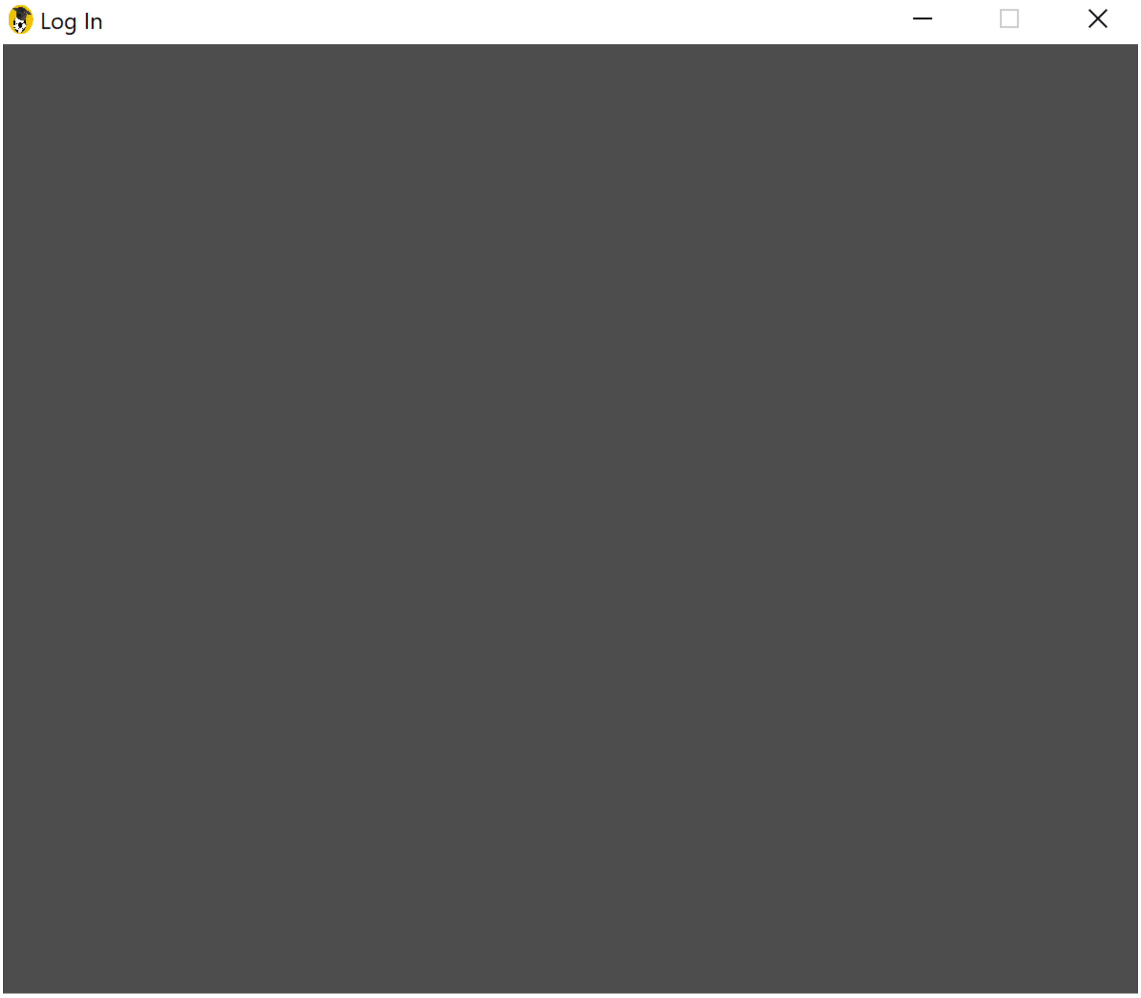
Line 1 defines the Login class.[215] The "__init__ ()" method in line two has an important function. "Informally, the __init__ instance method of a class is called its constructor since its semantics is similar to "that" of a Java constructor. However, a fitting name for __init__ is initializer, since it does not actually build the instance, but receives it as its self-argument."[216] This means the "__init__()" method in line two is

[210] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[211] Cf. *Python Software Foundation*, Sqlite3, 2021, no page number.
[212] Cf. *ibid*. no page number.; Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[213] Cf. *Broy*, *M*., *Malkis*, *A*., Programming, 2019, p. 384 – 385.
[214] Cf. *ibid*. p. 384 – 385.
[215] Cf. *Python Software Foundation*, Classes, 2021, no page number.
[216] *Ramalho*, *L*., Init Method, 2015, p. 719.

automatically called whenever an object of the Login class is instantiated.[217] The self parameter passed in the method returns the reference of the executing object.[218] Since several objects of a class can be created and the values of the attributes are partly different, the self argument is essential in order to be able to assign executing objects exactly.[219] In the following lines, the attributes are defined, which characterize the login window within the program.[220] This includes the Tkinter root widget, which is created in line three by calling "tki.Tk()".[221] The root widget forms the first level at which all other elements are created.[222] Lines four through ten demonstrate customizations to the root widget.[223] Besides the title in line four, the window size is defined in line five.[224] According to line seven, the window size cannot be reduced or enlarged.[225] Line six places a logo as an iconbitmap.[226] In line nine and ten, the Tkinter ttk module is accessed, through which prefabricated window themes can be selected.[227] The root window is now created and visible in Figure 16.

[217] Cf. *Ramalho*, *L*., Init Method, 2015, p. 719.
[218] Cf. *Python Software Foundation*, Self-Argument, 2021, no page number.
[219] Cf. *ibid*., no page number.
[220] Cf. *Python Software Foundation*, Classes, 2021, no page number.
[221] Cf. *Python Software Foundation*, Interface, 2021, no page number.
[222] Cf. *ibid*., no page number.
[223] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[224] Cf. *ibid*., no page number.
[225] Cf. *Python Software Foundation*, Frontend, 2021, no page number.
[226] Cf. *ibid*., no page number.
[227] Cf. *ibid*., no page number.

Figure 16: Root widget



Source: Own presentation

In Listing 2, the login and regstrier button is implemented, as well as the logo on the root window.

Listing 2

```
     # put logo img in its location
(1)   canvas = Canvas(self.root, width = 120, height = 80,
(2)   background='#4D4D4D',highlightthickness=0)
(3)   img = PhotoImage(file="Data\Img\Logo2.png")
(4)   canvas.create_image(0,0, anchor=NW, image=img)
(5)   canvas.place(x = 10, y = 10, width = 120, height = 80)

     # create the login button and assign it to the progstart method
(7)   btn = tki.Button(self.root, text= "Log In", foreground="white",background='gray55',  (
8)    font = ('Arial', 16),command=self.progstart)
(9)   btn.place(x = 250, y = 300, width = 100, height = 50)

     # create the Create new account button and assign it to the register method
(10)  btn2 = tki.Button(self.root, text= "Create New Account", foreground="white",
(11)  background='gray55', font = ('Arial', 16), command=self.register)
(12)  btn2.place(x = 175, y = 400, width = 250, height = 50)
```

Source: Own presentation

Lines one to six show the logo from Figure 9 placed on the canvas. The first two lines specify the size, background color, and highlighting of the canvas.[228] "A canvas is a rectangular area intended for drawing pictures or other complex layouts. On it, someone can place graphics, text, widgets, or frames."[229] To place the logo on the canvas, line three loads the desired logo from a local folder.[230] The "create_image()" instance is used in line four to put the logo on the canvas.[231] The instance needs x and y coordinates set to null because the image can be placed to the top left corner using the anchor.[232] The third parameter defines the image that should be used.[233] Finally, line five determines the position of the image canvas within the parent widget.[234]

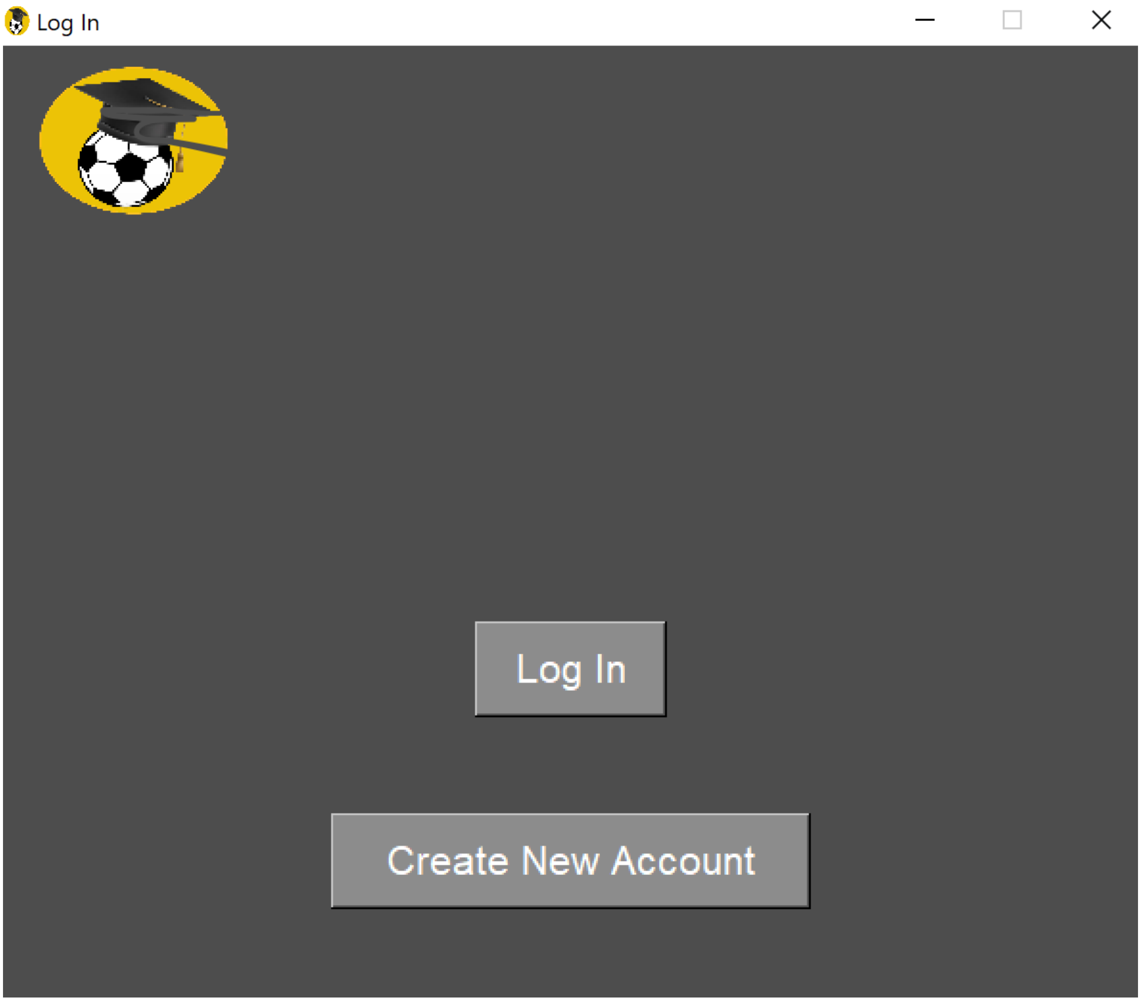
[228] Cf. *Shipman*, *J.*, Canvas Image, 2013, no page number.
[229] *Shipman*, *J.*, Canvas Widget, 2013, no page number.
[230] Cf. *Shipman*, *J.*, Images, 2013, no page number.
[231] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[232] Cf. *ibid.*, no page number.
[233] Cf. *ibid.*, no page number.
[234] Cf. *Python Software Foundation*, Frontend, 2021, no page number.

In lines seven to nine, the login button gets created and aligned. The use of the "command=self.progstart" points to the "progstart()" method that is to be called using the button.[235] This method compares login data with existing database entries and will be explained in more detail later on.

The implementation of the register button is shown in lines ten to twelve.[236] It is used to redirect users to a new window, where they can create a new account.[237] This is technically done by the "command=self.register" which refers to the register method.

The result from the explained source code is visible in the following Figure 17.

Figure 17: Login



Source: Own presentation

According to section 3.4.2, the user name, password, and page title are input fields implemented in Listing 3.

[235] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.
[236] Cf. *ibid*., no page number.
[237] Cf. *ibid*., no page number.

Listing 3

```
    #create the used labels and set its location
(1)     label1 = tki.Label(self.root,foreground="#D6D6D6",background='#4D4D4D', text =
(2)     'User Name',font=('Bold', 14)).place(x = 30, y = 100, width = 100, height = 60)
(3)     label2 = tki.Label(self.root,foreground="#D6D6D6",background='#4D4D4D',
(4)     text = 'Password',font=('Bold', 14)).place(x = 30, y = 200, width = 100, height = 60)
(5)     label3 = tki.Label(self.root,foreground="#D6D6D6",background='#4D4D4D',
(6)     text = 'Log In',font=('Bold', 18)).place(x = 200, y = 50, width = 300, height = 60)

        #create the used entries and set its location
(7)     self.entry1 = tki.Entry(self.root, width = 45, font = ('Arial', 12))    # user name entry
(8)     self.entry1.place(x = 150, y = 105, height = 50)
(9)     self.entry2 = tki.Entry(self.root, width = 45, font = ('Arial', 12))    # password entry
(10)    self.entry2.place(x = 150, y = 205, height = 50)
(11)    self.entry2.config(show="*")        # show password as *
```

