TU WIEN
DEPARTMENT OF
GEODESY AND
GEOINFORMATION

Automated scoring in climbing competitions

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Geodesy and Geoinformation

by

Maximilian Jan Michenthaler
Registration Number 01325555

at the Department of Geodesy and Geoinformation

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Norbert Pfeifer
Co-Advisor: Univ.Ass. Dipl.-Ing. Bakk.techn. Michael Wimmer
Univ.Ass. Dipl.-Ing. Bakk.techn. Claudio Navacchi

Vienna, 28th March, 2022

Maximilian Jan Michenthaler | Norbert Pfeifer

Erklärung zur Verfassung der Arbeit

Maximilian Jan Michenthaler

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 28. März 2022

Maximilian Jan Michenthaler

Abstract

With all the technological advances and research in sports tracking and human movement analysis in climbing due to the increasing popularity of the sport, no evidence of the use of an automated scoring system for climbing competitions is found in literature. The lack thereof motivated the evaluation of state of the art technology regarding its fitness for such task and its practical viability. Therefore, in this thesis a framework for an automated scoring system for climbing competitions with a focus on lead climbing is built. A novel one-camera approach is taken and a low-cost system is proposed where a single perspective from behind the climber is used to provide video data. The system utilizes the video data from the camera in conjunction with object tracking, color masking and image processing techniques in Python to derive scores post-competition. With YOLOv3, state of the art computer vision technology is used for climber detection and tracking, while the climbing holds are manually marked and further delineated with HSV color masks in a simple user interface. The declaration of holds as "reached" utilizes difference calculation with the structural similarity (SSIM) index and thresholds it against empirically derived values. Two video resolutions, 4k and 1080p, are used for building and testing the system. The results from the testing are manually reviewed and provide insights for the system *integrity* (the system not failing) with mean *integrity* values for the tested video resolutions of 82.6% and 76.6% respectively. The lower *performance* values (correctly declared holds) of 48.7% and 40.4% respectively show potential for improvement. The workflow established in this thesis yields a usable scoring system for post-competition reviews and serves as a proof of concept for possible future developments in regards of automated scoring systems for climbing competitions.

Kurzfassung

Trotz all des technologischen Fortschritts und der Forschung im Bereich des Sporttracking und der Analyse menschlicher Bewegungsvorgänge beim Klettern, die mit der wachsenden Beliebtheit des Sports einhergehen, wurden keine Beweise für die Nutzung eines automatisierten Auswertungssystems bei Kletterwettbewerben in der Fachliteratur gefunden. Das Fehlen eines solchen Systems motivierte die Evaluierung moderner Technologien, um die Eigung und die Praktikabilität dieser für solch eine Aufgabe zu ermitteln. Daher wird in dieser Thesis ein Framework für ein automatisiertes Auswertungssystem für Kletterwettbewerbe aufgebaut. Ein neuartiger ein-Kamera Ansatz wird gewählt und ein low-cost System vorgeschlagen, bei welchem eine einzige Kameraperspektive, die den Kletterer von hinten zeigt, als Datenquelle verwendet wird. Das System nutzt die Videodaten in Verknüpfung mit Objektverfolgungs-, Farbmaskierungs- und Bildverarbeitungstechnologien in Python, um die Punktzahl nach dem Bewerb zu ermitteln. Mit YOLOv3 wird eine hochmoderne Computer-Vision Technologie verwendet, um den Kletterer zu erkennen und zu verfolgen. Die Klettergriffe werden von Hand markiert und mit HSV Farbmaskierung in einem simplen User-Interface weiter von der Wand abgegrenzt. Die Deklarierung der Griffe als „erreicht“ erfolgt anhand von Differenzenberechnung mit dem Structural Similarity (SSIM) Index, durch das Vergleichen von diesem mit empirisch bestimmten Grenzwerten. 1080p und 4k Videos werden verwendet um das Auswertungssystem aufzubauen und zu testen. Die Ergebnisse werden manuell überprüft und geben Aufschluss über die *Integrity* (ein nicht Versagen des Systems) mit mittleren *Integrity* Werten von jeweils 82.6% und 76.6% für die beiden getesten Auflösungen. Die geringere *Performance* (korrekt als „erreicht“ erkannte Klettergriffe) von jeweils 48.7% und 40.4% zeigt Potential für Verbesserungen. Der Workflow, der in dieser Thesis etabliert wurde, liefert ein Auswertungssystem, welches für Überprüfungen nach einem Wettbewerb herangezogen werden kann und als Proof-of-Concept für mögliche zukünftige Entwicklungen im Bezug auf automatisierte Auswertungssysteme bei Kletterwettbewerben dient.

Danksagung

Zunächst möchte ich mich bei meinen Betreuen Michael Wimmer und Claudio Navacchi bedanken. Diese standen zu jeder Tages- und fast jeder Nachtzeit für jedwede Beratung zur Verfügung. Nur durch ihre fachliche und menschliche Kompetenz war es mir möglich die Diplomarbeit in dieser Form zu verfassen.

Des Weiteren möchte ich mich bei der Kletterhalle Wien, sowie bei Viktor Wiesner und Anna Tsombanis, für die Gelegenheit und den reibungslosen Ablauf der Datenbeschaffung bedanken.

Die Forschungsgruppe der Geoinformation darf nicht unerwähnt bleiben, da mir diese die notwendigen Rechenkapazitäten zur Verfügung gestellt habt. Dafür danke ich herzlichst.

Außerdem möchte ich mich bei Patrick Greger, Patrick Hrncir, Lukas Gokl und Christian Watzinger von ganzem Herzen bedanken, da sie mir die Möglichkeit zur Entspannung, zwischen dem Schreiben und Arbeiten, gegeben haben.

Auch bei meiner Familie, meinem Bruder Paul und meinen Eltern Andrea und Stefan, die mich in jeder Hinsicht unterstützt und mir dieses Studium erst ermöglicht haben.

Zuletzt möchte ich mich noch bei Nathalie Roser bedanken die in dieser Zeit meine emotionale Stütze war und ohne jene ich nicht in der Lage gewesen wäre eine Arbeit diesen Ausmaßes durchzuführen.

Contents

Chapter 1

Introduction

1.1 Motivation

As sports climbing in general has become increasingly popular over the last decade (commercial climbing industry growth rates of 6.9% in 2016, 10% in 2017, and 11.8% in 2018; Olhorst, 2019), the scientific interest in this activity is also thriving. Subjects such as human motion measurement and analysis, performance and motivation enhancement, psychological aspects of climbing or activity tracking are being researched intensively (e.g., Aladdin and Kry, 2012; Dovgalecs et al., 2014; Kosmalla et al., 2016; Ebert et al., 2018 Torino et al., 2020; Reveret et al., 2020; Iguma et al., 2020; Efstratiou, 2021). While few of the researchers focus solely on the tracking and competition scoring aspects, most of the publications implement methods of detection or tracking of the climber and/or the holds.

The whole outcome of a climbing competition is primarily dependent on the score the athlete gains while climbing. While trained personnel is tasked with the judging whether a hold is reached or not, an automated scoring system could assist them. The current process of scoring in climbing competitions is described by the International Federation of Sports Climbing (IFSC; IFSC, 2021). At least 2 people, a national judge and a time keeper, record the climbing time and the achieved score for each competitor. While the criteria for counting holds as reached or gripped are described in depth, it is not mentioned, whether the scores are applied manually, or if the IFSC is already using an automated system. This leads to the assumption, that the introduction of an automated scoring system would benefit climbing competitions greatly.

1.2 Aim

This thesis aims to investigate the viability of a one-camera low-cost system for automatic scoring in climbing competitions with a focus on lead climbing. Throughout the process of building the such a system, detection technologies are evaluated if they fit the task and if they are mature enough to work with only a single camera perspective to produce useful results. Furthermore, processing times and system performance are investigated, to check if a near real-time scoring system is viable. The final goal is to build a system, that takes a video facing the back-view of the climber and enables a potential user to track the progress of the climber built on the following points:

- train and use automatic computer vision detection methods

- build a rudimentary graphical user interface (GUI) to increase the control over the processes
- automatically detect whether a hold is reached/gripped or not using primarily the overlap between the climber's hand and the hold
- low-cost system with only one camera
- establish ground truth for system performance and integrity evaluation

This makes it possible to investigate the proposed research questions. Additionally, an important goal is to evaluate existing technologies and projects in the scientific vicinity to realize a proof of concept, that can serve as a basis for possible future developments.

1.3 State of the art

With Strickler et al. (1994), the only system listing scoring as one of its main purposes was found in a United States patent document, dating as far back as 1994. The proposed system uses buttons mounted underneath the holds to register the climbers progress. Since this is only a patent document, no statements about the usability, performance or applicability can be made.

Some systems use wearable sensors or devices, such as inertial measurement units (IMU) (e.g. Dovgalecs et al., 2014) and electromyography (EMG) sensors (Kalyanaraman et al., 2015) to track the climber's movement directly. Others measure the forces inflicted onto the holds with e.g force torque (Aladdin and Kry, 2012) or capacitive sensors (Parsons et al., 2013) to derive climber motion data. Another way of detecting and tracking a climber are optical systems. Most prominently used are RGB-D cameras such as the microsoft kinect system (e.g., Wiehr et al., 2016; Pandurevic et al., 2019). Intel LiDAR depth cameras (time of flight system) (Efstratiou, 2021) or drones (Reveret et al., 2020) are used in some cases.

Additionally, some papers propose combined solutions using wall- or climber-mounted sensors in conjunction with optical systems (e.g., IROZHLAS, 2019; Cordier et al., 1994; Tiator et al., 2018). One other system uses simple RGB cameras such as the proposed solution in this thesis, but uses seven cameras with additional markers on the participants and a climbing hold force measurement system (Iguma et al., 2020). In Richter et al. (2020), an overview of different technologies used for climbing motion analysis or tracking was found. All these different methods could potentially be used to track a climber's progress along a predefined route, but none of these systems uses only a single RGB camera as this thesis proposes.

The limited capabilities of a single RGB camera leave only a few methods for tracking and detecting the climber. The proposed solution is utilizing YOLOv3 for the detection task in this thesis. With the real time detection capabilities (Redmon et al., 2016b), the algorithm lends itself for the task of detection in sports applications.

In Thulasya Naik et al. (2021) YOLOv3 is used to detect soccer players and balls successfully with high precision with the same Microsoft COCO (Common objects in Context) dataset pre-trained weights for person detection (Lin et al., 2014) proposed in this thesis (section [2.2.2.2]). Even the previous version of the algorithm, YOLOv2, has seen successful use in scientific studies regarding detection and tracking in sports

(S. Zhang et al., 2019) to build an intelligent tracking system for curling. Additionally, YOLOv4, the newest version of the algorithm, has also already been utilized use as a basis for tracking and detecting athletes (Y. Zhang et al., 2020) with DeepSort (Wojke et al., 2018). YOLOv4 shows very good performance and is described as the technical-grade object detection algorithm taking over as "[...]Best object detection algorithm[...]"[Y. Zhang et al., 2020] after YOLOv3.