Source: Own presentation

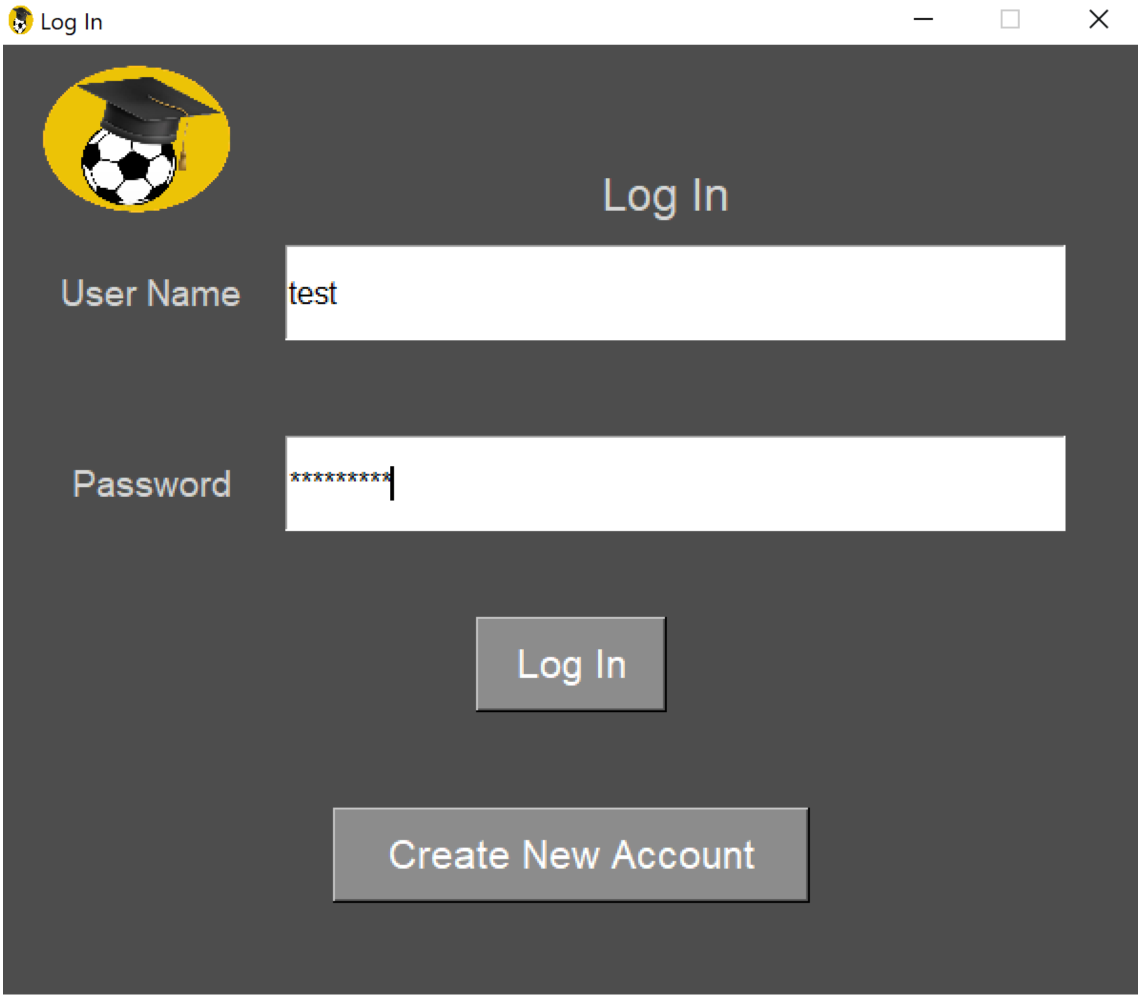
Lines one to six show how the titles "User Name," "Password," and "Log In" are created on the canvas. [238] Therefore, the label widget of the tkinter.ttk module is used.[239] A label is a Tkinter class that represents text or an image without interacting with them.[240] The procedure is identical for all three titles.[241] According to lines one and two, first, a variable, e.g., label 1, is assigned the label method.[242] Then, the method is passed the font color, background color, title, font, and font size as parameters.[243] In addition, the position and size are determined via the dot operator.[244]

The input fields associated with the username and password are part of lines seven to eleven. The Entry widget is used for this purpose.[245] In line seven, the input field, its width, font, and font type are defined.[246] The position and height of the input field are coded in line eight. The following lines repeat this analogously for the

[238] Cf. *Roseman*, *M*., Label, 2021, no page number.
[239] Cf. *ibid*., no page number.
[240] Cf. *ibid*., no page number.
[241] Cf. *Python Software Foundation*, Frontend, 2021, no page number.
[242] Cf. *ibid*., no page number.
[243] Cf. *ibid*., no page number.
[244] Cf. *ibid*., no page number.
[245] Cf. *ibid*., no page number.
[246] Cf. *Roseman*, *M*., Basic Widgets, 2021, no page number.

password,[247] with the particular case in line eleven, where it is specified that the characters of the password entry are to be replaced by asterisks.[248] Figure 18 shows the result of the previously described source code.

Figure 18: Final login



Source: Own presentation

Appendix 9 explains how to test the prototype and provides a user name and password for the login.

[247] Cf. *ibid*., no page number.
[248] Cf. *Python Software Foundation*, Password, 2021, no page number.

4.2.1.2 The progstart() method

The "progstart()" method is the second method of the login class. In this one, the previously made login entries are compared with the existing user data in the database. The source code is shown in Listing 4.

Listing 4

```
(1)  def progstart(self):
(2)      global user, vs          # define the used global variables
(3)      userName = str(self.entry1.get())      # get user namee
(4)      password = str(self.entry2.get())      # get password
(5)      conn = sqlite3.connect("Data\Db\JugDB.db") # intialize the Database connection

         # check if the user name and password exist in the table of users
(6)      query = "SELECT user_name FROM Users WHERE user_name = '"
(7)      + userName + "' and password = '" + password + "'"
(8)      cursor = conn.cursor()
(9)      cursor.execute(query)
(10)      records = cursor.fetchall()
(11)     if(len(records)==0):          # if user enters a wrong user name or password
            tkinter.messagebox.showerror(title="Error", message=
            "Wrong user name or password")
(12)     else:       #if the user name and password has a match in the table
(13)         user = username # save the user name into the global
             variable to use it in the other functions
(14)         self.root.destroy()  # close the login window
             # start the main window
(15)         pba = MainWindow(vs)
(16)         pba.root.mainloop()
```

Source: Own presentation

The creation of the "progstart ()" method is done in line one. Two global variables are created in line two.[249] These are valid for all classes and methods of a program.[250] Thus, for example, the username can be assigned to the global variable user and reused for other functionalities.[251] For instance, the username is necessary for the

[249] Cf. *Python Software Foundation*, FAQ, 2021, no page number.
[250] Cf. *ibid*., no page number
[251] Cf. *ibid*., no page number.

login and the leader board or registration. The expression "str(self.entry1.get())" from line three returns the user name entered for the login as a string.[252] After that, the user name is assigned to the variable "username".[253] The same procedure is repeated for the password in line four.[254]

Accessing the database requires the creation of a connection object. This represents the database. Here, line five demonstrates this using the local database file "JugDB.db". If the connection to the database is established, creating a cursor object corresponding to line eight is necessary. By calling its "execute()" method, SQL commands can be executed, as shown in line nine.[255] In lines six and seven, the SQL commands compare the column "user_name" of the database "Users" according to the data entered in the login.[256] By calling the method "fetchall()" of the cursor in line ten, the matching rows of the SQL statement are retrieved. These are linked to a condition that is implemented in lines eleven and twelve. If the username or password is not entered in the database, the error message "Wrong username or password" should be displayed. Otherwise, the "username" should be stored in the global variable "user".[257] In this case, the last three lines close the login window and open the main window.[258]

4.2.1.3 The register() method

It has already been explained in the "__init__()" method that the "register()" method is called up via the "Create New Account" button of the login window.[259] The source code of this method is visible in the following listing.

Listing 5

```
def register(self):   # if the user wants to create a new a account
        self.root.destroy()     # close the login window
        app=Register()          # open the register window
        app.root.mainloop()
```

Source: Own presentation

[252] Cf. *Python Software Foundation*, Frontend, 2021, no page number.
[253] Cf. *ibid*., no page number
[254] Cf. *ibid*., no page number
[255] Cf. *Python Software Foundation*, Sqlite3, 2021, no page number.
[256] Cf. *SQLite*, SQLite, 2021, no page number.
[257] Cf. *Python Software Foundation*, Sqlite3, 2021, no page number.
[258] Cf. *Roseman*, *M*., Window, 2021, no page number.
[259] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.

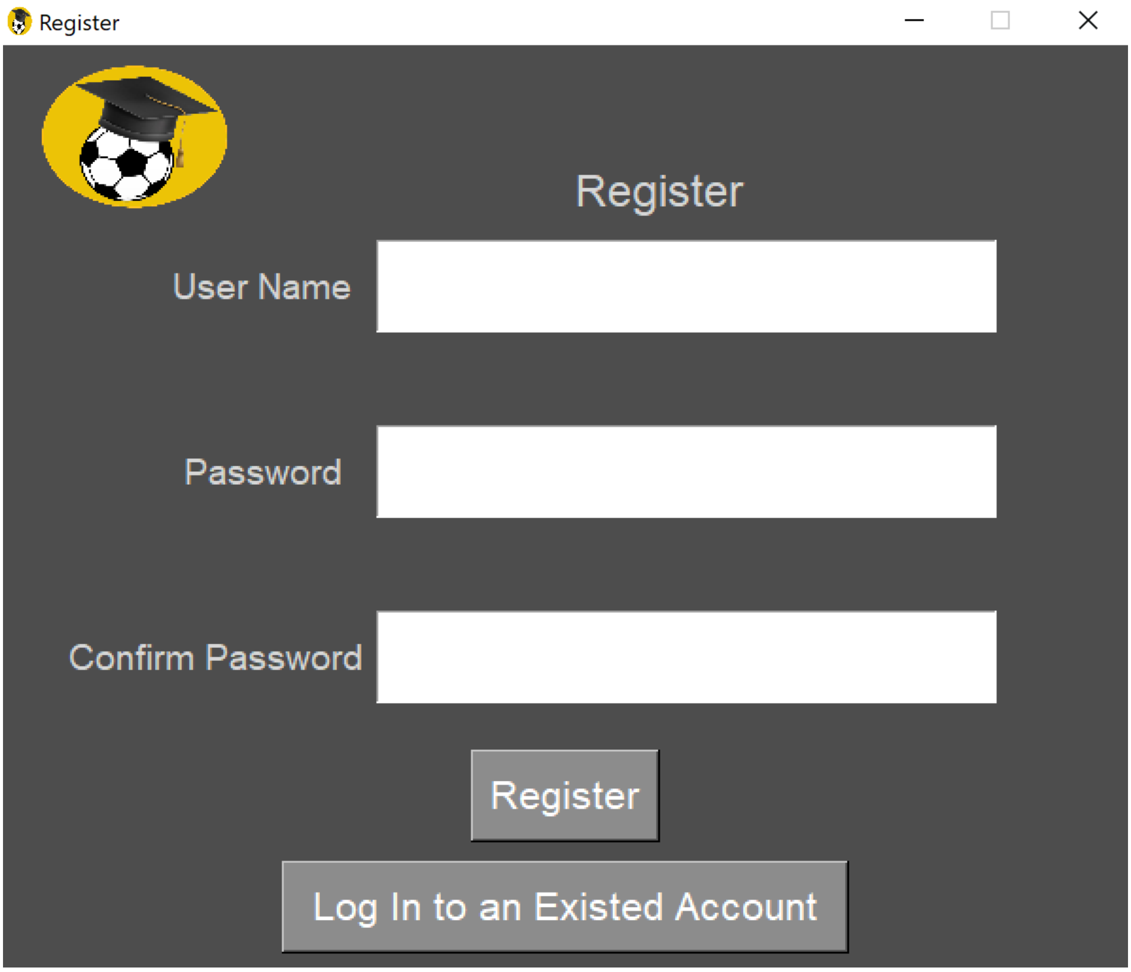
Within the body of the method, the login window is first closed in line one. Afterward, the class "register" is opened, and its code is executed following lines two and three.

4.2.2 Registration

The class "register" consists of three methods, "__init__()", "progstart()" and "login()". In these methods, instructions from the "login" class are repeated. Therefore the source code is not explained in detail.

In the "__init__()" method, the same elements from the "login" class that form the frontend are implemented.[260] These include titles, buttons, input windows, logos, background colors, and windows.[261] The resulting frontend appears in Figure 19.

Figure 19: Register window



Source: Own presentation

[260] Cf. *Python Software Foundation*, Interface, 2021, no page number.
[261] Cf. *ibid*., no page number

The "progstart()" method forms the interface to the database. The username and password are thereby stored in the database after registration and queried during a login attempt. Finally, "the login()" method is called by pressing a button. It closes the current window and automatically executes the user login.

4.2.3 Main application

Once a user has successfully logged in, the main application starts, implemented in the class "MainWindow". In order not to exceed the scope of this bachelor thesis, only the most essential source code sections are examined in more detail below. This includes source code sections that are relevant to the functionality of the application. The entire source code, including the EXE file and the prototype manual, is attached to appendix 6–8.

4.2.3.1 Main __init__() method

Following the same procedure of the "__init__()" method as in the class "Login," all elements visible on the screen are implemented using the Tkinter module.[262] That includes all the visible elements from Figure 20 below.

Figure 20: Main screen

Source: Own presentation

[262] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.

In order to keep redundancies as low as possible, the source code for the frontend elements from the previous figure is not explained. However, the init method also has other important purposes. For example, it initiates the video source, the application window, the video thread, and the thread's stop event. Threading, in particular, has an important role to play and is explained in more detail.

The Python threading module allows to create, control, and manage threads in Python.[263] This enables the apparent simultaneous execution of multiple tasks.[264] The prototype uses the "Event" and "Timer" object types defined by the threading module.[265]

The Event object controls communication between threads.[266] Thus, an event is signaled by one thread while all other threads are waiting for the event.[267] Here, the Event object is used to stop the video threads.[268] Timer objects allow executing actions with a specified expiration time.[269] Thus, own threads can be created that run almost in parallel.[270]

To better understand the topic, the following Listing 6 shows how the prototype uses the threading module. First, it should be mentioned that the source code consists of different sections. The character string "- - -" separates the individual sections.

[263] Cf. *Python Software Foundation*, Threading, 2021, no page number.
[264] Cf. *ibid*., no page number.
[265] Cf. *ibid*., no page number.
[266] Cf. *ibid*., no page number.
[267] Cf. *ibid*., no page number.
[268] Cf. *ibid*., no page number.
[269] Cf. *ibid*., no page number.
[270] Cf. *ibid*., no page number.

Listing 6

```
(1) import threading
    - - - - - - - - - - - - - - - - - - - - - - - - -
    class MainWindow:
      def __init__(self, vs):
(2)       self.vs = vs                #initialize the video source
          self.frame = None           #initialize the window frame
          self.thread = None          #initialize the video thread
          self.stopEvent = None       #initialize the stop event  of the thread
(3)       self.stopEvent = threading.Event()  #create stop event to stop the video thread
(4)       self.thread = threading.Timer(1,self.videoLoop)
#create the video thread and assign it to video loop method
       def startVid(self):
         - - - - - - - - - - - - - - - - - - - - - - - - -
(5)          self.thread.start()          #start the video thread
```

Source: Own presentation

A thread is a flow line in a process and has a particular task. For example, a so-called main thread is part of every application and executes the main method.[271] However, an application itself can also start threads that independently execute tasks seemingly in parallel.[272] The threading module in Python is supposed to poll the frames of a video source to a separate file through a loop. Hence, processing the frames of a video, e.g., for object detection, is possible.

In the first line, initially, the threading module is imported to be able to create threads.[273] Then, in the second line, the video source, the resulting frame, the thread, and the stop event are initialized by the "__init__()" method.[274] The stop event defines conditions for the thread to stop processing and is created in line three.[275] It is implemented several times in the source code because the thread must interrupt the execution at every setting made during the juggling or a mode change.