Chapter 2

Data and Software

2.1 Data Acquisition

2.1.1 Acquisition Site

The photos and videos that were used throughout this thesis were taken at the Naturfreunde Wien Kletterhallen GmbH climbing facility at Erzherzog Karl Straße 108, 1220 Wien. It is an indoor climbing facility with 2300m² of rope climbing area with wall heights up to 16m. The best suited climbing routes, in terms of good lighting, reasonable contrast between hold and wall colors and enough space to capture the ascent, were selected for filming climbing processes (Figure 2.1). Aside from the aforementioned factors the walls and routes were chosen according to the volunteers preferences. The resulting photos and videos were then used for training detection algorithms and testing the automatic scoring system.

(a) *Routes for the Videos 1 to 6, 9 and 10.*

(b) Routes for the Videos 9, 8 and 11 to 14.

Figure 2.1 *The climbing walls and routes from the Kletterhalle Wien recorded in the last data acquisition session and used during testing of the automated scoring system.*

2.1.2 Equipment

All videos were captured with the camera set up on a tripod to minimize movement during the recordings (Figure2.2). The setup was leveled with a circular level before the the vertical camera angle was adjusted for the desired view.

For the data acquisition two different cameras were used during three sessions at the Kletterhalle Wien. The Olympus OM-D E-M5 Mark II (Table 2.1) was used with a 12-100mm lens (Figure 2.3) during the first and second data acquisition sessions at the climbing facility. This E-M5 Mark II camera features a 4/3 Live MOS Sensor with 16.1 mega pixels and a *Supersonic Wave Filter*, which is an image sensor dust reduction system, that removes dust from a protective glass plate in front of the sensor with supersonic vibrations (Olympus, 2021a). This could prove advantageous in a chalk dust rich area such as a climbing facility.

The used ED 12-100mm F4 IS PRO lens (Table 2.1) features a focal length from 12-100mm, which is equivalent to 24-200mm with a 35mm movie film gauge (Olympus, 2021c). The whole zoom range was used to accommodate the different wall-to-camera distances and take close-up pictures of the climbers and holds. The aperture was set to f/5.6. The photos were taken in the highest possible resolution of 4608x3456 pixels. The videos were taken in the highest possible resolution setting, 1920x1080 pixels (1080p), and 60 frames per second. The resulting videos and photos, were used for the initial testing of detection training and data processing.

For the third data acquisition session the OM-D E-M1 Mark III (Table 2.1) was used in conjunction with a 12-40mm lens (Figure2.4). The E-M1 Mark III is equipped with a 4/3 Live MOS Sensor with an effective resolution of 20.4 mega pixels. It also features the fore-mentioned *Supersonic Wave Filter*. One major improvement and the reason why this camera was chosen over the previously used E-M5 Mark II is the ability to record 4k videos (Olympus, 2021b).

Figure 2.2 *The tripod mounted camera setup that was used for the data acquisition.*

The ED 12-40mm F2.8 PRO lens (Table 2.1) has a smaller focal length range, but a better possible light incidence with a minimum aperture of f/2.8 (Olympus, 2021d). This lens was chosen for the third data acquisition, as the previous recording sessions showed, that a higher focal length was not needed for the video recordings and the bigger aperture range was expected to be of higher utility. More importantly, the lower weight of the lens (382g), and higher weight of the OM-D E-M1 Mark III (580g), helped to provide a more stable platform for the video recordings, as the comparatively higher weight of the ED 12-100mm F4 IS PRO lens (561g) and the lower weight of the E-M5 Mark II (469g) resulted in occurrences of camera drift.

Figure 2.3 *OM-D E-M5 Mark II with the M.ZUIKO DIGITAL ED 12-100mm F4 IS PRO lens*

Figure 2.4 *OM-D E-M1 Mark III with the M.ZUIKO DIGITAL ED 12-40mm F2.8 PRO lens*

In the third data acquisition session only videos were captured. Based on previous experience they were better for testing and improving the automated scoring system, as the videos were able to capture a climber's ascent similar to the intended usage scenario of the system, namely the live-feed of a climbing competition. Both 3840x2160 pixel (4k) and 1080p videos were recorded to explore the impact of the resolution on the processing. The 4k and 1080p videos were recorded with the highest possible frame rates of 30 fps and 60 fps, respectively. The aperture was set to f/5 for all videos, as this value provided a good depth-of-field to keep the object's of interest, the climber and the holds, in the range of acceptable sharpness.

Table 2.1 *Important specifications of the used cameras and lenses.*

Olympus cameras	**OM-D E-M5 Mark II**	**Olympus OM-D E-M1 Mark III**
Sensor type	4/3 Live MOS Sensor	4/3 Live MOS Sensor
Number of pixels	16.1 million pixels	20.4 million pixels
Dust reduction	Supersonic Wave Filter	Supersonic Wave Filter
Max. photo resolution	4608 x 3456 pixels	5184 x 3888 pixels
Max. video resolution/framerate	1080p at 60fps	4k at 30fps
Weight	469g	580g
M.Zuiko lenses	**ED 12-100mm F4 IS PRO**	**ED 12-40mm F2.8 PRO**
Focal length	12-100mm	12-40mm
Min. aperture	f/4	f/2.8
Weight	561g	382g

2.1.3 Data

The resulting data includes 2227 images of holds and climbers, as well as 19 videos of climbing processes (Table 2.2). Four volunteers and myself were recorded for the climber pictures and videos, namely four males of varying physique and one female person.

Table 2.2 *The Data acquired during the three recording sessions at the Kletterhalle Wien*

Data acquisition Session Nr.	**Session 1**	**Session 2**	**Session 3**
Recording date	16.10.2020	26.05.2021	15.09.2021
Resulting Data	1385 picture	842 pictures; 5 videos in 1080p	7 videos in 4k; 7 videos in 1080p
Data usage	HAAR-Cascade and YOLO training	HAAR-Cascade and YOLO training; initial system testing	Final scoring system testing

The images were used for experimenting with the training of specialised detection algorithms (section [3.3]) for either climbers and holds, where they were used as positive and negative image examples for HAAR-Cascades (section [2.2.2.1]) and with labeled contents as image set for YoloV3 (section [2.2.2.2]). The videos were split into their frames and used for building, refining and testing the scoring script (section [3.5]). The data from the first two acquisition sessions was mostly used for experimenting with different detection methods and building the processing framework. Experience from these sessions and experiments helped to determine the requirements the recordings had to meet for further development of the automated scoring system. Therefore, the 14 videos recorded in the last acquisition were the most useful for many important development steps (sections [3.4], [3.5]). Of theses videos seven are in 4k resolution and

seven in 1080p. Additionally, half the videos show the female volunteer resulting in a good variety in both resolution and content.

2.2 Software

2.2.1 Python

For the purposes of data processing a Python 3.7 interpreter was used in a Pycharm IDE. The most important packages that were used are described in this section.

2.2.1.1 OpenCV2

The Open Computer Vision Library v3.4.2 (OpenCV) is an open source software library, that provides tools for computer vision and machine learning applications (OpenCV, 2021). OpenCV is written natively in C++. Python is used as an interface to integrate the library into the data processing to utilize it on a wide variety of image processing tasks. OpenCV was implemented in various instances of image processing throughout this thesis such as the reading and writing of images and videos, switching between colour models, edge detection, training the HAAR-Cascade detection (section [2.2.2.1]) and comparing images.

2.2.1.2 Scikit-image

Scikit-image v0.18.1 is an open source image processing library (Van der Walt et al., 2014). Throughout this thesis, it's structural similarity (SSIM) function is used for comparing images.

2.2.2 Detection algorithms

For the detection of the climber and holds, different detection algorithms were assessed. The idea was to detect the holds before the ascent of the climber and the climber themself during said ascent. The bounding boxes, that were derived with these algorithms, served as a cornerstone for the scoring process.

The use of HAAR-Cascades (named after the HAAR-like features it uses, which are in turn named after the structural similar HAAR-wavelets which are based on the HAAR-sequence proposed by Alfréd Haar in Haar, 1910) was proposed, as it represents a detection algorithm where the detection features are less of a blackbox than with most state of the art algorithms. Only the weights of the features are trained with machine learning algorithms on test data, while the HAAR-like features are determined beforehand (Viola and Jones, 2001a).

The YOLO (You Only Look Once) algorithm is a very fast detection algorithm and therefore well suited for detection tasks needed for the automated scoring system. It's real time detection capabilities, provided sufficiently strong hardware is used, and good performance (Redmon et al., 2016b) lend itself for the task at hand. Most other state of the art algorithms are fully convolutional neural network (CNN) based such as YOLO, but outperformed by it due to its unified detection approach (section 2.2.2.2).

2.2.2.1 HAAR-Cascades

HAAR-Cascades are a family of detection algorithms that "[...] use a Cascaded Reduced Set Vector [RSV; author's note] expansion of a Support Vector Machine (SVM)"[Rätsch et al., 2004]. These RSVs have a HAAR-like structure and HAAR-like features (Rätsch et al., 2004). These HAAR-like features are geometric features which are similar to HAAR basis functions (Viola and Jones, 2001a). In Figure 2.5 some HAAR basis functions are shown. These enable a very fast SVM kernel evaluation by utilization of the *Integral image*, an image representation proposed by Viola and Jones (2001b) and therefore provide a means for reduction of the computational complexity of a SVM classifier with negligible loss of accuracy (Rätsch et al., 2004). In Figure 2.6 one can see a usage example of HAAR-like features in face detection as proposed in Rätsch et al. (2004).

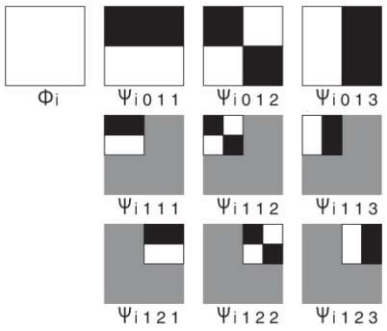


Figure 2.5 *Examples of HAAR basis functions; they take 1, 0, and -1 in white, gray, and black regions. Figure from Okabe et al. (2004)*

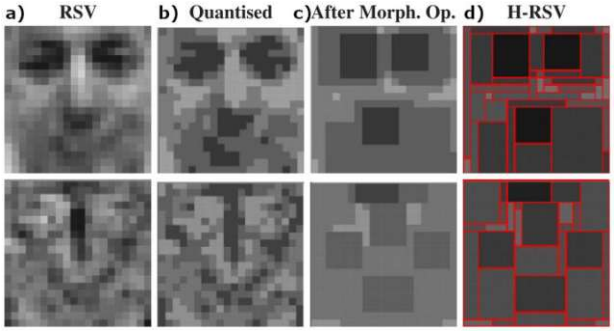


Figure 2.6 *Left to right: a) Example of the Haar-like approximation of a face and an anti-face such as RSV; b) discretized vectors by four gray levels; c) smoothed vector by morphological filters; d) H-RSVs with computed rectangles. Figure from Rätsch et al. (2004)*

The cascade part refers to the cascaded evaluation that is used to classify an image patch. First, the hyperplane is approximated by a single HAAR-RSV (H-RSV). If the classification function for the patch is negative, it is classified as *non-face* and the evaluation stops. This process is repeated incorporating more H-RSVs, to make the classifier more complex, and rejecting as early as possible, until a positive evaluation using the last H-RSV is reached. Then, the full SVM is used to classify the image patch (Rätsch et al., 2004).