On the other hand, the thread also needs a start event defined in line five.[276] This event is triggered whenever the user starts a mode via the start button.[277] According to line 4, a thread is created inside this mode that is passed to the video Loop method

[271] Cf. *Abts*, *D*., Thread, 2018, p. 277–288.
[272] Cf. *ibid*., p. 277–288.
[273] Cf. *Python Software Foundation*, Threading, 2021, no page number.
[274] Cf. *ibid*., no page number.
[275] Cf. *ibid*., no page number.
[276] Cf. *ibid*., no page number.
[277] Cf. *ibid*., no page number.

after one second.[278] There the thread processes the frame of the video source in a loop.[279]

4.2.3.2 Capture video from camera

In order to count the number of juggled ball contacts, the camera must first be accessed, or an already recorded video needs to be inserted. The following method in Listing 7 displays all available cameras of a device and returns them in a list.[280]

Listing 7

```
   #this method is used to return all  cameras from the program
(1)     def returnCameraIndexes(self):
   # checks the first 10 indexes.
(2)         index = 0
            arr = ["Video"]     #import video file option
            i = 10              #maximum numbers of the attached cameras
   #append each camera with its id and return the list
(3)         while i > 0:
                cap = cv2.VideoCapture(index)
                if cap.read()[0]:
                    arr.append("Camera " + str(index))
                    cap.release()
                index += 1
                i -= 1
(4)         return arr
```

Source: Own presentation

The method name "returnCameraIndexes()" is defined in the first section. The second section defines the variables that display the cameras.[281] "I = 10" represents the maximum number of cameras that can be displayed.[282] Afterward, the variable "index" defines the currently selected camera. At the same time, the variable "arr" provides the option to return video files as camera alternatives in the list.[283] In the third section, the available cameras are displayed and output by a while loop.[284] The programming library OpenCV is used for this.[285] More precisely, a VideoCapture

[278] Cf. *Python Software Foundation*, Threading, 2021, no page number.
[279] Cf. *ibid*., no page number.
[280] Cf. *OpenCV*, Capture Video, 2021, no page number.
[281] Cf. *ibid*., no page number.
[282] Cf. *ibid*., no page number.
[283] Cf. *ibid*., no page number.
[284] Cf. *ibid*., no page number.
[285] Cf. *ibid*., no page number.

object is created whose argument is the index variable.[286] "Camera 0" is the name of the first camera to be displayed. Then the index is counted up, and the run variable "i" is counted down. Thus the second camera is displayed as "Camera 1" and so on. Once no other camera is present, the "Video" option from line four is added to the list. The video file selection, which is done by the button "Import Video," can be taken from the method "openFile()" of appendix 7.

4.2.3.3 Ball detection

After the camera is active, the ball is to be detected in the video and outlined with a circle. The python libraries NumPy and OpenCV are used for this purpose.[287] A video is a series of frames strung together.[288] Therefore, the functionality is applied to a single video frame and run over and over again until a stop event is triggered.[289]

The traditional approach for object recognition described in section 2.5.1 now applies.[290] In other words, the ball is recognized by its color and shape.[291] Reducing the noise of a frame speeds up the processing time and is visible as the first step of ball detection in Listing 8.[292]

Listing 8

```
#blur the frame and convert it to the hsv format
(1)  imgBlur = cv2.GaussianBlur(self.frame, (7, 7), 1)
(2)  hsv = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2HSV)
```

Source: Own presentation

Noise in an image creates edges that are not present in reality.[293] The Gaussian filter from line one blurs the image, so only actual edges are visible.[294] The RGB color space encodes colors with only three color channels. [295] Therefore, objects in the image are difficult to identify.[296] For this reason, the RGB color space is converted to the HSV color space in line two.[297] It defines a color by its hue, saturation, and

---

[286] Cf. *OpenCV*, Capture Video, 2021, no page number
[287] Cf. *OpenCV*, Numpy, 2021, no page number.
[288] Cf. *Hoisington*, *C*., Video Definiton, 2017, p. 16.
[289] Cf. *ibid*., p. 16.
[290] Cf. *Kadouf*, *H*. *H*., *Mustafah*, *Y*. *M*., Traditional Approach, 2013, p. 7.
[291] Cf. *ibid*., p. 7.
[292] Cf. *OpenCV*, GaussianBlur, 2021, no page number.
[293] Cf. *ibid*., no page number.
[294] Cf. *ibid*., no page number.
[295] Cf. *OpenCV*, Object Detection, 2021, no page number.
[296] Cf. *ibid*., no page number.
[297] Cf. *Bora*, *D*. *et al*., HSV Explanation, 2015, p. 193.

intensity and is better suitable for object detection.[298] The next step is to search the frame for the colors that the ball contains.[299] The ball intended for the prototype is visible in the figure below.

Figure 21: Soccer ball

Source: Own Representation

The soccer ball consists of two colors: orange and blue. A computer cannot recognize the colors directly. Instead, it can only evaluate numbers.[300] That is why lower and upper HSV color values must be defined in a Numpy Array for these two colors.[301] Thus the pixels in the image can be matched with the color values, and the

[298] Cf. *ibid*., p.193.
[299] Cf. *Kadouf*, *H*. *H*., *Mustafah*, *Y*. *M*., Traditional Approach, 2013, p. 2–5.
[300] Cf. *Rheinboldt*, *W*., *Rosenfeld*, *A*., Picture Array, 2014, p. 1–3.
[301] Cf. *NumPy*, Array, 2021, no page number.

ball can be extracted.[302] The following source code in Listing 9 is initialized directly when the application is opened and defines the color values.

Listing 9

```
#intialize the default color range of the red-blue ball
(1)   redLower = np.array([0, 145, 110])
(2)   redUpper = np.array([13, 255, 255])
(3)   blueLower = np.array([110, 30, 75])
(4)   blueUpper = np.array([179, 125, 140])
```

Source: Own presentation

Line one defines the lower and line two the upper bound for the orange and red hue, respectively.[303] The same happens in the two following lines for the blue hue.[304] Thereby values were determined using the range-detector script from the imutils library.[305] After the upper and lower bounds of both colors are defined, a mask is created.[306] A mask creates a new frame in which only the color pixels of the previously defined color range are present.[307] Thus, all pixels that are not orange and blue are blackened.[308] The following Listing 10 shows the corresponding source code snippet.

Listing 10

```
#get the mask from the colors range
(1)     mask1 = cv2.inRange(hsv, redLower, redUpper)
(2)     mask2 = cv2.inRange(hsv, blueLower, blueUpper)
(3)     mask = mask1+mask2

 #apply the mask to the current frame
(4)     res = cv2.bitwise_and(self.frame,self.frame, mask= mask)   #mask for the object color
```

Source: Own presentation

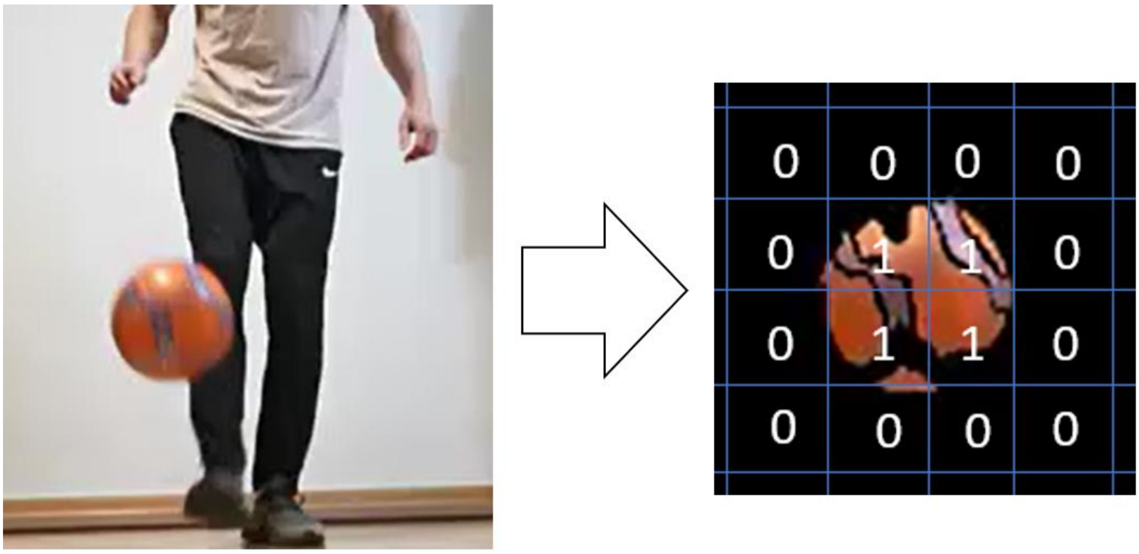
In the first line, a mask is created that sets all the pixels of the current HSV frame to zero that are not within the red HSV color space.[309] Those pixels that are set to zero appear black and are no longer visible.[310]

---

[302] Cf. *NumPy*, Array, 2021, no page number.
[303] Cf. *NumPy*, Boundaries, 2021, no page number.
[304] Cf. *ibid*., no page number.
[305] Cf. *PyImageSearch*, Range-Detector, 2016, no page number.
[306] Cf. *OpenCV*, Thresholding, 2021, no page number.
[307] Cf. *ibid*., no page number.
[308] Cf. *ibid*., no page number.
[309] Cf. *OpenCV*, Array-Operations, 2021, no page number.
[310] Cf. *ibid*., no page number.

The same is repeated in the second line for the Blue color space in the same frame.[311] The following third line merges both masks into one mask.[312] Finally, the last line applies the merged mask to the original frame.[313] The OpenCV function "bitwise_and()" is used for this.[314] It allows linking two sources of the same size with a mask bitwise.[315] The following figure illustrates the "bitwise_and()" function based on the example of a still frame of the prototype.

Figure 22: Bitwise_and() function



Source: Own Representation

The function compares each pixel of the frame with that of the mask.[316] If the color of the pixel is defined in the color range of the mask, the value one, otherwise, the value null is returned.[317] This means that the pixel will be blackened.[318] The blue boxes on the right side of the image are to illustrate the pixels. In reality, of course, the ball consists of more than four pixels.[319] Now the areas of the frame that should be focused on are highlighted in two different RGB colors in a new frame.[320]

[311] Cf. *OpenCV*, Array-Operations, 2021, no page number.
[312] Cf. *ibid*., no page number.
[313] Cf. *ibid*., no page number.
[314] Cf. *ibid*., no page number.
[315] Cf. *ibid*., no page number.
[316] Cf. *ibid*., no page number.
[317] Cf. *ibid*., no page number.
[318] Cf. *ibid*., no page number.
[319] Cf. *Rheinboldt*, *W*., *Rosenfeld*, *A*., Picture Array, 2014, p. 1–3.
[320] Cf. *OpenCV*, Array-Operations, 2021, no page number.

For the sake of reducing processing time, the frame is converted to a binary image afterward according to the following two lines from Listing 11.[321]

Listing 11

```
#convert img to gray
(1)    gray = cv2.cvtColor(res, cv2.COLOR_BGR2GRAY)
#convert to a binary image
(2)    thrimage = cv2.threshold(gray, 20, 255, cv2.THRESH_BINARY)[1]
```

Source: Own presentation

A binary image consists of two colors, black and white. As a result, only the ball will appear in white.[322] Therefore the frame must first be converted to grayscale by the OpenCV function "BGR2Gray()" described in line one.[323] After that, the function "treshhold()" from the second line converts the grayscale image into a binary image.[324] Here, the grayscale image, the threshold value, the maximum value, and the thresholding type are passed as parameters.[325] For each pixel of the grayscale image, the pixel intensity is matched with the threshold value.[326] If the intensity is above twenty, the maximum value is assumed to be 255.[327] If the intensity is below twenty, the value zero is assumed.[328] The ultimate result is shown in Figure 23 below.

[321] Cf. *OpenCV*, Array-Operations, 2021, no page number.
[322] Cf. *ibid*., no page number.
[323] Cf. *OpenCV*, Converting image, 2021, no page number.
[324] Cf. *OpenCV*, Array-Operations, 2021, no page number.
[325] Cf. *OpenCV*, Image Transformation, 2021, no page number.
[326] Cf. *ibid*., no page number.
[327] Cf. *ibid*., no page number.
[328] Cf. *ibid*., no page number.

Figure 23: Binary image

Source: Own Representation

The relevant object, here the soccer ball, is now highlighted in white. In the next step, the "findCountours()" function detects the contours of the soccer ball in a frame by recognizing the outer boundaries of the objects.[329] In Listing 12 below, the implementation of the function is visible.

[329] Cf. *OpenCV*, Contour, 2021, no page number.

Listing 12

```
#find Contours
(1)    cnts = cv2.findContours(thrimage.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPR
        OX_SIMPLE)

(2)    if(len(cnts)!=0):
       cnts =  cnts[0]

(3)    c = max(cnts, key=cv2.contourArea)  #get the largest contour
(4)    (x,y),radius = cv2.minEnclosingCircle(c)   #get the position of the contour
(5)    area = cv2.contourArea(c)              #calculate the shape area
       peri = cv2.arcLength(c, True)
       approx = cv2.approxPolyDP(c, 0.02 * peri, True)
```

Source: Own presentation

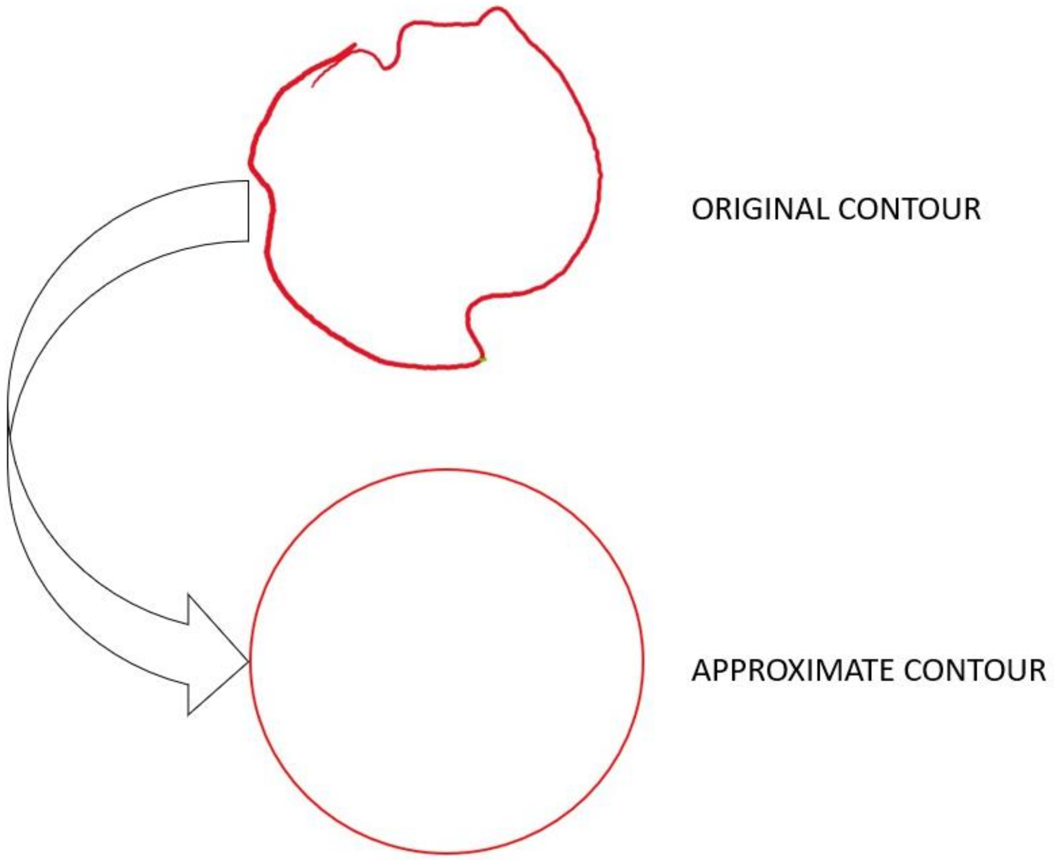
In line one, the function "findContours()" is applied.[330] Here the first parameter, "thrimage.copy()" is the source, a copy of the binary image.[331] The second parameter, "cv2.RETR_EXTERNAL" is the retrieval mode.[332] This returns an array showing how contours are connected.[333] In Figure 24, multiple small white contours are visible on the frame. The mode "cv2.RETR_EXTERNAL" returns only extreme contours.[334]Thus, the small white contours are disregarded, and only the outer pixels of the ball are considered.[335] Then, the third parameter, "CHAIN_APPROX_SIMPLE" saves the contours and reduces the length of the saved contours to a minimum.[336] This way, a circle consisting of numerous contour points can be compressed to a few endpoints.[337] Since a ball has the same shape as a circle, the same is done for the soccer ball.[338] To achieve higher accuracy, the code in line two sets the length of the contours to zero everywhere no shape is found.[339] Then the area of the object is calculated from the given contours in line three and passed to the variable c.[340]

---

[330] Cf. *OpenCV*, Contour, 2021, no page number.
[331] Cf. *ibid*., no page number.
[332] Cf. *ibid*., no page number.
[333] Cf. *ibid*., no page number.
[334] Cf. *ibid*., no page number.
[335] Cf. *ibid*., no page number.
[336] Cf. *OpenCV*, Shape, 2021, no page number.
[337] Cf. *ibid*., no page number.
[338] Cf. *ibid*., no page number.
[339] Cf. *ibid*., no page number.
[340] Cf. *ibid*., no page number.