The training of the HAAR-Cascade classifiers for testing purposes was done with a simple GUI tool called "Cascade Trainer GUI" (Ahmadi, 2017), following the instructions from Tejas R. Phase (2020). HAAR-Cascade detection is implemented with the OpenCV2 package for Python.

2.2.2.2 YOLOv3

YOLO is a real-time detection algorithm where a single CNN simultaneously predicts multiple bounding boxes. These bounding boxes are predicted by resizing the image, running said CNN and thresholding the resulting detections by the YOLO model's confidence. For each box class probabilities are calculated. In YOLO the object detection is reframed as a single regression problem, straight from image pixels to bounding boxes

and class probabilities, hence You Only Look Once (YOLO) at an image to predict what objects are present and where they are. Since the detection is framed as a regression problem, no complex pipeline is needed, making YOLO very fast in comparison to other detection systems, such as deformable parts models, DPM (Felzenszwalb et al., 2010) or R-CNN, Regions with CNN features (Girshick et al., 2014). The algorithm is trained on full images, meaning YOLO reasons globally about the image when making predictions, so it implicitly encodes contextual information about classes as well as their appearance (Redmon et al., 2016b).

The input image is divided into an S x S grid (S being the width and height in number of grid cells as seen in Figure 2.7), where the responsibility for objects is assigned to grid cells by checking whether the object's center falls into one of those grid cells. Bounding boxes and confidence scores for those boxes are predicted cell-wise. These scores reflect how confident the model is, that the box contains an object and how accurate the predicted box is. A confidence score of zero indicates the lack of an object in that cell. Conditional class probabilities are also predicted grid cell-wise. These probabilities are conditioned on the grid cell containing an object. Only one set of class probabilities per grid cell is predicted. The class probabilities multiplied with the box confidence score results in a class specific confidence score for each box (Redmon et al., 2016b). In Figure 2.7 the YOLO process is displayed in a simple visual representation.

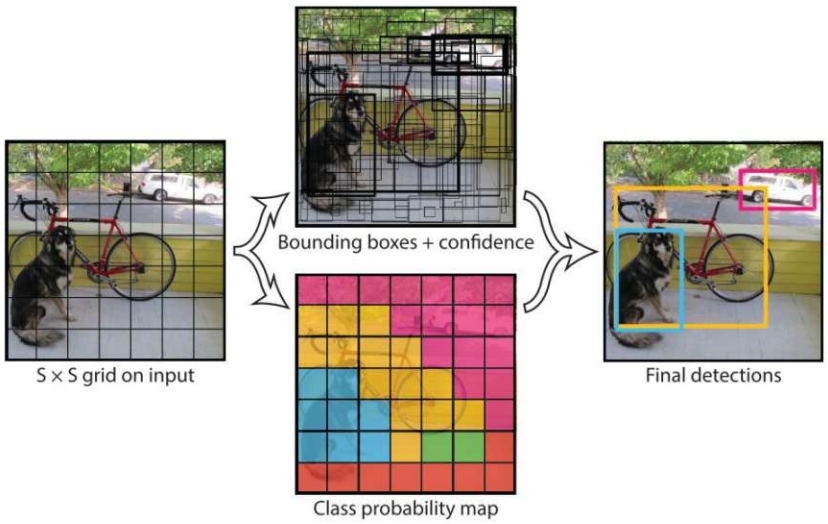


Figure 2.7 *A visual representation of YOLO's detection model. Figure from Redmon et al. (2016b).*

YOLOv3 is an improved further developed Version of YOLO (Redmon and Farhadi, 2018). A TensorFlow 2 (version 2.1) Python implementation of YOLOv3 was used throughout data processing and script development with TensorFlow being an end-to-end open source platform for machine learning (*TensorFlow* 2021).

For training a set of weights with YOLO, the *LabelImg* graphical image annotation tool was used, which is a Python software that uses Qt (*Qt | Cross-platform software development for embedded desktop* 2022) for the graphical interface. With this tool the desired object can be marked, labeled and saved in the appropriate format needed for YOLO training. It is possible to either train using the pre-trained darknet feature

extractor weights as a base, also called transfer learning (Tensorflow, 2022b), or from random weights, called training from scratch.

Ultimately, YOLO's COCO pretrained set of weights was used to detect persons which were then labeled as climbers. This set represents weights that are already trained by Redmon and Farhadi (2018) on the Microsoft COCO dataset. It is an open source database of photos of 91 object types with a total of 2.5 million labeled instances in 328k images, that was made for detecting and segmenting objects found in everyday life in their natural environments (Lin et al., 2014).

Chapter 3

Methods

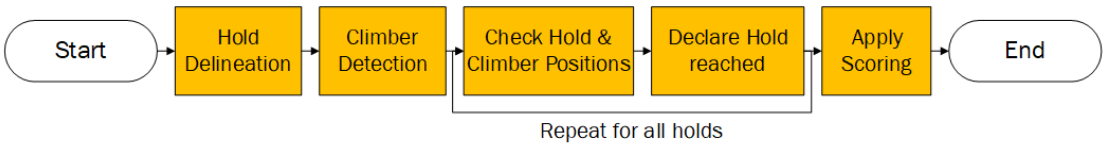


Figure 3.1 *Main steps required for the automated scoring.*

As discussed in Section [1.2] the most important components of an automated scoring system for climbing (Figure 3.1) are the delineation of holds, the detection of the climber, the comparison of the hold and climber position, the decision whether a hold is reached and the application of the score. According to this scheme, an overall concept was developed that tackles the implementation of the components using state of the art detection algorithms and image comparison tools to get the final score of the climber based on a video input. In this chapter, the main processing steps are discussed with focus on their practical implementation. The sections first describe the requirements for each step, followed by the proposed solution and the methods, that were used to achieve it. Afterwards, an overview of the data- and workflow is given.

3.1 General methodic remarks

As the aim was to create a relatively low-cost system, only one camera perspective is used, namely the camera pointing straight at the climbing wall from behind the climber. The positioning aimed to create an approximately perpendicular recording in the horizontal plane. Vertically, the camera was tilted between 0° - 45° in a way, that the whole climbing wall was covered. The distance to the climbing wall was chosen individually for each video to accommodate to the specific height of each route. As the camera was mounted on a tripod and not moved while recording, a stable platform and therefore a fixed position was assumed.

In respect to processing, two PCs from the GIS-Lab at TU Wien and a personal office PC were used. The PCs a TU Wien are equipped with an AMD Ryzen 7 2700x eight-core processor, 32GB of RAM and a NVIDIA GeForce GTX 1060 6GB graphics card respectively. The office PC features an AMD Ryzen 5 3600 six-core processor, 16GB of RAM and an AMD Radeon RX 5700 8GB graphics card. For YOLO training, the GIS-Lab computers were set up for parallel processing with the TensorFlow 2 framework. For all other tasks, only one GIS-Lab computer or the personal office device were used.

OpenCV uses a coordinate system that has its origin in the top left pixel of each image. All coordinate based computations refer to this system (Figure 3.2).

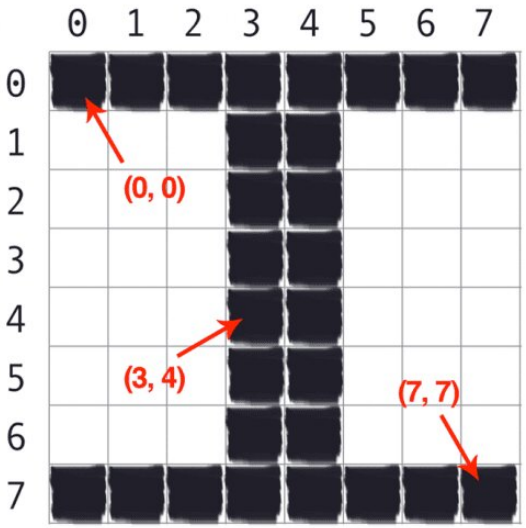


Figure 3.2 *A visualisation of the picture coordinate system used in OpenCV (Figure from Rosebrock (2021))*

Due to the restrictions of the used setup, other means of automatically detecting and marking holds, such as the use of photogrammetric targets on every hold, were not further investigated. Methods such as reflectors (e.g. IROZHLAS, 2019) or markers (e.g. Cordier et al., 1994) attached to the climber were considered, but not pursued, to avoid interfering with the climber and keep the system's complexity down. Systems with multiple cameras (e.g. Iguma et al., 2020) or drones (e.g. Tiator et al., 2018), as well as RGB-D (e.g. Pandurevic et al., 2019; Pandurevic et al., 2020; Wiehr et al., 2016; Kajastila and Hämäläinen, 2014; Kajastila et al., 2016 and Kosmalla et al., 2017) or LiDAR depth cameras (Efstratiou, 2021) were avoided, to keep the cost of the system low.

As mentioned in section [2.1.3], pre-recorded videos split into their individual frames were used to develop the scoring system. This procedure was chosen to reduce the computational expense during data import, as the videos would have to be split into frames anyways for image processing. Additionally, by splitting the videos into frames beforehand, parameter testing procedures could be conducted in less time. This limits the implementation in a way, that the system was built around these pre-recorded videos and not an actual live competition situation. The system accepts videos as input and was therefore theoretically set up to work with live-video feeds.

For the declaration of holds as "reached" and the scoring process, systems that use sensors in holds (e.g. Pandurevic et al., 2019; Pandurevic et al., 2020; Strickler et al., 1994; Quaine et al., 1997b; Quaine et al., 1997a; Quaine and Martin, 1999; Torino et al., 2020; Parsons et al., 2013 and Aladdin and Kry, 2012) are disregarded to keep the proposed system simple and cheap.

The scores itself were chosen in a way, that they could help with the evaluation, the first hold got a score of 1, the second 2 and so forth.

3.1.1 IFSC climbing and scoring process

The climbing process that is subject of this thesis follows very specific guidelines according to IFSC (2021). Lead competitions take place on artificial climbing wall with a minimum height of 12m, while the routes should be at least 15m long and 3m wide. For the sake of simplicity it is assumed, that the investigated climbing route does not contain overhangs in an extent that the climber would move towards the camera.

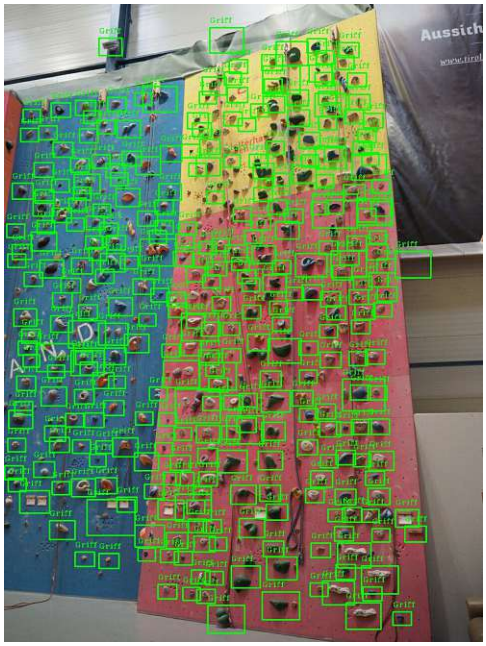
In climbing competitions the hold scores are determined by the chief routesetter in a "Topo", a graphical representation of the climbing route, with arbitrary values he sees fit. The scoring is executed by at least 2 people, a national judge and a time keeper who record the competitors score. This score represents the climber's progress on the route, each hold the climber reaches gains them a point. The further the climber climbs, the higher is their score. The final score determines the winner of a competition. In case of a tie, the elapsed climbing time is used as a tie-breaker (IFSC, 2021).

3.2 Hold delineation and determination

To check whether a hold is reached, its position must be known. Ideally the hold would be delineated by its pixel coordinates. A direct delineation from video-frames to exact pixel coordinates was unrealistic with the limitations of resolution and the aim for fastest possible processing times. Therefore a method for creating bounding boxes (BB) to get approximate hold positions was pursued. Once these bounding boxes were created they were used to further delineate the holds for more precise results.

3.2.1 Detection and bounding box determination

An automated detection of the holds was tested with HAAR-Cascades and YOLOv3. For the HAAR-Cascade detection the pictures from the first two data acquisition sessions [2.1] were used to train a detector with the "Cascade Trainer GUI" (Ahmadi, 2017). The pictures showing the holds were used as positive sample images and pictures of non hold objects (e.g.: quickdraws, rock climbers, rope segments etc.) were used as negatives. These pictures had to be prepared in a way, that they only show the positive or negative object and fit an aspect ratio of 1.33:1. This aspect ratio represents not only the size of the input images, but also dictates the size of the resulting bounding boxes. Using different picture sets, different HAAR-Cascade classifiers were trained. The default settings recommended by Ahmadi (2017) were used for training, with only the buffer-sizes adjusted to the computer specifications, the positive image usage set to 90% and the aspect ratio to 1.33:1, by setting the sample width parameter to 32.



(a) HAAR-Cascade detector results

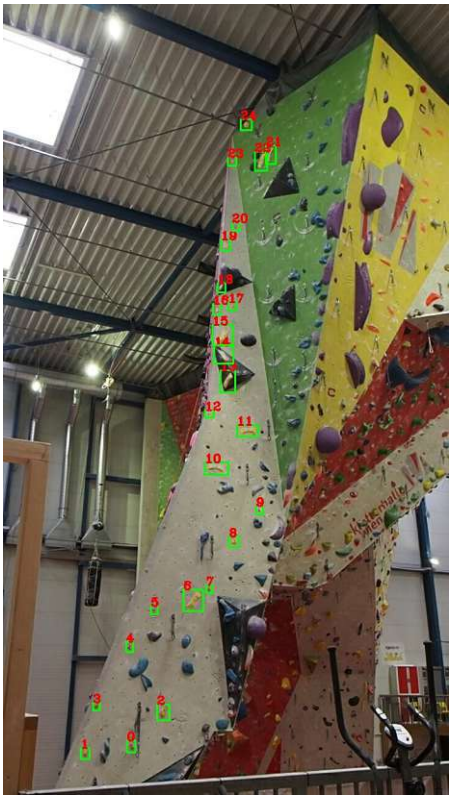
(b) Exemplary positive image

(c) Exemplary negative image

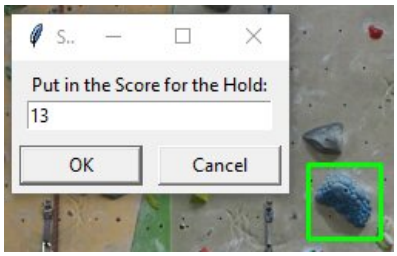
Figure 3.3 *a) Results of the HAAR-Cascade detector; b) positive image for HAAR-Cascade training; c) negative image for training.*

The resulting detector worked reasonably well in regards of precision, but was very slow as it took several minutes to classify the wall seen in Figure 3.3a. Additionally, multiple instances of *false positives* and *false negatives* (*FP* and *FN*) can be observed (cf. Figure 3.3a).

Considering these obstacles and an already existing infrastructure for YOLO training, an approach with a YOLOv3 detector was tested next, even though YOLO isn't well suited for detecting multiple objects close to each other (Redmon et al., 2016b). In scope of Figure 3.3a this may seem to be a dealbreaker, but climbing walls usually have less holds than the one pictured, especially during competitions. As YOLO training works differently than training HAAR-Cascades, not only pictures containing solely holds can be used as positive examples. To train a YOLO detector, a set of images and corresponding XML files must be created. These XML files contain bounding boxes and class labels for each object of interest in an image. After creating a training dataset, following the instructions from Z. Zhang (2019), transfer learing (Tensorflow, 2022b) was attempted using the pre-trained Darknet (Redmon, 2016a) feature extractor weights. Furthermore, training from scratch (Z. Zhang, 2019), also called training from random weights, was attempted. Unfortunately, none of the resulting detectors was able to handle the task of hold detection properly.



(a) Example of Marked holds



(b) BB and Score UI

(c) Score and BB closeup

Figure 3.4 *a) An example of a route with the marked holds and corresponding scores displayed; b) The user interface for creating bounding boxes and setting the scores, after the boxes lower right corner is defined, the window automatically pops up; c) A closeup of a bounding box and the corresponding score.*

After failing to create an automated hold detection, a manual approach was pursued. For manually creating bounding boxes, first a user interface (UI) was created. It takes a picture previously defined as baseline, generally the first frame of the video, and displays it scaled to the size of the used monitor. On this picture, rectangles can be drawn by clicking the top-left corner and dragging to the bottom right corner of the desired bounding box. Additionally, during this step the score of each hold can be defined. A window (Figure 3.4b) to enter said value appears, after marking a rectangle (Figure 3.4c).

The bounding boxes and the corresponding scores are then saved in a file for later processing steps. After all holds of the desired route are marked, an image containing all bounding boxes and corresponding scores (Figure 3.4a) is created and saved for easier evaluation of the results later on.

3.2.2 Further delineation

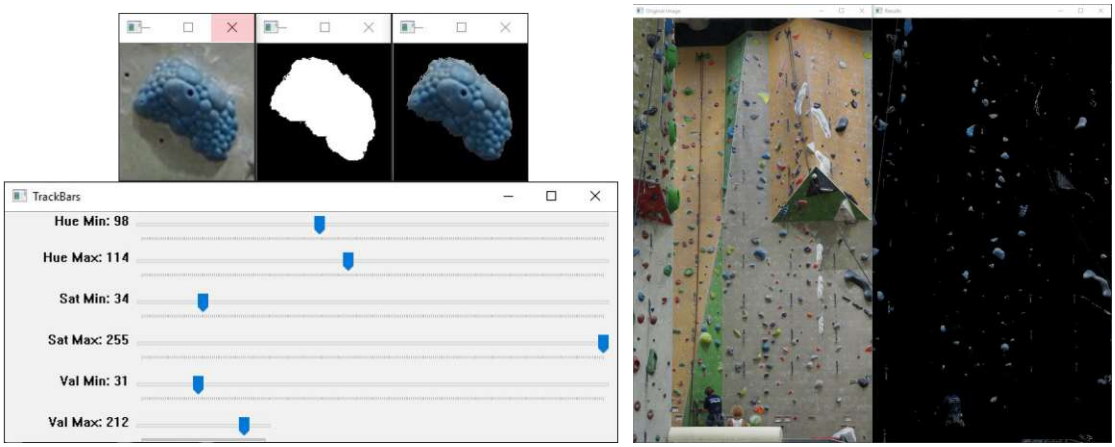
Since the bounding boxes created in the previous step are only a rough representation of the hold's extent, a more precise delineation was desired. At first, a simple contour extraction was performed using the OpenCV2 library (OpenCV, 2018b) to find the contour with the largest surface area, which should represent the hold. The contour in this context is an outline of an object in the image, derived with the OpenCV2 structured edge detecion function (OpenCV, 2018a). Even though this extraction could be tuned

quite well for single images, shadows, wall texture (Figure 3.5a) and variances in contrast between holds and walls pose serious problems for using this method on bigger scales.

(a) *Examplary problematic contour* ***(b)*** *Examplary good contour*

Figure 3.5 *a) Extracted contour; the problematic shadow and abrasion on the wall can be seen clearly; b) Example of a well extracted contour.*

Considering these obstacles, image segmentation using a colormask with the hue, saturation, value (HSV) color model to extract the holds from the bounding boxes was implemented. The HSV color model is more robust towards external light changes than RGB/BGR (red green blue/blue green red) systems and is therefore better suited for the color delineation tasks. The colormasking was achieved by building a UI where the baseline image (section [3.2.1]) is displayed and six sliders are provided, as seen in Figure 3.6a, with a single hold. These six sliders represent the maximal and minimal values of the three HSV parameters, which can be moved to adjust to the colour of the holds of the desired route. OpenCV usually stores images in a Blue/Green/Red (BGR) color model, therefore they have to be transformed to HSV first. While Figure 3.6a only shows a single hold, the color picking process incorporates the whole baseline image of the route (Figure 3.6b), to compensate potential differences within the HSV values between the holds. The resulting HSV ranges are saved to a file for later purposes (section [3.4]).



(a) *Color picker UI with an examplary hold* ***(b)*** *Color mask applied to the entire wall*

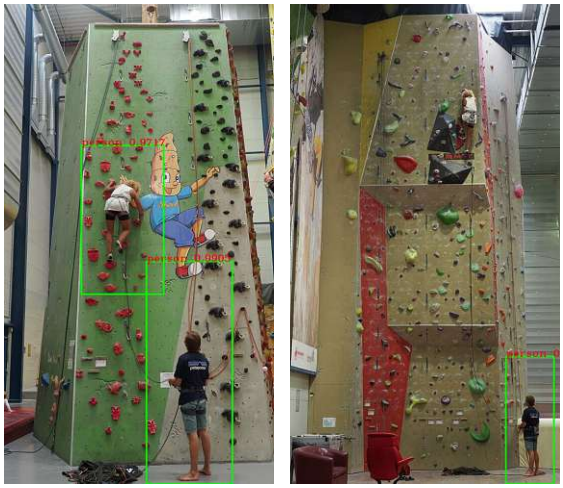
Figure 3.6 *a) Color picker UI with exemplary color segmentation; b) Color mask with the values picked in 3.6a applied to the entire wall.*

3.3 Climber detection

The detection of the climber is a crucial part of the system, as their position has to be known to derive the climber's route progress. The climber has to be located in many video-frames, ideally in real time, making a manual approach unrealistic. The detection algorithm YOLOv3 (section [2.2.2.2]) was predestined for this task, as it is a very capable state of the art algorithm for live detection and the algorithm's native outputs are bounding boxes with confidence scores. The positions of the climber as bounding box coordinates were desirable, as they are easy to work with and the holds were already stored in the same format (section [3.2]). In contrast to this, the bounding boxes are not the best solution, as they are sub-optimal to detect the movement of the climber's hand accurately. Regardless of this, the bounding boxes from the climber detection are an easy solution at hand to create an approximate area to restrict the overlap search to (section [3.4]).