Finally, the function "minEclosingCircle()" of line four returns the contour area as a radius.[341] The functionality of line five is explained in Figure 24.

Figure 24: Irregular contours



Source: Own Representation

The function "cv2.approxPolyDP()" adjusts irregular contours to create a simplified regular shape.[342] For this purpose, the input array of the contour is passed, and epsilon is defined for the contour.[343] The epsilon defines how much the new contour may deviate from the original contour and is specified in the unit perimeter.[344] The last parameter is True, which means that the contour is a merging contour.[345] This can improve object detection accuracy, especially for objects that resemble regular shapes like circles or rectangles.[346]

[341] Cf. *OpenCV*, Shape, 2021, no page number.
[342] Cf. *OpenCV*, Contour Features, 2021, no page number.
[343] Cf. *ibid*., no page number.
[344] Cf. *ibid*., no page number.
[345] Cf. *ibid*., no page number.
[346] Cf. *OpenCV*, Contour, 2021, no page number.

In the following Listing 13, conditions are set for outlining the soccer ball with a circle.[347] In this way, the performance of the prototype can be better observed during the evaluation.

Listing 13

```
#filter the contour to work on the ball contour
(1)    if(radius>20 and area>1000 and len(approx)> 6):
#draw the circle around the ball
(2)    center = (int(x),int(y))
(3)    radius = int(radius)
(4)    cv2.circle(self.frame,center,radius+15,(208,255,44),2)
```

Source: Own presentation

The first line defines the requirements for the ball's contour.[348] Through trial error attempts, it was determined that the radius should be greater than 20 pixels, the area greater than 1000 pixels, and the number of contour points of the ball should be at least six.[349] If the contour of the ball meets the conditions, a circle is drawn around the ball with the function "cv2.circle()".[350] For this purpose, the function needs the frame, the coordinates of the object center, and the object radius as passing parameters.[351] These are defined in lines two and three. As can be seen in line four, numbers are specified after the radius.[352] For example, the first number, "+15," means that the detected radius should be extended by 15 pixels.[353] Furthermore, the following four numbers define the circle color, the circle thickness, the line type, and the number of decimal places of the coordinates.[354]

4.2.3.4 Ball tracking

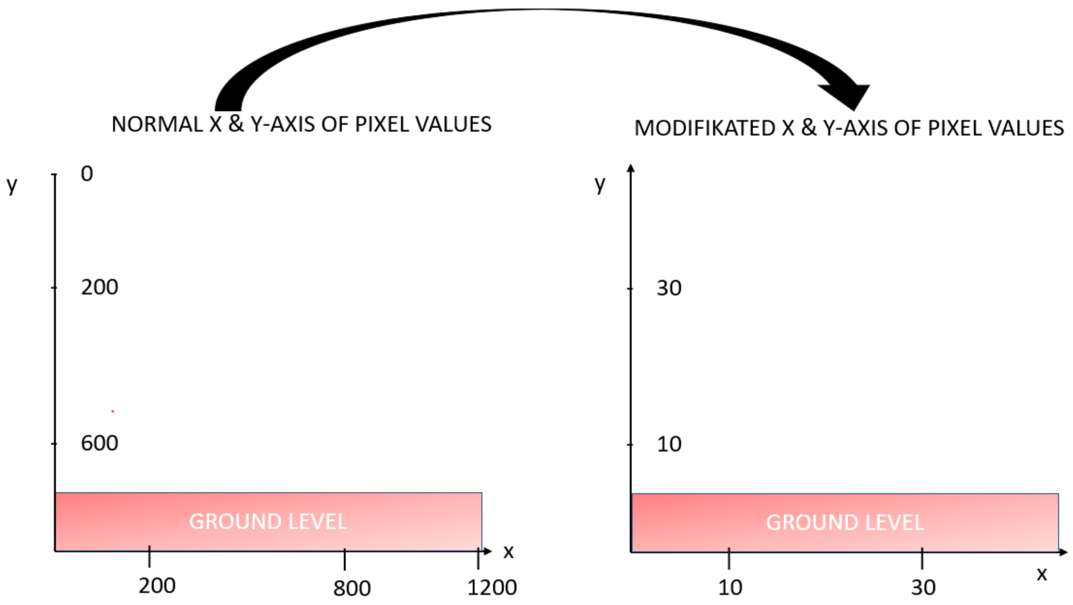
From the source code for ball detection implemented in the previous section, the tracking of the soccer ball during juggling follows in this section.

Previously it must be mentioned that while tracking the ball, movements on the y-axis take place. The y-axis for pixel values runs in the opposite direction to a standard coordinate system.[355] Therefore, the y-axis is shown inverted in the examples below

[347] Cf. *OpenCV*, Contour Features, 2021, no page number.
[348] Cf. *OpenCV*, Contour, 2021, no page number.
[349] Cf. *ibid*., no page number.
[350] Cf. *OpenCV*, Drawing Functions, 2021, no page number.
[351] Cf. *ibid*., no page number.
[352] Cf. *ibid*., no page number.
[353] Cf. *ibid*., no page number.
[354] Cf. *ibid*., no page number.
[355] Cf. *Python Software Foundation*, Tkinter, 2021, no page number.

to better follow the logic of soccer ball tracking. Besides that, fictitious two-digit pixel values are used for the sake of simplicity. As a result, all of the following explanations occur in reverse in reality. This modification is illustrated in the following Figure 25.

Figure 25: Modifikation of y-Axis

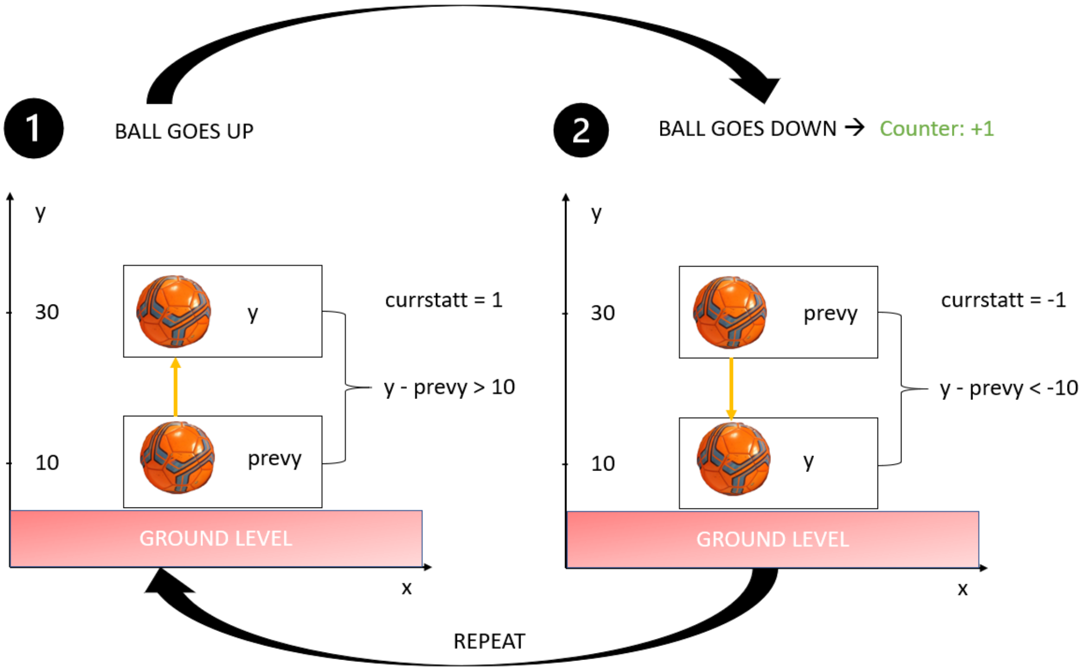


Source: Own presentation

In reality, per juggled contact, the ball flies once from the foot into the air and towards the foot again. Mathematically, the ball initially moves positively upwards along the y-axis until gravity causes it to change direction downwards along the y-axis. This process is repeated until the ball flies to the ground or touches the ground level. Two variables, y, and prevy, are needed to recognize in which direction the ball flies. Y represents the current height, while prevy acts as the previous height of the ball. Thereby only lower and upper bounds are considered. These are the values on the y-axis before and after changes of direction of the ball.

By subtracting y and prevy, the direction of flight of the ball can be determined. If the result is positive, the ball flies upwards. On the contrary, if the result is negative, the ball flies downwards. Initial test trials showed that the prototype often perceived incorrect changes in direction. To reduce the sensitivity and thus the error rate of the prototype, a movement of at least ten pixels in one direction is assumed for a change of direction. A simulation of the following Figure 26 illustrates the ball juggle counting logic.

Figure 26: Counting logic



Source: Own presentation

This fictitious simulation is used to understand the logic of the prototype and is directly related to Figure 26. Suppose person A is juggling a soccer ball and has already juggled the ball eight times. He then starts to juggle the ball for the ninth time at the height of ten pixels. The ball flies up to a height of 30 pixels and then changes its trajectory downwards. If y, in this example 30, is subtracted with prevy, here 10, the result is 20. The result is higher than 10, so the currstatus is set to 1. Each time the currstatus is updated, the ball changes direction, and the variables y and prevy are swapped. Then the ball flies down from 30 pixels until it changes direction again at 10 pixels. Subtracting y, here ten pixels, and prevy, here 30 pixels, equals -20. The result is less than -10, so the currstatus is set to minus one. Thereby, the following conditions from Listing 14 must be met in order to add a juggled contact.

Listing 14

```
(1)    if((currstat * prevstat == -1 and currstat == -1) or n==0) and y > groundLevel:
(2)      n+=1
```

Source: Own presentation

From line one, it can be noticed that the multiplication of currstat and prevstat must equal minus one, while the currstat itself has to be minus one. Additionally, y is not allowed to be in the range of the ground level, which ensures resetting the score as soon as the ball touches the ground. Additionally, the yellow-marked sign differs from the original source code because the y-axis is reversed, as explained earlier.

If the ball moves upwards, the currstat is 1. This means the condition is not met, and no ball contact is added. Contrarily, if the ball flies downwards, the multiplication of currstatt and prevstat equals -1, and the currstat itself has the value -1. In addition, Figure 26 shows that the ball is above ground level. That means all conditions are fulfilled when the ball flies down, and the prototype counts the ninth ball contact. The same procedure is repeated until the ball touches the ground level. Finally, line one of the Listing 14 above shows the condition "or n==0". In this case, n stands for the number of juggled contacts and excludes the previously described conditions for the first ball contact upwards. This particular feature is necessary because there is no prevstat for the first ball contact upwards.

5 Prototype evaluation

5.1 Evaluation procedure

Section 3.3 defined the requirements for the prototype. From this, a mockup was created in section 3.4, which was implemented in chapter 4. The task in this chapter is to find out whether and to what extent the requirements for the prototype are met. The evaluation takes place in three steps. First, all requirements that do not require external evaluations are evaluated through a prototype simulation. In the second step, a homogeneous target group of five people is selected. Finally, each person tries to perform the ball juggling exercise in both modes.

Meanwhile, the prototype and additionally two other people count the number of juggled contacts with human eyes. The prototype has one task to perform. It is supposed to evaluate the technical skill of ball juggling in soccer. For the evaluation of this task performance, the manually counted results are compared with those of the prototype. If the results differ, the task is considered not completed. However, otherwise, the task is completed. As a third step, each user is asked to fill out a questionnaire after juggling. This questionnaire is based on the requirements from section 3.3, for which external evaluation is relevant.

Often the centrality effect occurs with standardized questionnaires.[356] This means that the user chooses the middle of the scale for questions that he or she cannot assess.[357] The user only has yes and no answer options and free lines for comments as an answer option to avoid this.[358] Even though comments can only be evaluated statistically to a limited extent, they give users a feeling of appreciation and occasionally lead to valuable aspects for a particular topic.[359]

5.2 The accomplishment of the evaluation

This subsection evaluates the prototype. This is done by a prototype simulation, a test group, and a questionnaire.

---

[356] Cf. *Hammann*, *P*., *Erichson*, *B*., Cebtrality Effect, 2021, p. 342.
[357] Cf. *ibid*., p. 342.
[358] Cf. *ibid*., p. 342.
[359] Cf. *Baur*, *N*., *Blasius*, *J*., Questions, 2014, p. 675 et seqq.

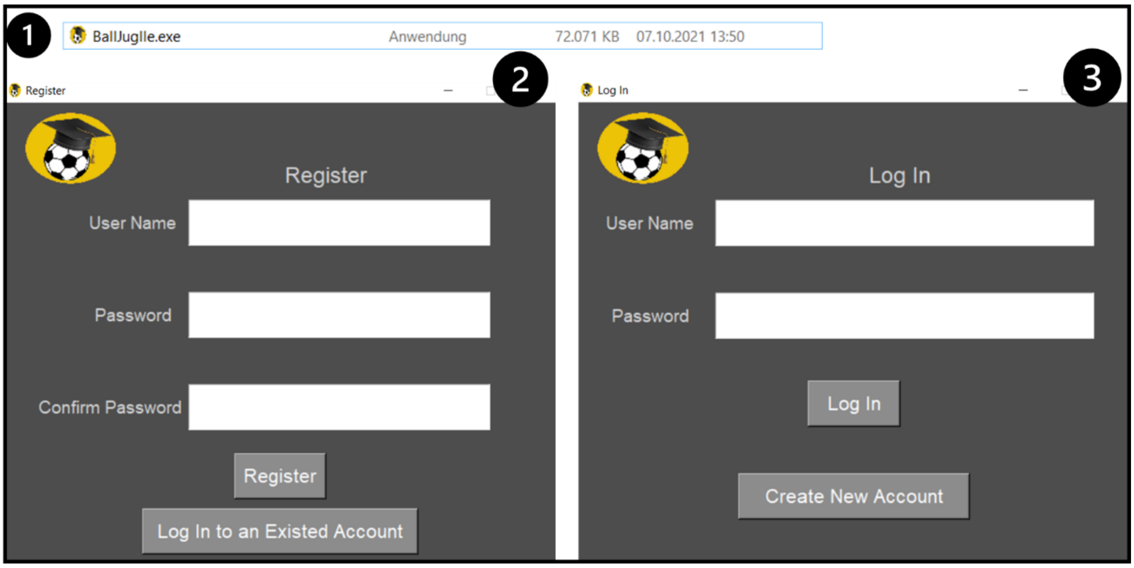
5.2.1 Prototype simulation

During prototype simulation, the prototype is started and explicitly checked for the required criteria from section 3.3. Initially, the following questions are answered:

1) /G4.0/ Is the prototype provided to test users as an EXE file?
2) /F1.1/ Are new users able to set their username and password right away through a registration function?
3) /F1.2/ Is the prototype access protected by a login function for registered users?

The following Figure 27 is used to answer the questions. First, the figure demonstrates the implementation of the requirements on the prototype. Meanwhile, the numbering of the questions corresponds to the graphic.

Figure 27: Evaluation of /G4.0/, /F1.1/, /F1.2/



Source: Own presentation

According to the figure, the prototype is opened via the EXE file next to number one. Afterward, the user has the option to register. As number two shows, a username and password must be set for this. If the user is already registered, he or she can log in directly via the login window from number three.

This brief simulation results in evaluating the requirements /G4.0/, /F1.1/, and /F1.2/ in Table 2.

Table 2: Evaluation of /G4.0/, /F1.1/ ,/F1.2/

NR.	REQUIREMENT	FULFILLED
1	/G4.0/ The prototype must be provided to test users as an EXE file.	✓
2	/F1.1/ New users should be able to set their username and password right away through a registration function.	✓
3	/F1.2/ Access is to be protected by a login function for registered users.	✓

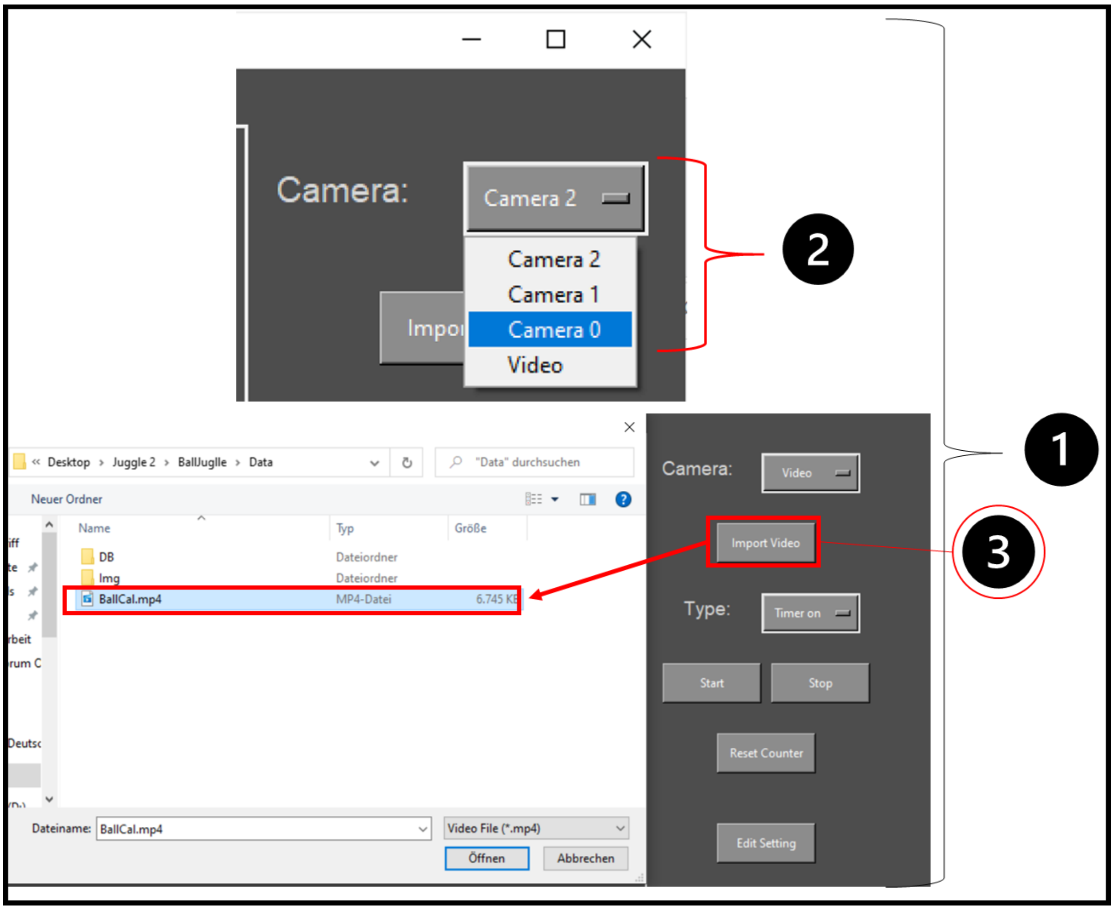
Source: Own presentation

The following further questions can be answered by the prototype simulation to evaluate the requirements /NF2.0/, /F2.0/ and /F2.1/:

1) /NF2.0/ Is the default language of the prototype English?
2) /F2.0/ Does the prototype provide the user with a way to view and select all cameras connected to the device to perform the exercise?
3) /F2.1/ Does the prototype allow users to upload previously recorded videos?

The following Figure 28 shows whether the prototype meets the requirements just mentioned.

Figure 28: Evaluation of /NF2.0/, /F2.0/, /F2.1/



Source: Own presentation

The number one refers to all the elements in Figure 28. The labels on the interface confirm that the default language of the prototype is English. Furthermore, number two shows that all cameras connected to the computer starting with "Camera 0" are displayed and can be selected. In addition, the format "Video" is available for selection. In this case, the "Import Video" button may be used to upload a local video file, as shown in number four. Assuming the user uploads a video file in which he juggles the standard ball, this is evaluated as well. Therefore, the attachment includes the video file "Juggle Tetvideo.mp4", which can be used for testing purposes of the just described requirement /F2.1/. The exact instructions for this are described in Appendix 9.

From the prototype simulation just described the requirements /NF2.0/, /F2.0/ and /F2.1/ are evaluated in Table 3.

Table 3: Evaluation of /NF2.0/, /F2.0/, /F2.1/

NR.	REQUIREMENT	FULFILLED
1	/NF2.0/ The default language of the prototype must be English.	✓
2	/F2.0/ The prototype must provide the user with a way to view and select all cameras connected to the device to perform the exercise.	✓
3	/F2.1/ The prototype must allow users to upload previously recorded videos.	✓

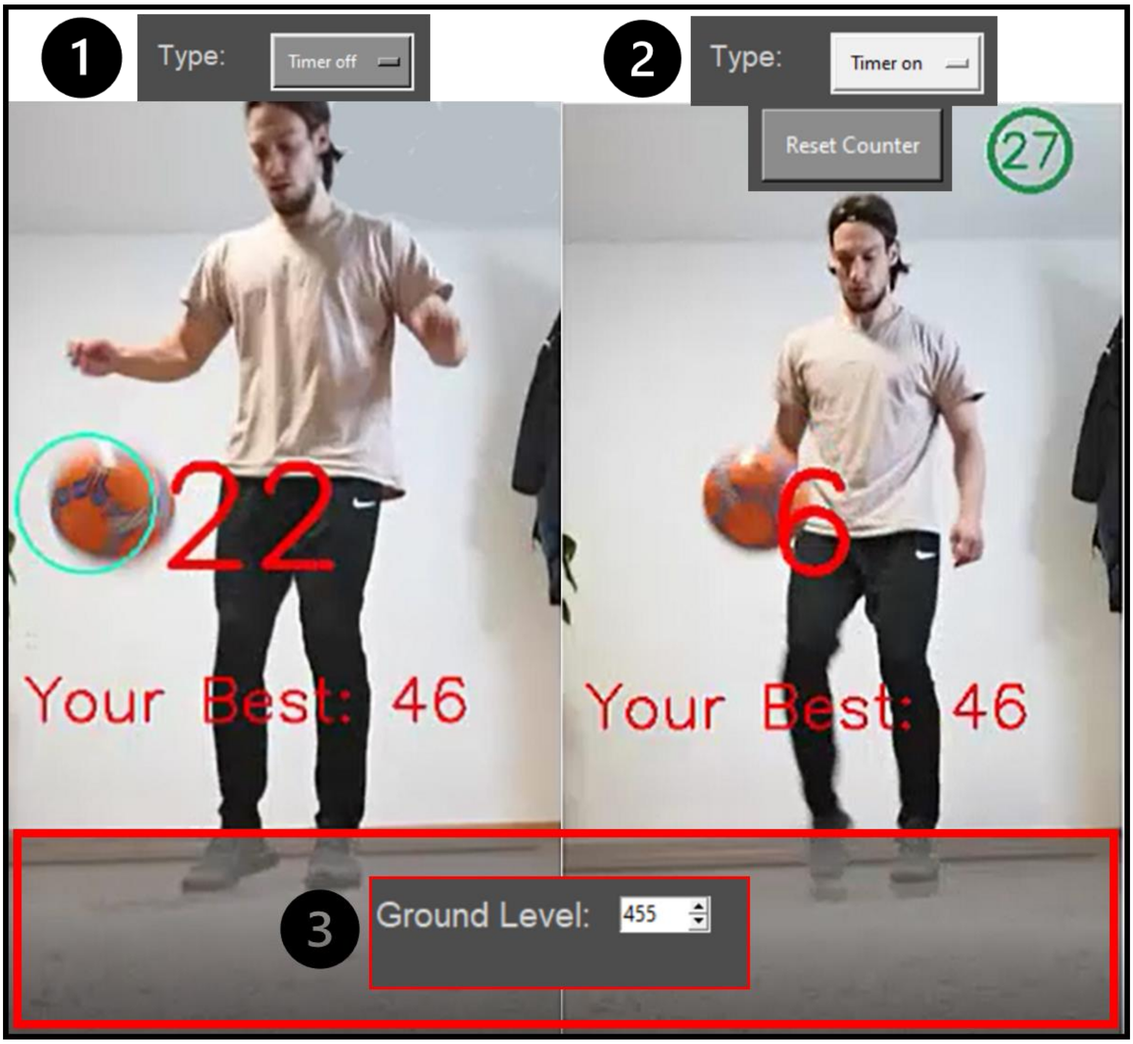
Source: Own presentation

Similarly, the requirements /F3.1/, /F3.2/, /F4.1/ and /F4.3/ raise questions that need to be answered:

1) /F3.1/ Is the first mode must be a free mode. where the prototype measures the number of juggled ball contacts without any time limit?
2) /F3.2/ Does the second mode count the number of contacts during ball juggling within a time window of thirty seconds. And is the timer resettable via a button?
3) /F4.1 & F4.3/ Is the ground adjustable manually and highlighted by a transparent bar while juggling the ball?

In the following figure, image snippets are shown that respond to the questions.

Figure 29: Evaluation of /F3.1/, /F3.2/, /F4.1/, /F4.3/



Source: Own presentation

On the left in the figure beneath number one, the free mode is visible. In this mode, the user is able to juggle without a time limit. Right next to it beneath number two, the second mode can be seen. Here the user has 30 seconds to juggle a maximum possible number of times. Additionally, the user has the option to reset the time via the "Reset Counter" button. The corresponding mode can be selected via the drop-down menu "Type". Afterward, number three shows the ground level, which is outlined thickly in red and highlighted transparently in gray. The "Ground Level" input field can be used to increase or decrease the ground level manually.

Table 4 evaluates the just demonstrated requirements /F3.1/, /F3.2/, /F4.1/ and /F4.3/.

Table 4: Evaluation of /F3.1/, /F3.2/, /F4.1/, /F4.3/

NR.	REQUIREMENT	FULFILLED
1	/F3.1/ The first mode must be a free mode. In this mode, the application must measure the number of juggled ball contacts without any time limit.	✓
2	/F3.2/ In the second mode, the number of contacts must be counted during ball juggling within a time window of thirty seconds. The timer must be resettable via a button.	✓
3	/F4.1 & F4.3/ To enable a user to test the prototype on different devices, in different environments, and with different camera perspectives, the ground level must be set manually. The ground level must be highlighted by a transparent bar while juggling the ball.	✓

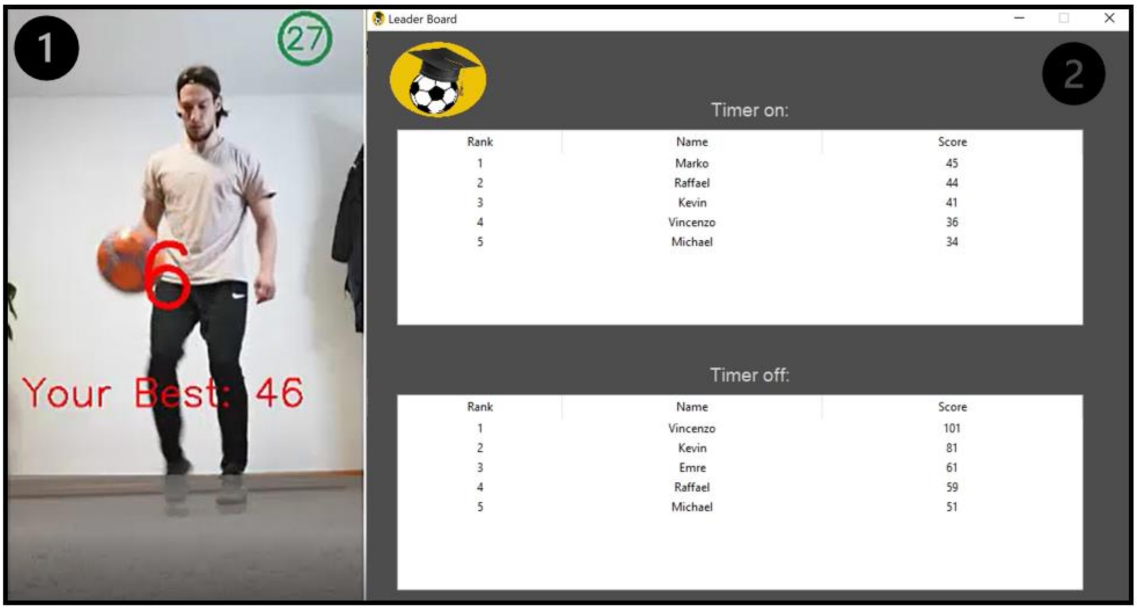
Source: Own presentation

The following questions arise from the final requirements of the prototype simulation:

1) /G1.0 & L1.0/ Is the prototype limited to ball juggling in this bachelor thesis?
2) /G3.0 & L3.0/ Does the prototype must ensure comparability through a ranking list, and is the prototype ranking limited to five places for the sake of clarity?
3) /G2.0/ Is the design ensured by a mockup?

The evaluations of the first two requirements can be seen in Figure 30, while /G2.0/ is evaluated in the text below.