It was decided, that a custom YOLOv3 detector should be trained for this task. This custom climber detector was trained according to Z. Zhang (2019) with pictures provided beforehand in addition to data from the first two acquisition sessions. Five sets of training data were created and trained for 200-1000 epochs in nine instances. The batch size, steps per epoch and validation steps were adjusted to the size of the training dataset to accommodate parallel computing (Tensorflow, 2022a), while the training rate and other parameters were left on their default values as suggested by Z. Zhang (2019) . After multiple attempts of creating a reliable detection, the custom detector only ever worked on the same pictures it was trained on (Figure 3.7).

Figure 3.7 *Custom detector working on training images; the bounding box around the climber is much closer to the climber compared to the pre-trained YOLO; the method of comparing hold and climber bounding boxes used in this thesis (section 3.4) would not work with this custom detector*



(a) *pretrained YOLO* ***(b)*** *YOLO failing*

Figure 3.8 *a) the pretrained YOLO is working even with uncommon poses; b) an instance of the pre-trained YOLO failing*

As already stated in section [2.2.2.2] the COCO pre-trained set of weights by Redmon

and Farhadi (2018) was ultimately used to detect persons. These detected persons were then relabeled as climbers, drawn on the video-frame they were detected on (Figure 3.8a) and saved to a list. Despite YOLOs good performance, it still failed sometimes, as seen in Figure 3.8b. If no climber is detected in a frame, a faux climber with very small x- and y-coordinates $[0, 0; 1e-12, 1e-12]$ is saved instead, as one climber per frame is needed to ensure smooth processing. If multiple climbers are detected in a frame, only the bounding box with the smallest coordinates, the assumed climber, is saved. This measure is taken to have only one bounding box per image and to make it therefore possible to refer to the bounding box and image with the same index number. These bounding boxes were then used to compare the climber and hold positions. To increase utility of the bounding boxes, a further delineation with an additional YOLO hand detection was considered, but no sufficiently accurate pre-trained hand detection was found. A self-trained hand detection wasn't an option, due to the lack of training data, as at least 2000 images are recommended (Bochkovskiy et al., 2020).

3.4 Declare holds reached

Following the delineation of bounding boxes for climbers and holds, the next step was to implement a way to declare a hold "reached" or "gripped". It has to be stated, that with image differences and only one camera angle the resulting algorithm won't be able to differentiate between a simple overlap of a hand and a hold or if a hold is really gripped. Therefore, from here on the terms "reached" and "gripped" are used interchangeably, because the system is not set up to differentiate between the two states. Considering the system's restriction of only one camera angle showing the climber from behind, the possibilities for a reliable and accurate system are limited. It was chosen to focus on image differences for this part of the scoring system. Therefore, a hand in front of a hold counts as gripping it, assuming that gripping the hold is easier, than just keeping one's hand in front of the hold. Furthermore, the possibilities for distinguishing, if either a hand or another body part is the one in front of the hold are limited.

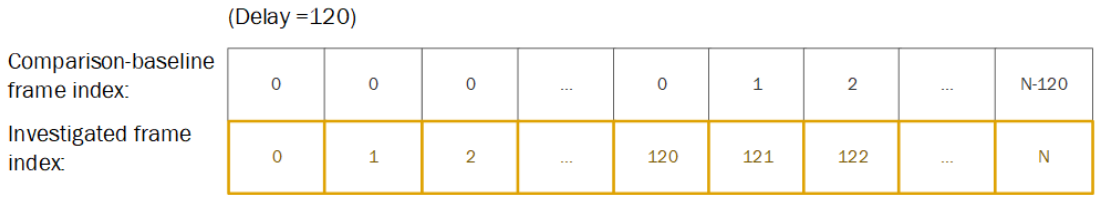


Figure 3.9 *graphical representation of the logic behind the comparison-baseline delay with an exemplary delay of 120 frames; N is either the index of the frame where the hold is considered "reached" or the last frame of the video.*

For every hold a check is conducted, if its bounding box is inside of the bounding box of the climber during any frame of the video. To achieve this, the overlap of both bounding boxes is computed and its area is compared to the area of the hold bounding box. If the two compared areas are, computational inaccuracies considered, the same, the hold bounding box is inside the climber bounding box. Furthermore the index of each frame is checked, if the modulo operation of said index with the frame-reduction parameter is 0. The frame-reduction parameter is defined as a value n that enables the system to check only every n-th frame (Table 3.1). It is introduced to reduce computational expense, as checking every frame didn't benefit the performance of the system during testing

noteworthy. Only if both conditions are met, the image comparison is executed. The next step is to compare the corresponding frame, where the hold bounding box is inside the climber bounding box, to a comparison-baseline image. This comparison-baseline image is set to be a number of frames before the currently investigated frame. It was experimented with a *comparison-baseline delay* of 30 to 120 frames for 4k, and 60 to 240 frames for 1080p videos equalling 1 to 4 seconds, to ensure that the climber in the comparison-baseline image is not near the currently investigated hold's bounding box. Longer delays were not investigated, as higher delays meant bigger discrepancies in lighting and shadows, which caused problems with the image comparison in previous testing. As long as the index of the current frame is lower than the *comparison-baseline delay*, the first frame of the video is used as the comparison-baseline (Figure 3.9).

This image comparison has several stages. If the colormask boolean is set to true, the images are masked to only contain values from the desired HSV range derived in the color picking process (section [3.2]). Otherwise no colormask is applied. Assuming this colormask is accurately enough determined, it leaves mostly pixels containing only the hold. Then, the frames are handed to the comparison method, alongside the bounding box of the currently investigated hold. The two pictures are converted to greyscale to speed up the computation, as commonly done in image computation tasks. The mean structural similarity index (SSIM index; Z. Wang et al., 2004a) is computed, as it represents a robust and fast measure for the differences between two images. The SSIM takes luminance, contrast and structure into account to create a measure called the SSIM index. These three values are derived from the respective means and variances and the covariance of two compared signals (image patches). They are then corrected with constants that take the dynamic range of pixel values into account and combined to create the SSIM index as a measure that considers various types of distortions (Z. Wang et al., 2004a).

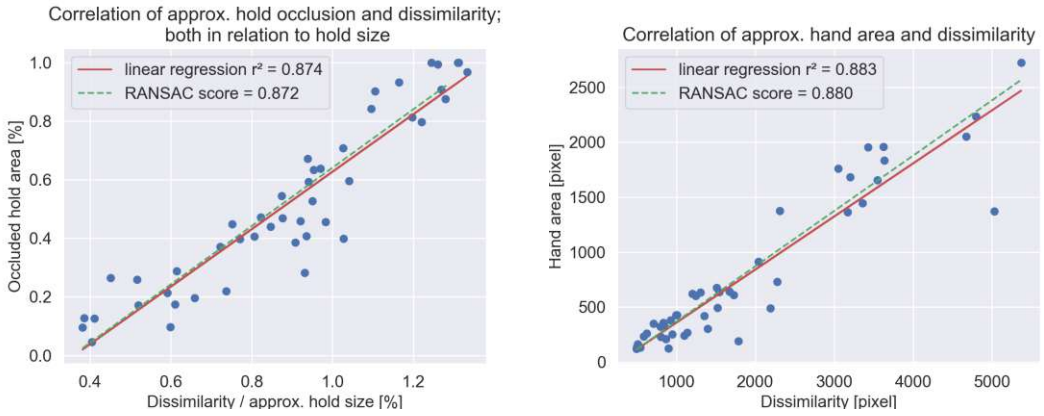
For the calculation of the SSIM index, the structural similarity method from scikit-image package is used, resulting in a similarity score (ranging from 0 to 1), the mean SSIM index, and an image containing the differences (e.g Figure 3.11). Thresholding and contour extraction methods are applied to this difference image, to gain a better visual representation, as seen in Figure 3.11. The difference images are written to .jpg files, categorized by their frame number and the number of the corresponding hold. Additionally, the similarity score is converted to pixels by multiplying it with the area of the investigated bounding box, for later thresholding. In summary, the comparison method returns a difference image, a similarity score in percentage and the similarity score times the corresponding bounding box area. The similarity score in relation to the bounding box is subtracted from the bounding box area to gain another value used for thresholding methods. This value will be referred to as dissimilarity. Although the similarity score is not an areal metric, empirical testing with the score converted to pixels revealed that using this converted score improves the determination whether a hold is "reached". This is additionally validated when looking at Figure 3.10, as a linear dependency between the second and third thresholding values and SSIM in pixels is observed.

Three threshold values are used to compare the outputs of the structural similarity method and the dissimilarity to:

1. The first value represents the mean structural similarity (Z. Wang et al., 2004a) below which the hold is considered "reached".

2. A value representing the minimal occlusion of the hold in percent, over which we assume the hold was "gripped". It is compared to the dissimilarity divided by the approximate hold size (Figure 3.10a).

3. The last value represents an empirically derived approximation of the number of pixels of the climber's hand in the video, the hand area. A pixel value was chosen instead of a percentage threshold, because the size of the climber bounding box from the YOLOv3 detection (section [3.3]) varies strongly even within one video. If the dissimilarity is bigger than this pixel value, the hold is considered "gripped" (Figure 3.10b).