Figure 30: Evaluation of /G1.0 & L1.0/, /G3.0 & L3.0/, /G2.0/



Source: Own presentation

The fact that the prototype is designed only for ball juggling becomes clear when using it. Number one shows the current score during exercise execution. When attempting another soccer exercise, e.g., dribbling with the ball on the ground, an incorrect or no score would be displayed due to the ground level. Number two displays two rankings with the five best attempts of all users. This way, the users have their score for both modes transparently available and can compare themselves with other users. The mockup from requirement number three has been discussed in more detail in section 3.4. Comparing the mockup with the current prototype shows that it served as the foundation for the design. Thus, the requirements /G1.0 & L1.0/, /G3.0 & L3.0/ and /G2.0/ can be evaluated as follows.

Table 5: Evaluation of /G1.0 & L1.0/, /G3.0 & L3.0/, /G2.0/

NR.	REQUIREMENT	FULFILLED
1	/G1.0 & L1.0/ The prototype is limited to ball juggling in this bachelor thesis.	✓
2	/G3.0 & L3.0/ The prototype must ensure comparability through a ranking list. The prototype ranking must be limited to five places for the sake of clarity.	✓
3	/G2.0/ The design must be ensured by a mockup.	✓

Source: Own presentation

In summary, the evaluations performed through the prototype simulation all meet the expectations of the requirements.

5.2.2 Test group

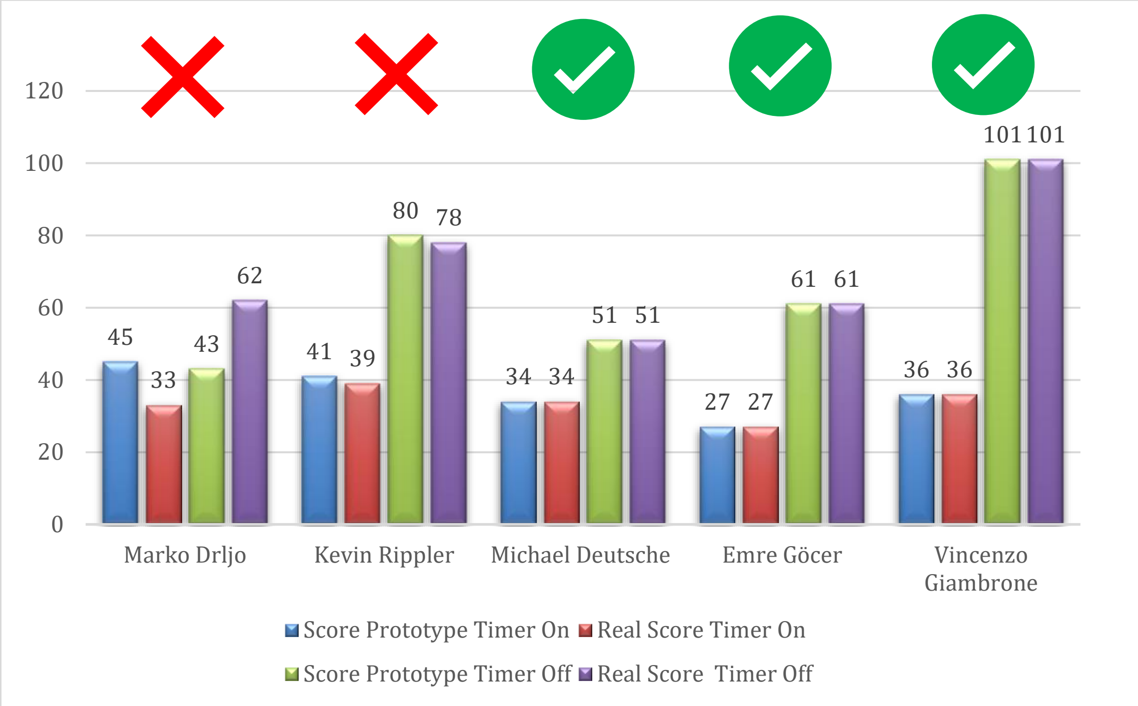
The prototype has the objective to evaluate ball juggling in soccer. "One important common element between soccer juggling and ball reception is the ability to keep the ball under control."[360] For this reason, a homogeneous target group of five people was selected, who are expected to have the skills and affinity for ball juggling. The five persons are active men's soccer players in the seventh-highest German league and play in the same club.

5.2.2.1 Task performance

In this subsection, the prototype is tested for its task performance. For this purpose, a test group of five people performs the ball juggling exercise in sequence. The behavior of the prototype during the execution of the exercise is analyzed. That is necessary to formulate hypotheses for error analyses. The test environment is a soccer field. Each user first starts with the timed constrained mode and then executes the free mode. As mentioned earlier, the number of contacts is manually counted by two people. Since manual counting does not exclude errors, results are only evaluated if two persons have counted the same score. The following diagram compares the manually counted scores and the scores measured by the prototype. The manually counted scores are evaluated as "real scores".

[360] Cf. *Raastad*, *O. et al*., Juggling, 2016, p. 339.

Figure 31: Task performance



Source: Own presentation

The diagram shows that the prototype performs its task 60% correctly. In three out of five, the results of the prototype match the real results. The results of Marko Drljo, with 19 ball contacts in free mode and 12 ball contacts in time mode, deviate significantly from the real results. Observation of the prototype during the exercise execution revealed that the prototype recognized two objects as a ball. Due to the orange training suit of Marko Drljo, the ball could not be permanently recognized as a target object. The enormous deviation of the results is most likely due to this fact. Also, in the case of Kevin Rippler, the results of the prototype deviate from the real results. In both modes, an overcount of two contacts was made by the prototype. Observations during the exercise execution revealed that those two contacts were added even before the ball was in the air for the first time. This indicates that the ground level was set too low. Once the ground level is set too low and the camera is not at 90 degrees to the ground, a movement of the ball on the x-axis can be interpreted as a movement on the y-axis. In case a user plays the ball up from the ground with his foot, there are always ball movements on the y-axis. This is why the prototype's settings were optimized through trial-error attempts after the second participant Kevin Rippler. These optimizations led to the desired results. Thus, the prototype measurements of Michael Deutsche, Emre Göcer, and Vincenzo Giambrone agreed with the real results for both modes. In the following subsection, the questionnaires of the test users are evaluated.

5.2.2.2 Evaluation of the feedback questionnaires

Following each test attempt, the test users received a questionnaire to evaluate the remaining open requirements. The questionnaire was provided to users in German to avoid misinterpretation due to language barriers.[361] The following Figure 32 shows the English version of the questionnaire.[362] In front of each question is the corresponding requirement. All completed questionnaires of the users are attached to the appendix.

[361] Cf. *Appendix*, Chapter 1, Questionnaire, p. 1.
[362] Cf. *ibid*., p. 1.

Figure 32: Feedback questionnaire

Prototype questionnaire

Surname: ___________ Age: _________

Name: ___________ Relation to soccer: Club Player ☐ ___________ ☐

1. /V2.0/ Is the prototype intuitive enough to test out ball juggling on the first try?
   Yes ☐ No ☐
   Note:____

2. /V3.0/ Does the prototype create comparability for ball juggling?
   Yes ☐ No ☐
   Note:____

3. /NF1.0/ Was the current score visible during exercise execution?
   Yes ☐ No ☐
   Note:____

4. /F1.0/ Is a registration and login required to use the prototype?
   Yes ☐ No ☐
   Note:____

5. /F3.0/ Was it possible to choose between the two modes for ball juggling?
   Yes ☐ No ☐
   Note:____

6. /F4.0/ Was the ground level and its relevance to the function clear right away?
   Yes ☐ No ☐
   Note:____

7. /F4.2/ Was the score reset as soon as the ball touched the ground?
   Yes ☐ No ☐
   Note:____

8. /L2.0/ Could the prototype only be used with a specific soccer ball?
   Yes ☐ No ☐
   Note:____

Source: Own presentation

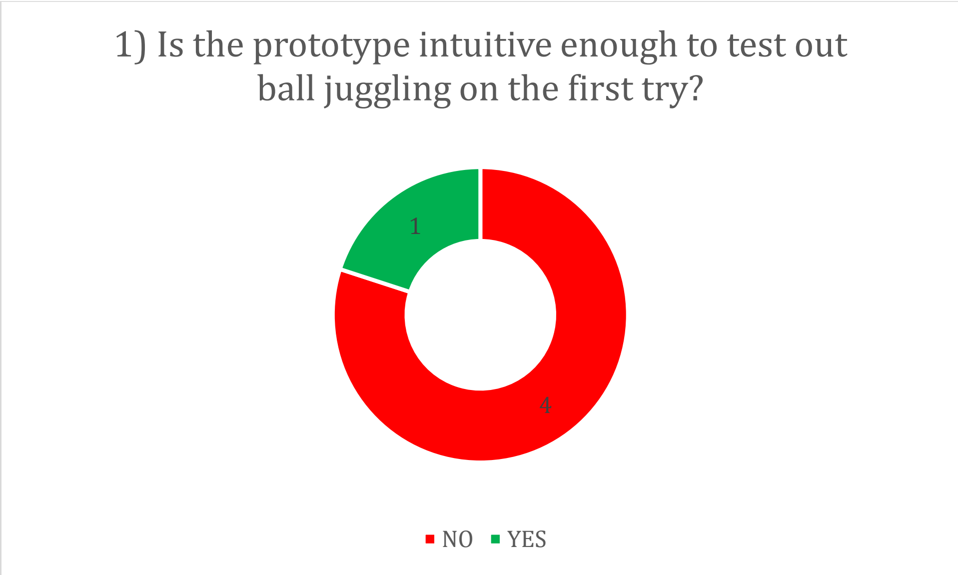
The responses of the test users are discussed as follows.

Question 1: /V2.0/ Is the prototype intuitive enough to test out ball juggling on the first try?

Four out of five test users answered with "No" to the first question, which comes from requirement /V2.0/.[363] Figure 33 illustrates the outcome.

[363] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.

Figure 33: Evaluation of /V2.0/



Source: Own Representation

The answers to the first question yielded the following result. Only Emre Göcer feels that the prototype is intuitive.[364] The added comments reveal the reasons for the four opposing votes. All these four perceive the setting options as non-intuitive. Vincenzo Giambrone explicitly criticizes the selection options for the camera and the setting of the ground level.[365] For Michael Deutsche, the prototype generally requires too many settings.[366] According to him, an intuitive application should only need to be started and stopped after the mode is selected.[367] Marko Drljo agrees and also remarks that too many settings are required.[368] Finally, for Kevin Rippler, the default settings of the prototype are poorly chosen.[369] All in all, 80% of the participants vote for no in the first question.[370] Therefore, the requirement /V2.0/ can be evaluated as "not fulfilled".

Question 2: /V3.0/ Does the prototype create comparability for ball juggling?

The second question is intended to evaluate the requirement /V3.0/ by getting feedback from the users on the comparability. The evaluation of the questionnaires

---

[364] Cf. *Appendix*, Chapter 5, Questionnaire 4 - Emre Göcer, p. 5.
[365] Cf. *Appendix*, Chapter 6, Questionnaire 5 -Vincenzo, Giambrone, p. 6.
[366] Cf. *Appendix*, Chapter 4, Questionnaire 3 - Michael Deutsche, p. 4.
[367] Cf. *Ibid*., p. 4.
[368] Cf. *Appendix*, Chapter 2, Questionnaire 1 - Marko Drljo, p. 2.
[369] Cf. *Appendix*, Chapter 3, Questionnaire 2 - Kevin Rippler, p. 3.
[370] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.

shows that this requirement strongly depends on the task fulfillment from the previous section. Both Kevin Rippler[371]and Marko Drljo assessed that the prototype does not establish comparability for ball juggling.[372] For both users, the prototype calculated a wrong result.[373] According to the comments, these are the reasons for their responses.[374] However, for the remaining three users, the prototype creates complete comparability for ball juggling.[375] Thus, the second question can be answered as follows. With 60% agreement, the requirement /V3.0/ is only partially fulfilled.

Question 3-8: All participants agree entirely with the remaining requirements from questions three to eight. Therefore, the requirements are evaluated directly in the following Table 6.

Table 6: Evaluation of /NF1.0/, /F1.0/, /F3.0/, /F4.0/, /F4.2/, /L2.0/

NR.	REQUIREMENT	FULFILLED
3	/NF1.0/ Was the current score visible during exercise execution?	✓
4	/F1.0/ Is a registration and login required to use the prototype?	✓
5	/F3.0/ Was it possible to choose between the two modes for ball juggling?	✓
6	/F4.0/ Was the ground level and its relevance to the function clear right away?	✗
7	/F4.2/ Was the score reset as soon as the ball touched the ground?	✓
8	/L2.0/ Could the prototype only be used with a specific soccer ball?	✓

Source: Own presentation

As the table shows, all participants answered "yes" to questions 3, 4, 5, 7, and 8.[376] In this respect, the corresponding requirements are rated as fulfilled. A significant shortcoming for the users is the handling of the ground level.[377] None of the

[371] Cf. *ibid.* p. 3.
[372] Cf. *Appendix*, Chapter 2, Questionnaire 1 - Marko Drljo, p. 2.
[373] Cf. *ibid*. p.2.; Cf. *Appendix*, Chapter 3, Questionnaire 2 - Kevin Rippler, p. 3.
[374] Cf. *ibid*. p.2.; Cf. *ibid*. p.3.
[375] Cf. *Appendix*, Chapter 4 – 6, Questionnaire 3 – 5, p. 4 – 6.
[376] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
[377] Cf. *ibid*. p.2.; Cf. *ibid*. p. 2 – 6.

participants could immediately understand how the ground level works, as confirmed by the answers to question 6.[378] Four users added comments that are relevant for improvement. For Emre Göcer, an infobox would have been helpful.[379] Marko Drljo thinks the ground level setting is too cumbersome.[380] According to Kevin Rippler, users only want to use the function, and such settings are not user-friendly.[381] Moreover, Michael Deutsche notes that the principle of the ground level is understandable after a short explanation.[382] Hence, the requirement /F4.0/ can be evaluated as "not fulfilled".

5.3 Consolidation of all results

The results from the prototype simulation, task performance results, and feedback questionnaires are combined in Table 7, Table 8, Table 9, and Table 10 below. Thereby, optional requirements are not taken into account.

[378] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
[379] Cf. *Appendix*, Chapter 5, Questionnaire 4 - Emre Göcer, p. 5.
[380] Cf. *Appendix*, Chapter 2, Questionnaire 1 - Marko Drljo, p. 2.
[381] Cf. *Appendix*, Chapter 3, Questionnaire 2 - Kevin Rippler, p. 3.
[382] Cf. *Appendix*, Chapter 4, Questionnaire 3 - Michael Deutsche, p. 4.

Table 7: Results functional requirements

FUNCTIONAL REQUIREMENTS	FULFILLED
/F1.0/ The prototype must have a protected area and a public area.	✓
/F1.1/ New users should be able to set their username and password right away through a registration function.	✓
/F1.2/ Access is to be protected by a login function for registered users.	✓
/F2.0/ The prototype must provide the user with a way to view and select all cameras connected to the device to perform the exercise.	✓
/F2.1/ The prototype must allow users to upload previously recorded videos.	✓
/F3.0/ The prototype must provide two modes for ball juggling.	✓
/F3.1/ The first mode must be a free mode. In this mode, the prototype must measure the number of juggled ball contacts without any time limit.	✓
/F3.2/ In the second mode, the number of contacts must be counted during ball juggling within a time window of thirty seconds. The timer must be resettable via a button.	✓
/F4.0/ Every user must understand that the ground level is the boundary between the instep of the foot and the ground. It represents the ground and is essential for counting the ball contacts during juggling.	✗
/F4.1/ To enable a user to test the prototype on different devices, in different environments, and with different camera perspectives, the ground level must be set manually.	✓
/F4.2/ As soon as the soccer ball reaches the ground level, it means that the ball hit the ground, and the score must be transferred to the database and set to zero.	✓
/F4.3/ The ground level must be highlighted by a transparent bar while juggling the ball.	✓

Source: Own presentation

Table 8: Results nonfunctional requirements & project limits

NONFUNCTIONAL REQUIREMENTS	FULFILLED
/NF1.0/ The prototype must display the results in near real time.	✓
/NF2.0/ The default language of the prototype must be English.	✓
PROJECT LIMITS	FULFILLED
/L1.0/ The prototype is limited to ball juggling in this bachelor thesis.	✓
/L2.0/ The prototype must be designed for a specific soccer ball by default.	✓
/L3.0/ The prototype ranking must be limited to five places for the sake of clarity.	✓

Source: Own presentation

Table 9: Results project goals

PROJECT GOAL	FULFILLED
/G1.0/ The prototype must evaluate ball juggling.	✓
/G2.0/ The design must be ensured by a mockup.	✓
/G3.0/ The prototype must ensure comparability through a ranking list.	✓
/G4.0/ The prototype must be provided to test users as an EXE file.	✓

Source: Own presentation

Table 10: Results project visions

PROJECT VISION	FULFILLED
/V1.0/ The prototype must serve as an aid for the automated measurement of technical skills in soccer. This will be demonstrated using the example of juggling.	?
/V2.0/ The prototype must be intuitive enough that any user can operate it on the first try.	✗
/V3.0/ The prototype must create an objective basis, especially comparability for users.	✓ ✗

Source: Own presentation

Out of a total of 24 requirements, the prototype fulfills 20. The prototype has deficits in terms of intuitive operation, which includes the ground level.[383] Therefore, the requirements /F4.0/ and /V2.0/ are not met. In addition, the prototype evaluates only six out of ten juggling attempts with the correct result. This means that requirement /V3.0/ is only partially fulfilled. Finally, requirement /V1.0/ is the only one that remains unanswered. This can be answered as follows from the overall picture that emerges. The prototype could be operated by all five test users and evaluated a total of six out of ten juggling attempts correctly. Thus, a positive proof of concept is achieved for six juggling trials. Since one valid trial alone would have been sufficient and the requirement does not define reliability as a prerequisite, the prototype can automatically evaluate ball juggling. The requirement /V1.0/ is therefore fulfilled.

Nevertheless, It is worth mentioning that the two quality criteria, reliability, and validity are currently not fulfilled.[384] Only when measurements of a system are constantly accurate is it possible to speak of valid results.[385] Moreover, only, in this case, a system is reliable.[386]

[383] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
[384] Cf. *Wagemaker*, *H*., Quality Criteria, 2020, p. 11 et seqq.
[385] Cf. *Ibid*., p. 11 et seqq.
[386] Cf. *Ibid*., p. 11 et seqq.

6 Recommendation for further development

Recommendations for the further development of the prototype can be derived from the evaluation results. The recommendations are intended to help improve the prototype and focus on aspects that were not satisfactory during the evaluation.

The evaluation shows that the prototype is not intuitive.[387] Specifically, the many setting options, especially the ground level, are criticized.[388] Ultimately, the setting options are essential for the prototype's functionality and cannot be omitted without further ado. This means that new technical solutions are required beforehand. At this point, the optional requirement /OF4.4/ should be highlighted. The idea behind this is the automated adjustment of the ground level. Thus, the user's shoe can be detected and serve as the upper limit of the ground level. A manual adjustment would therefore be superfluous.

Another measure that leads to the reduction of the setting options is to replace the traditional object recognition with a DL approach.[389] Although the ball was recognized and tracked without any problems for four out of five participants, the evaluation of participant Marko Drljo showed that the color choice of the clothes could distort the functionality of the traditional object recognition.[390] Whether and to what extent the DL approach can counteract this problem is to be found out. In this context, there is another advantage that speaks for the DL approach. While the traditional approach is only tuned to a specific soccer ball, the DL approach allows the evaluation of any soccer ball. How exactly this evaluation is and in which way the evaluation depends on the color and pattern of a soccer ball remains an open question.

The third recommendation may be a solution to increase Task Success. From the evaluation, it is evident that the incorrect results of the prototype were caused by poor default settings, among other things. In addition to the recommendations already mentioned, a box within the screen can also be displayed. This box should indicate to the user where to be during the exercise. Therefore, it can serve as an orientation for the user and thus creates constant distances and possibly higher accuracies in the evaluation.

---

387 Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
388 Cf. *ibid*. p. 2 – 6.
389 Cf. *Manning*, *C*., AI & ML Definition, 2020, no page number.
390 Cf. *Appendix*, Chapter 2, Questionnaire 1 - Marko Drljo, p. 2.

Lastly, the functionality of this prototype can be modified for other technical exercises, such as passing or dribbling exercises. For this, the movements of the ball on the x-axis need to be tracked. However, since movements in the y-axis were evaluated for ball juggling, it can be speculated that the development for further technical exercises can be realized without much additional effort.

7 Conclusion and outlook

This bachelor thesis addressed the question, "Is it possible to measure the technical skill of ball juggling in soccer using computer vision?" To answer this question, a prototype was to be developed, proving or disproving the feasibility of the research question. In order to create a knowledge base, first, the requirement profile of a soccer player and the topics CV, AI, and the term prototype was examined in more detail. Subsequently, two ML approaches to prototypical development were illustrated and compared. This resulted in the decision to choose the traditional ML approach for the development of the prototype. Afterward, requirements, which the prototype should fulfill, were written down – these requirements then had to be integrated into a mockup. The mockup depicted the various interfaces of the prototype and served as the basis for the following source code implementation. Finally, the prototype was evaluated by a prototype simulation, as well as a test group and questionnaire, and the results were compiled.

The results show that the goal of this bachelor thesis was achieved. The prototype succeeds in measuring the technical ability of ball juggling by using CV. Overall, the prototype achieves a success rate for a valid evaluation of 60% in ten attempts. This means that six out of ten trials were evaluated correctly, and the prototype proves the feasibility of automated ball juggling evaluation. Furthermore, this result shows that, insofar as all settings are made correctly, the basic idea of the prototype works, and the components have been implemented successfully. These components include the implementation of ball recognition, ball tracking, and all related implementations, as well as ground level, ranking list, database connection, login, registration, timer, modes, and finally, the frontend design.

Nevertheless, the shortcomings of the prototype have also been identified.[391] These include, in particular, the enormous dependence between the prototype settings and the evaluation success.[392] The settings greatly influence the evaluation and lead to the system's lack of reliability and validity.[393] The reason for this is that the necessary consistency in the results is missing.[394] Thus, the second question defined in the objective can be answered as follows. If the correct evaluation of results cannot be

[391] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
[392] Cf. *Ibid*., p. 2 – 6.
[393] Cf. *Wagemaker*, *H*., Quality Criteria, 2020, p. 11 et seqq.
[394] Cf. *Ibid*., p. 11 et seqq.

guaranteed, the added value of comparability is not ensured. This means that the current version of the prototype cannot be provided as an application for the masses.

Another critical point is the potential for improvement to make the prototype more intuitive.[395] The test group results have shown that especially the need to understand the ground level is too complex.[396] All setting options beyond the selection of modes and the start and stop buttons were also viewed critically.[397]

The extent to which the challenges addressed by the prototype will be solved in the future remains an open question that will not be answered in this bachelor thesis. However, possibilities that can be taken as a starting point have already been described in chapter 6.

For example, manual adjustments could be reduced by automated modification of the ground level. Here, the user's shoe could be detected and taken as the upper limit of the ground level. Also, the use of DL might minimize the adjustment possibilities. In addition, DL can extend the prototype's functionality to all soccer balls, which drastically increases user-friendliness. Last, the extension of the prototype with an added box in the frame that serves as an orientation for the user may create constant distances and thus lead to better evaluation results. Provided the deficiencies mentioned above are addressed, a functional extension to the prototype could add value. Some potential ideas were also explained in Chapter 6. For example, further development to include passing or dribbling exercises makes sense since the technical implementation is similar to the ball juggling approach in this bachelor’s thesis.

In conclusion, the prototype offers a solution to the problem described at the beginning. It automatically measures the technical ability to juggle the ball and thus creates a positive proof of concept. Nevertheless, some deficits prevent the release for the masses. These deficiencies need to be addressed in the further development of the prototype. For this, the suggestions and clues discussed here can be used as an aid.

[395] Cf. *Appendix*, Chapter 2 – 6, Questionnaire 1 – 5, p. 2 – 6.
[396] Cf. *Ibid*., p. 2 – 6.
[397] Cf. *Ibid*., p. 2 – 6.

Appendix

This bachelor's thesis does not contain an appendix. However, the appendix is available online in a separate document with the title "Appendix.docx". Below the appendix list is shown.

Appendix list

Appendix No. Description
Page

Bibliography

Abts, *Dietmar* (Thread, 2018): Grundkurs JAVA: Von den Grundlagen bis zu Datenbank- und Netzanwendungen, 10th Ed., Wiesbaden: Springer, 2018

Aditya, *Mathur P*. (Waterfall, 2013): Foundations of Software Testing: For VTU, London: Pearson Education, 2013

Antz, *Eva M*., *Franz*, *Julia*, *Frieters*, *Norbert*, *Scheunpflug*, *Annette*, *Tolksdorf*, *Markus* (Information, 2009): Generationen lernen gemeinsam: Theorie und Praxis intergenerationeller Bildung, Bielefeld: wbv Media, 2009

Balzert, *Helmut* (Software Lifecycle, 2011): Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, Wiesbaden: Springer, 2011

Balzert, *Helmut*, *Koschke*, *Rainer*, *Lämmel*, *Uwe*, *Liggesmeyer*, *Peter*, *Quante*, *Jochen* (Requirements, 2010): Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, Weinheim: Beltz Verlag, 2010

Baur, *Nina*, *Blasius*, *Jörg* (Question, 2014): Handbuch Methoden der empirischen Sozialforschung, New York: Springer, 2014

Bell, *Douglas* (Prototype, 2005): Software Engineering For Students: A Programming Approach, 2nd Ed., Boston: Addison Wesley, 2005

Bhasin, *Harsh* (Python Language, 2018): Python Basics: A Self-Teaching Introduction, Herndon: Mercury Learning & Information, 2018

Bisanz, *Gero*, *Gerisch*, *Gunnar* (Technique, 2007): Fußball: Kondition - Technik - Taktik und Coaching, Aachen: Meyer & Meyer Verlag, 2007

Bora, *Dibya*, *Gupta*, *Anil*, *Khan*, *Fayaz* (HSV Explanation, 2015): Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation, in: IJETAE, 5 (2015), no. 2, p. 193.

Bostrom, *Nick* (ML & AI, 2014): Superintelligence: Paths, Dangers, Strategies, Oxford: Oxford University Press, 2014

Broy, *Manfred*, *Malkis*, *Alexander* (Programming, 2019): Logische und Methodische Grundlagen der Programm- und Systementwicklung: Datenstrukturen, funktionale, sequenzielle und objektorientierte Programmierung - Unter Mitarbeit von Alexander Malkis, Wiesbaden: Springer, 2019

Burger, *Wilhelm*, *Burge*, *Mark* (HSV, 2010): Principles of Digital Image Processing: Fundamental Techniques, 2nd Ed., New York: Springer Publishing, 2010

Byernetzki, *Carsten* (Amateur-Kongress, 2019): Amateurfußball-Kongress 2019, in: hfv-info, 49, 2019, no. 9, p. 1–8.

Camburn, *Bradley*, *Viswanathan*, *Vimal*, *Linsey*, *Julie*, *Anderson*, *David*, *Jensen*, *Daniel*, *Crawford*, *Richard*, *Otto*, *Kevin*, *Wood*, *Kristin* (Design, 2017): Design prototyping methods: state of the art in strategies, techniques, and guidelines, in: Design Science, 3 (2017), no. 13, p. 1–33.

Coleman, *Ben*, *Goodwin*, *Dan* (Prototype, 2017): Designing Ux: Prototyping: Because Modern Design Is Never Static, 2nd Ed., Farnham: O'Reilly, 2017

Drobisch, *Dominik*, *Stratmann*, *Arne* (Soccer Technique, 2021): Kinderfußball: Trainieren wie die TSG Hoffenheim, Münster: Philippka, 2021

Frings, *Stephan*, *Müller*, *Frank* (Eye, 2019): Biologie der Sinne: Vom Molekül zur Wahrnehmung, 2nd Ed., Wiesbaden: Springer, 2019

Gollapudi, *Sunila* (NLP & CV, 2019): Learn Computer Vision Using OpenCV: With Deep Learning CNNs and RNNs, New York: Apress, 2019

Hammann, *Peter*, *Erichson*, *Bernd* (Centrality Effect, 2021): Marktforschung: Grundwissen der Ökonomik. Betriebswirtschaftslehre, 5. Ed., Stuttgart: UTB, 2021

Hoisington, *Corinne* (Video Definition, 2017): Technology Now: Your Companion to SAM Computer Concepts, 2nd Ed., Boston: Cengage Learning, 2017

Harari, *Yuval N*. (Light, 2015): Sapiens: A Brief History of Humankind, New York: Vintage, 2015

Kadouf, *Hani H*., *Mustafah Yasir M*. (Traditional Approach, 2013): Colour-based Object Detection and Tracking for Autonomous Quadrotor UAV, in: IOP Conference Series: Materials Science and Engineering, 53,(2013), p. 1-9

Karasik, *David*, *Demissie*, *Serkalem*, *Cupples*, *Adrienne L*., *Kiel*, Douglas P. (Biological Age, 2005): Disentangling the Genetic Determinants of Human Aging: Biological Age as an Alternative to the Use of Survival Measures, in:

The Journals of Gerontology Series A: Biological Sciences and Medical Sciences, 60 (2005), no. 5, p. 574–587.

Khanna, *Rahul*, *Awad*, *Mariette* (Machine Learning, 2015): Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, New York: Apress, 2015

Kirste, *Moritz*, *Schürholz*, *Markus* (AI Meaning, 2018): Einleitung: Entwicklungswege zur KI, Wittpahl, in: Volker (Eds.), Künstliche Intelligenz: Technologien | Anwendung | Gesellschaft, Wiesbaden: Springer, 2018, p. 21-35.