(a) *Correlative validation of the occlusion threshold*

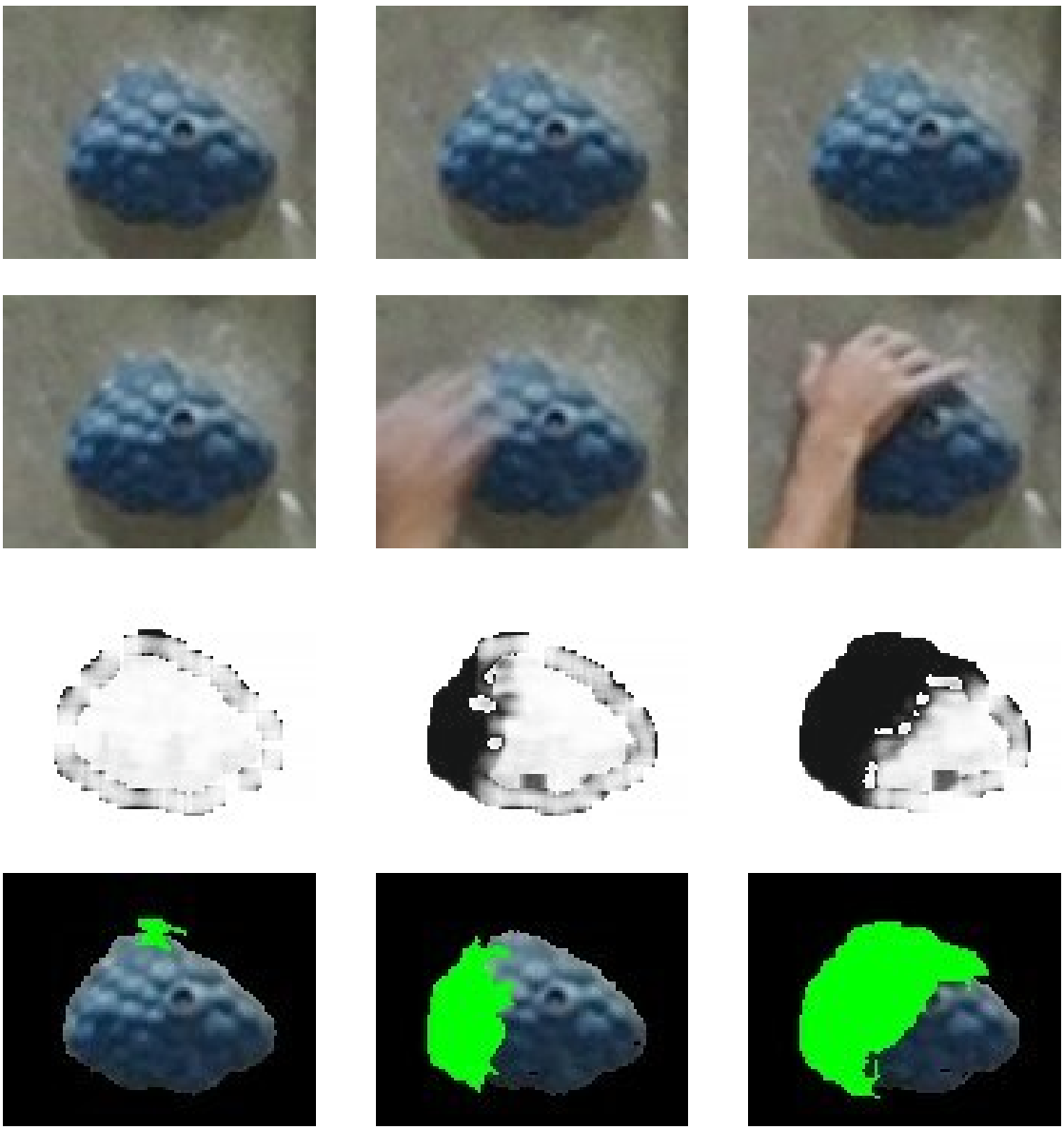
(b) *Correlative validation of the hand area threshold*

Figure 3.10 *Correlation plots of the threshold values on the y-axis and the respective metric they are compared to on the x-axis; each dot represents one analysed hold.*

For the validation process 46 holds from four videos, two 4k and two 1080p, were manually investigated. For each of the holds the metrics used for thresholding were calculated and the linear regression (Weisberg, 2005) and linear RANSAC (Fischler and Bolles, 1981) r^2 scores were computed to validate their use. The good linear correlation ($r^2 > 0.8$) supports the choice of these values for thresholding.

In Figure 3.11, the process of a hold getting gripped is seen. If either one of the three thresholds is exceeded the hold is "reached". The differences in Figure 3.11a could be considered optical noise, while in Figure 3.11b an occlusion is obvious. Figure 3.11c represents the first instance of this particular hold in which it is considered "gripped".

If none of these thresholds is reached, the next frame, which meets the condition of the hold being inside the climbers bounding box, is checked. It was tested to check every 0.33 seconds or, twice as often, every 0.167 seconds. These durations represent a significant reduction of handled frames(by 95% or 90% for 1080p videos and 90% or 80% for 4k videos respectively) but are still short enough to not miss important events. The values were handed to the script as frame values, adapted to the videos respective frame-rate. Longer pauses in-between checks were deemed as pointless, considering early tests and the fast paced situations that can arise in a climbing competition, while more frequent checks would not reduce the computational expense significantly. If the thresholds are not reached in any of these frames, the hold is considered not "reached" or "gripped". As soon as a hold is considered "gripped" or "not gripped", the next hold is dealt with until all marked holds (cf. [3.2]) are handled.



(a) *Frames 3035/3095; small visible image differences; clearly optical noise as there is no overlap apparent.*

(b) *Frames 3040/3100; bigger visible difference; definitive overlap, from hand moving over the hold.*

(c) *Frames 3045/3105; difference where the thresholds are reached and this particular hold is declared "gripped".*

Figure 3.11 *Different images showing the results from the image structural similarity comparison; first a) and second row b): The images are taken from a 4k video with a 60 frame/2 second comparison-baseline delay; third row c): the raw difference images range from white to black, where white represents no difference; last row: the differences are highlighted in a bright green color as an overlay over the masked images*

3.5 Score application

The score for a hold is determined simultaneously with the delineation of its bounding box. These scores have to be known beforehand. As stated in IFSC (2021) the scores for a competition are to be prepared in a "Topo" by the chief routesetter. The process of applying the score itself is trivial. As soon as a hold is declared "gripped", the climber's

score is set to the hold's corresponding score. This final score is shown as soon as the processing of the video is finished.

A method for life-video-feed use of automatic scoring system would be, to subject the score to a manual review. This could be done in the form of a dialog window, that appears when a hold is recognized as "gripped" by the system.

3.6 System validation

Each hold was manually inspected whether the system worked properly. During this inspection the holds were assigned to one of three categories:

1. holds that were declared "reached" or "not reached" correctly in the first possible instance: n_{corr}
2. holds where the system failed entirely
3. holds that were declared "reached" due to the lack of differentiation of hands and other body parts or late due to failing climber detection: n_{int}

The holds in these categories were then attributed to the *performance* (P) or the *integrity* (I) of the system in the particular video. The *performance* represents the real world performance where one can rely on the system to agree with the ground truth. Each hold assigned to the first category was correctly declared as "reached" or "not reached" and therefore attributed to the *performance*. This number of holds, where the system was performing well, was divided by the number of holds in the respective climbing route (equation 3.1) to make them comparable between videos and routes. The resulting percentage value is referred to as *performance* (P).

$$P = \frac{n_{corr}}{n_{route}} * 100 \tag{3.1}$$

The *integrity* value represents the percentage where the system works as intended but does not necessarily produce results that could be used in a real world application. For the *integrity* the holds contributing to the *performance* were expanded by adding holds from the third category. This includes holds where a body part other than the hand was responsible for the declaration as "reached" and delayed "reached" declaration caused by failing YOLOv3 detection. In this context, a YOLOv3 detection error was only counted as system failure, if a hold was entirely skipped. The sum of holds from category one and three represents the script working properly within its limitations. This value was divided by the number of holds in the respective climbing route (equation 3.2 to get the desired percentage value, which is referred to as *integrity* (I).

$$I = \frac{n_{corr} + n_{int}}{n_{route}} * 100 \tag{3.2}$$

If neither was the case and a *false positives* or *false negatives* (FP and FN) was observed, the system was also failing. A manual differentiation between *performance* and *integrity* is made, because of the system's inability to differentiate between the two automatically. For the purpose of comparing the videos, the *performance* and *integrity* originating from

the best performing parameter set for each video were taken. These parameter sets can be observed in table 4.1, where the videos are listed with their resolution, the set of delay and frame reduction values that worked best in their case, and their respective *performance* and *integrity* values. The percentage threshold values that were chosen for the videos can be found in table 3.1. The similarity threshold of 0.6 was the same for all videos and the overlap threshold was 0.9 for all videos except video 11 were 0.8 was chosen after reviewing the results and finding an unusual amount of *fales negatives*. The hand size in pixels was chosen for each video individually.

3.7 Process summary

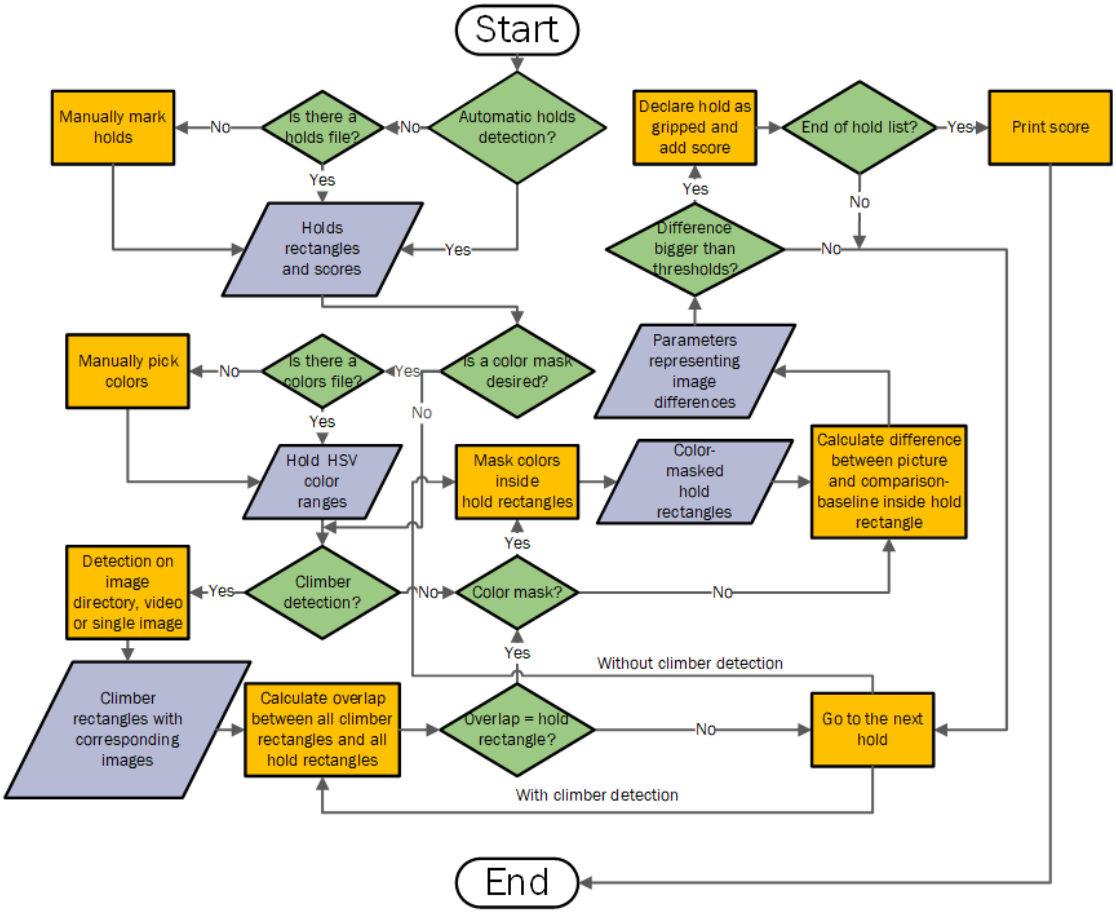


Figure 3.12 *Illustration of the processing steps and decisions of the automated scoring system; decisions are depicted with green rhomboid shapes, processes with orange rectangles and data with lavender colored parallelograms.*

This section describes the data- and workflow of the automated scoring system (Figure 3.12). As some of the data processing was already broached in the previous sections, the focus will be only on important steps, the interactions between the different parts and the workflow as a whole.