Kneuper, *Ralf* (Software Processes, 2018): Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes (English Edition), Basel: Springer, 2018

Lottermann, *Steffan*, *Laudenklos*, *Peter*, *Friedrich*, *Arne*, *Metaxas*, *Iraklis*, *Tritschoks*, *Jürgen*, *Ferrauti*, *Alexander*, *Weber*, *Klaus* (Leistungsdiagnostik, 2003): Technikdiagnostik und Techniktraining im Jugendfußball, in: Neumann, Gabriele (Eds.), Fußball vor der WM 2006: Spannungsbogen zwischen Wissenschaft und Organisation, Köln: Sport und Buch Strauß, 2003, p. 91-105.

Martin, *Robert* (Clean Code, 2008): Clean Code: A Handbook of Agile Software Craftsmanship, Upper Saddle River: Prentice Hall, 2008

McDowall, *Mike* (Director). (Ronaldo, 2011): Ronaldo: Tested to the Limit [Documentary], London: Plum Pictures, 2011

Michelucci, *Umberto* (CNN, 2019): Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection, New York: Apress, 2019

Raastad, *Olav*, *Aune*, *Kristian A*., van den Tillaar, Roland (Juggling, 2016): Effect of Practicing Soccer Juggling With Different Sized Balls Upon Performance, Retention, and Transfer to Ball Reception, in: Motor Control, 20 (2016), no. 4, p. 337–349.

Ramalho, *Luciano* (Init Method, 2015): Fluent Python, Farnham: O’Reilly, 2015

Rani, *Sujata*, *Kumar*, *Parteek* (Machine-Learning, 2018): Deep Learning Based Sentiment Analysis Using Convolution Neural Network, in: AJSE, 44 (2018), no. 4, p. 3305–3314.

Reinhold, *Thorsten* (Athletic, 2008): Leistungsdiagnostik im Fußball: Anforderungsprofil, Konditionstests, Diagnostik im Spitzenfußball, Saarbrücken: VDM Verlag, 2008

Ren, *Shaoqing*, *He*, *Kaiming*, *Girshick*, *Ross*, *Sun*, *Jian* (Deep Learning, 2017): Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017), no. 6, p. 1137–1149.

Rheinboldt, *Werner*, *Rosenfeld*, *Azriel* (Picture Array, 2014): Picture Languages: Formal Models for Picture Recognition, Cambridge: Academic Press, 2014

Ruck, *Rolf*, *Schmidbauer*, *Horst*, (Technical Skills, 2013): Erfolgreich Fußball spielen: Technik | Taktik | Training, 2nd Ed., Weinheim: Beltz Verlag, 2013

Russell, *Stuart*, *Norvig*, *Peter*, *Ming-Wei*, *Chang*, *Devlin*, *Jacob*, *Dragan*, *Anca* (AI, 2020): Artificial Intelligence: A Modern Approach (Pearson Series in Artifical Intelligence), 4. Ed., London: Pearson, 2020

Schreiner, *Peter* (Ball Juggling, 2009): Fußball - Perfekte Ballbeherrschung, Aachen: Meyer & Meyer Sport, 2009

Sehnbruch, *Lucia* (Camera, 2017): Eine Mediengeschichte des Bildschirms: Analyse der Dispositive visueller Wahrnehmungskonstruktion, Wiesbaden: Springer, 2017

Silaparasetty, *Nikita* (ML Workflow, 2020): Machine Learning Concepts with Python and the Jupyter Notebook Environment: Using Tensorflow 2.0, New York: Apress, 2020

Stein, *Manuel*, *Janetzko*, *Halldor*, *Lamprecht*, *Andreas*, *Breitkreutz*, *Thorsten*, *Zimmermann*, *Philipp*, *Goldlucke*, *Bastian*, *Schreck*, *Tobias* (Sport Analysis, 2018): Bring it to the Pitch: Combining Video and Movement Data to Enhance Team Sport Analysis, in: IEEE TVCG, 24 (2018), no. 1, p. 13–22.

Wagemaker, *Hans* (Quality Criteria, 2020): Reliability and Validity of International Large-Scale Assessment, New York: Springer, 2020

Wittpahl, *Volker* (Go, 2018): Vorwort, in: Wittpahl Volker (Eds.), Künstliche Intelligenz: Technologien | Anwendung | Gesellschaft, Wiesbaden: Springer, 2018, p. 7-13.

Internet sources

Borussia Dortmund GmbH & Co. KGaA (Geschäftsbericht, 2021): Jahresabschluss der Saison 2020/2021 (2021-08-09) https://aktie.bvb.de/Publikationen/Geschaeftsberichte/Geschaeftsbericht-2020-2021-KGaA-Konzern (Access 2021-08-31, 15:33 MEZ)

Brecht, *Claudio* (Technologies, 2020): TSG 1899 Hoffenheim Promotes Young Talent Through Cutting-Edge Technology (2020-04-28) https://news.sap.com/2020/04/tsg-1899-hoffenheim-young-talent-technology/ (Access 2021-07-26, 16:30 MEZ)

FC Bayern München AG (Jahresabschluss, 2020): Geschäftsbericht 2019/2020 (2020-12-18) https://fcbayern.com/de/news/2020/12/jahresabschluss-der-saison-2019-20---corona-bedingt-einbussen-bei-umsatz-und-gewinn (Access 2021-07-31, 12:35 MEZ)

Figma Inc. (UX, 2021): It's time to design on your terms (2021-07-23) https://www.figma.com/design/ (Access 2021-09-03, 09:11 MEZ)

Grand View Research, (Soccer Segment, 2021): Sports Technology Market Size, Share & Trends Analysis Report By Technology (Devices, Smart Stadium), By Sports (Basketball, Soccer, Tennis, Golf), By Region (APAC, Europe), And Segment Forecasts, 2021 - 2028 (2021-02) https://www.grandviewresearch.com/industry-analysis/sports-technology-market (Access 2021-07-26, 16:42 MEZ)

IBM (CV, n.d.): What is computer vision? (n.d.) https://www.ibm.com/topics/computer-vision (Access 2021-08-04, 12:11 MEZ)

Leviathan, *Yaniv*, *Matias*, *Yossi* (Google Duplex, 2018): Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone (2018-05-08) https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html (Access 2021-08-02, 19:27 MEZ)

Mannig, *Christopher* (AI & ML Definition, 2020): Artificial Intelligence Definitions (2020-09) https://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf (Access 2021-08-03, 10:21 MEZ)

Neander, *Joachim* (Chess, 1996): Computer schlägt Kasparow (1996-02-12) https://www.welt.de/print-welt/article652666/Computer-schlaegt-Kasparow.html (Access 2021-08-02, 19:00 MEZ)

NumPy (Array, 2021): Array creation (2021-06-22) https://numpy.org/doc/stable/user/basics.creation.html (Access 2021-09-22, 11:17 MEZ)

NumPy (Boundaries, 2021): numpy.append (2021-06-22) https://numpy.org/doc/stable/reference/generated/numpy.append.html (Access 2021-09-22, 11:22 MEZ)

OpenCV (About, 2021): About (2020-11-04) https://opencv.org/about/ (Access 2021-09-10, 08:01 MEZ)

OpenCV (Array-Operations, 2021): OpenCV_ Operations on arrays (2021-07-05) https://docs.opencv.org/3.4.15/d2/de8/group__core__array.html (Access 2021-09-25, 12:05 MEZ)

OpenCV (Capture Video, 2021): Getting Started with Videos (2021-07-05) https://docs.opencv.org/master/dd/d43/tutorial_py_video_display.html (Access 2021-09-22, 10:17 MEZ)

OpenCV (Contour Features, 2021): Contour Features (2021-07-05) https://docs.opencv.org/4.5.3/dd/d49/tutorial_py_contour_features.html (Access 2021-10-02, 08:41 MEZ)

OpenCV (Contour, 2021): Contours Hierarchy (2021-07-05) <https://docs.opencv.org/master/d9/d8b/tutorial_py_contours_hierarchy.html> (Access 2021-10-01, 09:18 MEZ)

OpenCV (Converting image, 2021): Graph API: Converting image from one color space to another (2021-07-05) https://docs.opencv.org/4.5.3/dc/d38/group__gapi__colorconvert.html#ga3e8fd8197ab16811caf9b31cb1e08a05 (Access 2021-09-25, 14:15 MEZ)

OpenCV (Drawing Functions, 2021): Drawing Functions (2021-07-05) https://docs.opencv.org/4.5.3/d6/d6e/group__imgproc__draw.html#gaf10604b069374903dbd0f0488cb43670 (Access 2021-10-02, 08:58 MEZ)

OpenCV (GaussianBlur, 2021): GaussianBlur (2021-07-05) https://docs.opencv.org/4.5.3/d4/d86/group__imgproc__filter.html#gaabe8c836e97159a9193fb0b11ac52cf1 (Access 2021-08-05, 10:08 MEZ)

OpenCV (Image Transformation, 2021): Miscellaneous Image Transformations (2021-07-05) https://docs.opencv.org/3.4.13/d7/d1b/group__imgproc__misc.html#gae8a4a146d1ca78c626a53577199e9c57 (Access 2021-09-25, 14:24 MEZ)

OpenCV (Numpy, 2021): Basic Operations on Images (2021-07-05) https://docs.opencv.org/master/d3/df2/tutorial_py_basic_ops.html (Access 2021-09-10, 09:16 MEZ)

OpenCV (Object Detection, 2021): Thresholding Operations using inRange (2021-07-05) https://docs.opencv.org/3.4.15/da/d97/tutorial_threshold_inRange.html (Access 2021-08-05, 10:05 MEZ)

OpenCV (Shape, 2021): Structural Analysis and Shape Descriptors (2021-07-05) https://docs.opencv.org/master/d3/dc0/group__imgproc__shape.html (Access 2021-10-01, 09:33 MEZ)

OpenCV (Thresholding, 2021): Basic Thresholding Operations (2021-07-05) https://docs.opencv.org/3.4.13/db/d8e/tutorial_threshold.html (Access 2021-09-25, 12:15 MEZ)

PyImageSearch (Range-Detector, 2016): imutils_range-detector at master (2016-05-12) https://github.com/PyImageSearch/imutils/blob/master/bin/range-detector (Access 2021-09-22, 11:42 MEZ)

Python Software Foundation (Classes, 2021): 9. Classes (2021-09-11) https://docs.python.org/3/tutorial/classes.html (Access 2021-09-11, 10:48 MEZ)

Python Software Foundation (FAQ, 2021): Programming FAQ (2021-09-16) https://docs.python.org/3/faq/programming.html#what-are-the-rules-for-local-and-global-variables-in-python (Access 2021-09-16, 09:54 MEZ)

Python Software Foundation (Frontend, 2021): tkinter.ttk — Tk themed widgets (2021-09-13) https://docs.python.org/3/library/tkinter.ttk.html (Access 2021-09-13, 22:36 MEZ)

Python Software Foundation (Interface, 2021): tkinter — Python interface to Tcl/Tk (2021-09-13) https://docs.python.org/3/library/tkinter.html#tkinter.Tk (Access 2021-09-13, 22:37 MEZ)

Python Software Foundation (Password, 2021): pwd — The password database (2021-09-15) https://docs.python.org/3/library/pwd.html (Access 2021-09-15, 09:48 MEZ)

Python Software Foundation (Self-Argument, 2021): 3. Data model (2021-09-16) https://docs.python.org/3/reference/datamodel.html (Access 2021-09-16, 11:36 MEZ)

Python Software Foundation (Sqlite3, 2021): sqlite3 — DB-API 2.0 interface for SQLite databases (2021-09-09) https://docs.python.org/3/library/sqlite3.html (Access 2021-09-09, 07:48 MEZ)

Python Software Foundation (Threading, 2021): threading — Thread-based parallelism (2021-09-22) https://docs.python.org/3/library/threading.html (Access 2021-09-22, 10:37 MEZ)

Python Software Foundation (Tkinter, 2021): tkinter — Python interface to Tcl/Tk (2021-09-06) https://docs.python.org/3/library/tkinter.html#module-tkinter (Access 2021-09-06, 07:34 MEZ)

Roseman, *Mark* (Basic Widgets, 2021): Basic Widgets (2021) https://tkdocs.com/tutorial/widgets.html#entry (Access 2021-09-15, 09:37 MEZ)

Roseman, *Mark* (Label, 2021): Label (2021) https://tkdocs.com/widgets/label.html (Access 2021-09-15, 09:14 MEZ)

Roseman, *Mark* (Window, 2021): Windows and Dialogs (2021) https://tkdocs.com/tutorial/windows.html (Access 2021-09-20, 10:14 MEZ)

Roth, *Klaus* (Performance Factors, 2017): Merkmale eines Fußballers (2017-01-15) http://www.spitta-medizin.de/sport/fachinformationen/sportwissenschaft/artikel/der-perfekteschuss-das-anforderungsprofil-eines-fussballspielers.html (Access 2021-08-06, 12:13 MEZ)

SAP SE (SAP, 2021): Globale Unternehmensinformationen (2021-07-15) https://www.sap.com/germany/about/company.html (Access 2021-07-31, 12:35 MEZ)

Schwan, *Ben* (Podcast, 2018): Regie: KI, Buch: KI, Gesprochen von: KI (2018-06-06) https://www.heise.de/-4068201 (Access 2021-08-02, 19:33 MEZ)

Shipman, *John* (Canvas Image, 2013): 8.9. Canvas image objects (2013-12-31) https://tkdocs.com/shipman/create_image.html (Access 2021-09-14, 22:46 MEZ)

Shipman, *John* (Canvas Widget, 2013): 8. The Canvas widget (2013-12-31) https://tkdocs.com/shipman/canvas.html (Access 2021-09-14, 22:48 MEZ)

Shipman, *John* (Images, 2013): 5.9. Images (2013-12-31) https://tkdocs.com/shipman/images.html (Access 2021-09-14, 22:51 MEZ)

SQLite (SQLite, 2021): The SQLite Query Optimizer Overview (2021-06-18) https://www.sqlite.org/optoverview.html#introduction (Access 2021-09-18, 10:04 MEZ)

STATSports Group Limited (STATSports, 2020): IMPROVE YOUR GAME WITH APEX ATHLETE SERIES (2020-06-11) https://statsports.com/apex-athlete-series (Access 2021-08-01, 12:35 MEZ)

TRACKTICS GmbH (TRACKTICS, 2020): ALLES RAUSHOLEN MIT TRACKTICS (2020-02-11) https://tracktics.com/spieler/ (Access 2021-08-02, 12:27 MEZ)

Zeppenfeld, *Björn* (Neymar, 2021): Rekordtransfers im internationalen Fußball bis 2021 (2021-10-01) https://de.statista.com/statistik/daten/studie/12465/umfrage/rekordtransfers-die-10-teuersten-transfers-im-fussball/ (Access 2021-07-31, 12:55 MEZ)

Declaration in lieu of oath

I hereby declare that I produced the submitted paper with no assistance from any other party and without the use of any unauthorized aids and, in particular, that I have marked as quotations all passages, which are reproduced verbatim or nearby verbatim from publications. Also, I declare that the submitted print version of this thesis is identical with its digital version. Further, I declare that this thesis has never been submitted before to any examination board in either its present form or in any other similar version. I herewith disagree that this thesis may be published. I herewith consent that this thesis may be uploaded to the server of external contractors for the purpose of submitting it to the contractors' plagiarism detection systems. Uploading this thesis for the purpose of submitting it to plagiarism detection systems is not a form of publication.

Stuttgart, 04.11.2021