At first, some input data is needed, which is, in the case of this thesis, a prerecorded video split into single frames (section [2.1.3]). Alternatively, some sort of camera system

could be connected to the PC the Python script is run on to provide live-video data. The YOLO detection input parameters are left on their default settings for the most part, while the other parameters are set according to the resolution and frame-rate of the input video, as well as to the object distance. In Table 3.1 the input parameters are described in respect of their function, data type and/or value range.

Table 3.1 *Examplary input parameters of the automated scoring system.*

Input Parameters	**Functions of parameters**	**Data Type/Content/Value ranges**
YOLO detection parameters	Parameters for the YOLO detection, are set once and used for all videos;Includes paths to classes and weights, values for image resizing and number of classes	Path strings Booleans Integers
Directory paths	The paths to the input and output directories	OS friendly path variables as string
CSV paths	Path to the hold and colour range CSV files or where they should be saved	OS friendly path variables as string
Worklfow control booleans	Declare whether you want to use the automatic detection methods, colormasking or a source different than an image directory	Booleans
Comparison-baseline delay	Number of frames that the baseline image is delayed in comparison to the current investigated frame; Has to be adjusted to the video framerate	Frame values equalling [1, ... ,4] seconds
Frame-reduction	Reduction of used frames by the set factor; Has to be adjusted to the video framerate	Frame values equalling [0.167, ... ,0.33] seconds
Hand Size Pixels	Approximate hand size in pixel; derived from empirical testing; Has to be adjusted to video resolution and object climbing wall distance	[500, ..., 4000] pixels
Similarity threshold	Similarity percent under which a hold is declared "gripped"; an empirical derived value between 0 and 1	(0.6) equalling to 60%
Overlap threshold	Percentage over which a hold is declared "gripped"; Compared to value representing the differences of current frame to the comparison-baseline image in the not colormasked area; also derived from experiments; expressed in a value between 0 and 1	(0.8, 0.9); equalling to 80% or 90%

After the input parameters are defined, the process is started:

1. At the start of the process, some preparatory checks are conducted, mainly comprised of seeing if the decisive booleans are either set to true or false.

2. The need for an automatic holds detection and the existence of a holds.csv file is checked, leading to either said detection, a manual delineation of holds or a read out of the file as described in section [3.2]. This step provides the bounding box coordinates in pixels and scores for each hold, stored as a list and a CSV file. If the holds were already handed over as a CSV file, they are just read and written to a list.

3. Next, a check if a color mask should be applied, is conducted. If the corresponding input parameter is true, the existence of a color.csv file is verified, otherwise the color picker is used (see section [3.2.2]). This step yields said color ranges in a CSV file and stored as a list.

4. The climber detection is executed. The pre-trained weights are used to detect all objects, that are present in the set by Redmon and Farhadi (2018). The detection infrastructure provided by *TensorFlow* (2021) is modified in a way, that only instances of detected persons are displayed and labeled as climbers. This detection is applied to the video frames in the input directory. Only one of the detected climbers is saved for each picture, namely the one closest to the upper left corner of their bounding box. These are stored in a list, while each of the frames is renamed

and saved in a new folder, to match the index of its respective climber bounding box inside this list.

5. The next step is to check whether the holds are "reached". Two loops are used for this task, the outer iterates through the hold list in the same sequence they were marked in by the routesetter and the inner through the list of detected climbers. To avoid checking every hold for an overlap in all videos frames, the bounding boxes are compared. The difference calculation is executed by computing the mean structural similarity between the comparison-baseline and the current frame using OpenCV2 and Scikit-image.

6. The similarity score values, both percent and pixels, are compared to the similarity threshold, the overlap threshold and the hand size pixel input parameters. This comparison is executed as described in section [3.4], resulting in a declaration, whether a hold is "reached" at the moment of the given frame or not.

7. Once a hold is deemed "reached" or all video frames processed, the next hold is investigated. As a hold is "reached", its score value is set as the climbers current score and the difference image is written to a .jpg file with a "gripped_" prefix for a manual review.

8. As the end of the hold list is reached, the score is shown.

To asses the performance of the automated scoring system, a manual review is conducted, comparing the difference images, where the hold is deemed "reached", with the unedited video-frames (Figure 3.11). The user's interpretation of the video is presumed as ground-truth for this purpose.

Chapter 4

Results and Dicussion

In this chapter the results of the thesis are presented. Afterwards, the results and the obstacles that came up during development are discussed. The ways the obstacles were overcome will be referred to, otherwise their origin will be investigated and possible solutions will be discussed.

4.1 Results

The results from the data processing consist of the manually reviewed outputs of the automated scoring system tested with the 14 videos (seven in 4k and seven in 1080p resolution) from the third data acquisition session. Different input parameter sets were tested and compared in regards of their *performance* and *integrity*. The results from the respectively best parameter sets for each video (Table 3.1) are shown in Figures 4.1 and 4.2. In Figure 4.1 it can be observed, that the *performance* of correctly declaring holds "reached" for 4k videos is overall better. The mean *performance* of 4k is around 48% while the mean *performance* of the 1080p videos is around 40%.

When looking at Figure 4.2, it illustrates that the *integrity* is generally much higher than the *performance*. Additionally, the difference between the means is smaller by about 2 percentage points (Table 4.1) in comparison to the *performance*. The *integrity* for 4k videos is again higher than for 1080p videos. Furthermore, a smaller *integrity* variation between the 4k videos can be observed. The video pairs that show the same route respectively, videos 1/2, 3/4 and 13/14, generally favor 4k for both *integrity* and *performance*.

As seen in figures 4.1 and 4.2, the *performance* and *integrity* do not necessarily relate to each other. Video 10 is both the best performing video with a correct hold declaration of approximately 73% overall and has an *integrity* of 100%. In contrast, Video 2 is performing rather poorly with only 24% but has the same *integrity* of 100% (table 4.1). For the other videos, higher *integrity* generally yields higher *performance*.

Nonetheless, the generally high *integrity* indicates potential for improvement by adding methods to differentiate between the climber's hands and other body parts. As these additions would have been beyond the scope of this thesis, the proposed improvements will be discussed in section [5].

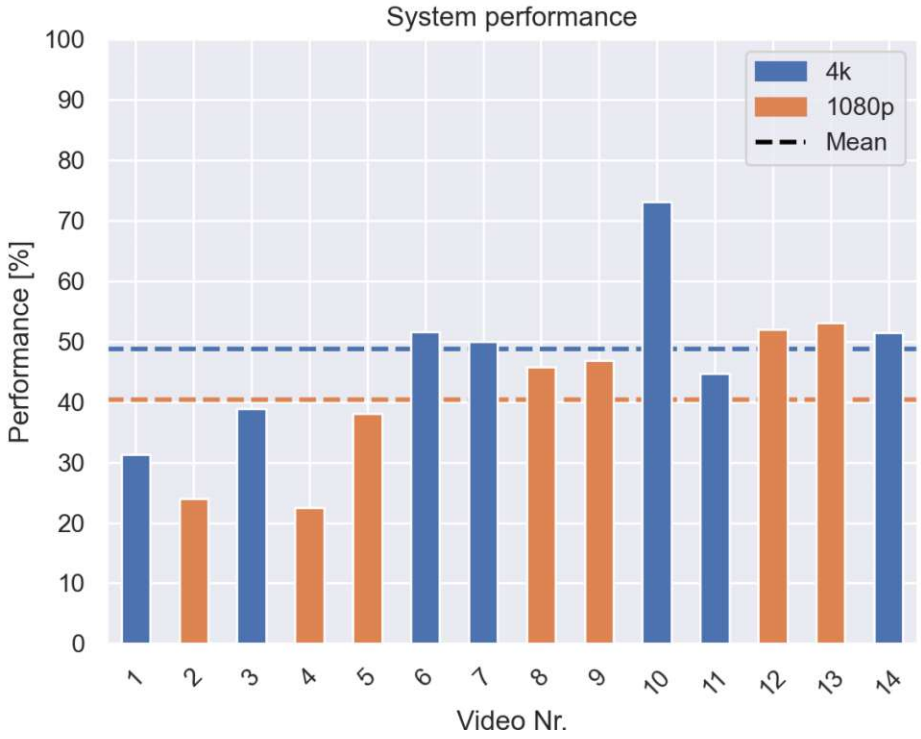


Figure 4.1 *System performance for the 14 different videos; the mean values for each resolution are displayed as a dashed line in the respective color*

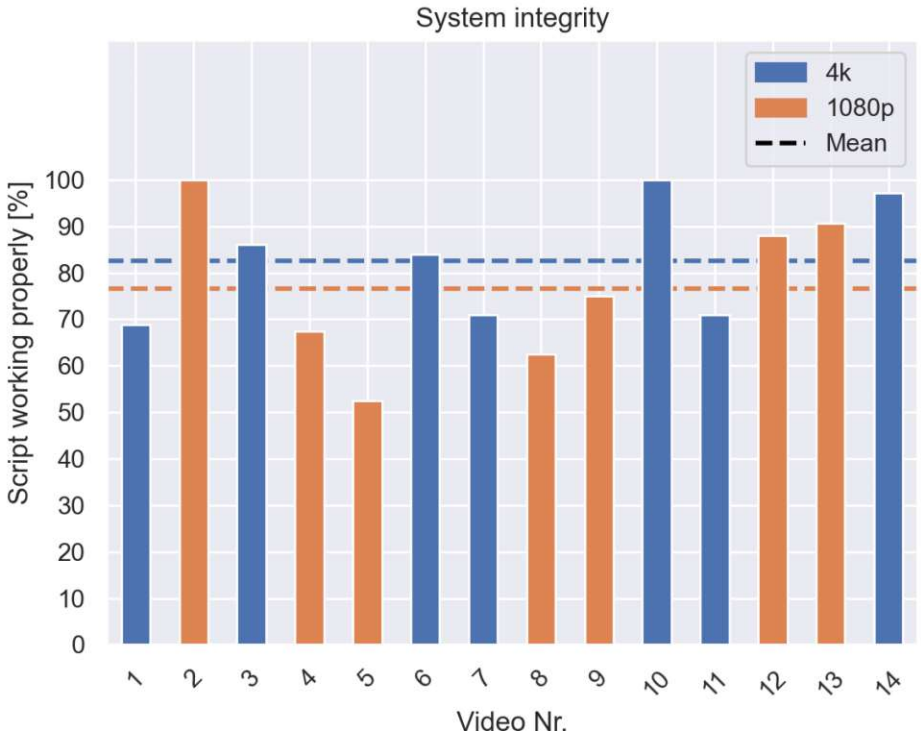


Figure 4.2 *System integrity for the 14 different videos; the mean values for each resolution are displayed as a dashed line in the respective color*

In Figure 4.3 the distribution of *comparison-baseline delay* and *frame-reduction* parameters is visualized by how often they were the best parameter set for the two resolutions and for all videos. If a video performed equally well or was of equal *integrity* for two parameter sets, the video is counted for both sets. The set of a 2 second baseline delay and 0.167 second (frame-reduction) is clearly the best for system *performance* and *integrity*. In contrast, the set with a 4 second baseline delay and 0.167 second (frame-reduction) was in no case the best. For system *integrity*, a 2 second baseline delay and 0.33 second (frame-reduction) also yielded good results. Between the two resolutions the parameter sets were very similarly distributed in both *integrity* and *performance*, although the 4k resolution seemed to handle a longer *comparison-baseline delay* better.

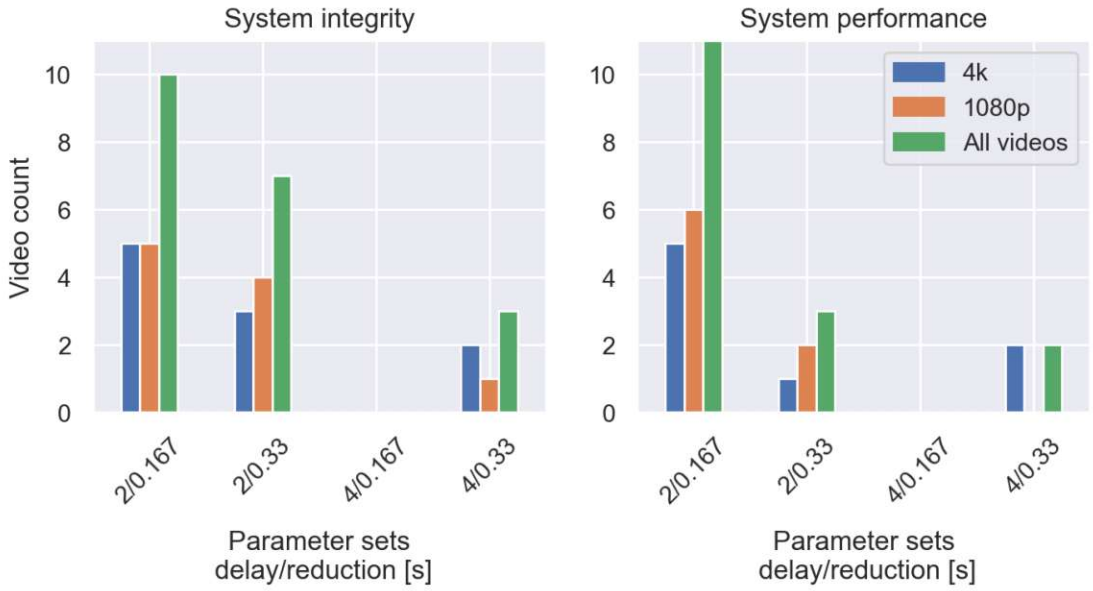


Figure 4.3 *Distribution of the parameters for the best integrity and performance respectively; how often was each parameter set the best set for the different video types; the sum of best parameters exceeds 7/14 as in some cases two parameter sets performed equally well; while more parameter sets were tested, these were not included as they were never close in regards of performance and integrity to the four depicted sets.*

The whole system runs 1h30m to 18h18m for 1080p 60fps videos and 1h30m to 22h17m for 4k 30fps videos, depending on the length of the video and the used (frame-reduction). The processing time differences between parameter sets were marginal, therefore for illustration purposes the mean was computed for each video (Figure 4.4). The current computational bottleneck can be attributed to the GPUs VRAM for the entire process. Approximately 25% of the processing times consisted of the YOLOv3 climber detection and the other 75% of the process of declaring the holds "reached".

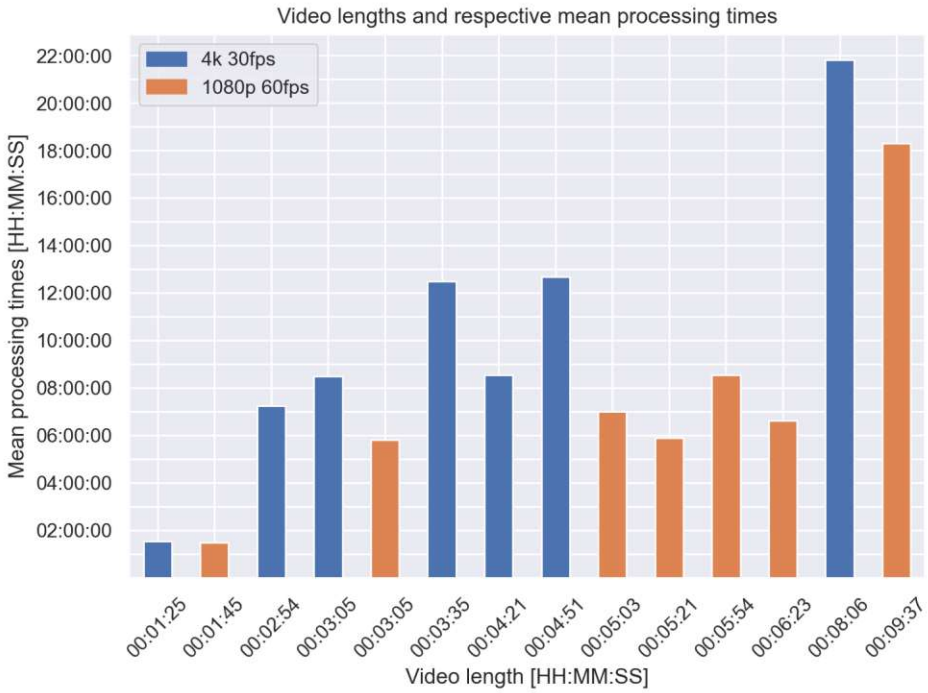


Figure 4.4 *Length of the videos and their respective mean processing times; It can be observed that the processing times for 4k videos are always higher for similar video lengths.*

Table 4.1 *Overview of the best parameter sets for either performance or the integrity; equally performing sets are connected with an "and".*

Video Nr.	**Resolution**	**Delay/Reduction [s]**	**Performance [%]**	**System integrity [%]**
1	4k	2/0.167 and 2/0.33	31.25	68.75
2	1080p	2/0.167 and 2/0.33	24.00	100
3	4k	4/0.33	38.89	86.1
4	1080p	2/0.33	22.50	67.5
5	1080p	2/0.167	38.10	52.4
6	4k	2/0.167	51.61	83.9
7	4k	4/0.33 4/0.33 and 2/0.167	50.00 -	- 71
8	1080p	2/0.167 4/0.33 and 2/0.167	45.83 -	- 62.5
9	1080p	2/0.167 4/0.33	46.88 -	- 75
10	4k	2/0.167	73.08	100
11	4k	2/0.167 2/0,33	44.74 -	- 71
12	1080p	2/0.167 2/0.167 and 2/0.33	52.00 -	- 88
13	1080p	2/0.167 2/0.167 and 2/0.33	53.13 -	- 90.6
14	4k	2/0.167 2/0.167 and 2/0.33	51.43 -	- 97.1
mean	4k	-	48.71	82.6
mean	1080p	-	40.35	76.6

4.2 Discussion

The higher resolution yielding better results could be attributed to a number of reasons:

1. more accurate color masks (Figure 4.5a)
2. easier recognition of holds and bounding box drawing as seen in Figure 4.5b
3. less influence of optical noise; fewer pixels affected in reference to the entire hold
4. more accurate SSIM computation, as differences can be determined more precisely with better color masks

For the climber detection the pictures are resized to the same size for the YOLOv3 detection for both resolutions, as the used PCs could not handle a higher sample size. The frame rate is also unlikely to have a large impact on the *performance*, as the baseline delay and *(*frame-reduction) parameters were scaled in order to result in the same time values for both resolutions.

(a) ***(b)***

Figure 4.5 *a) the same colormask applied to a 4k (top) and 1080p (bottom) bounding box showing the same hold; the results for this purely demonstrative colormask are better for the 4k image; b) images showing two examplary holds in 4k (top) and 1080p (bottom); while the 4k holds are similar colored, their extent is easily distinguishable from the background; in the 1080p images the holds almost blend into the background and their extent cannot be easily determined.*

One could assume that an even higher resolution could further improve the results since 4k performed better in average than 1080p. On the one hand this might improve the *performance* and *integrity* and solve some problems such as low pixel counts in small hold bounding boxes. Additionally, a higher resolution could reduce the system's sensitivity

to optical noise. On the other hand higher resolutions would drastically increase the already long processing times (Figure 4.4) and high PC system requirements. A way to introduce higher resolution would be in conjunction with an image pyramid system, which would help to reduce the computational effort of the high resolution, but preserve the detail for the image difference calculation. As the initial idea for the automated scoring system included real time capabilities, it could be considered to either introduce said image pyramid system, drastically increase the computational capabilities of the used PC, namely more or better GPUs, or lower the resolution of the videos. With the current system, a lower resolution is not a viable option, as the performance for the 1080p videos is not satisfying.

Besides the resolution, there are some other limitations in the current version of the system that have an impact on both *performance* and *integrity*. These limitations and the problems with the resolution can lead to the system failing for some holds entirely due to the following causes:

1. low pixel count in 1080p hold bounding boxes leading to higher susceptibility for errors (Figure 4.5)
2. optical noise causing one of the thresholds to be exceeded
3. changing ambient light between comparison-baseline and investigated frame
4. changing object distance throughout one video
5. holds gripped in a way, that the thresholds are not exceeded (Figure 4.6)
6. similar hold and background color causing problems in the colormasking process (Figures 4.7 and 4.8)
7. YOLOv3 detection failing entirely causing the frames where the hold is "reached" to be skipped

The most common error is a *false positive* that can be contributed to optical noise or low resolution (about 84% of all errors). The origin of the optical noise could not be determined clearly, but the most likely cause are small changes in lighting that are not visible to the naked eye. This problem cannot be solved easily, but could be mitigated by reducing the system's sensitivity by raising the thresholds for a hold to be considered "gripped". The problem hereby is, that this would cause more *false negatives*.

While the final thresholds for each video (Table 3.1) try to balance these two error sources, the pure empirical nature of these parameters leads to the assumption that more empirical testing or even the introduction of more and different threshold values could improve the system e.g. the incorporation of the bounding box size under the assumption of an equally large padding between the hold and the bounding box borders for all holds or the introduction of an additional difference measure to use in conjunction with the SSIM.

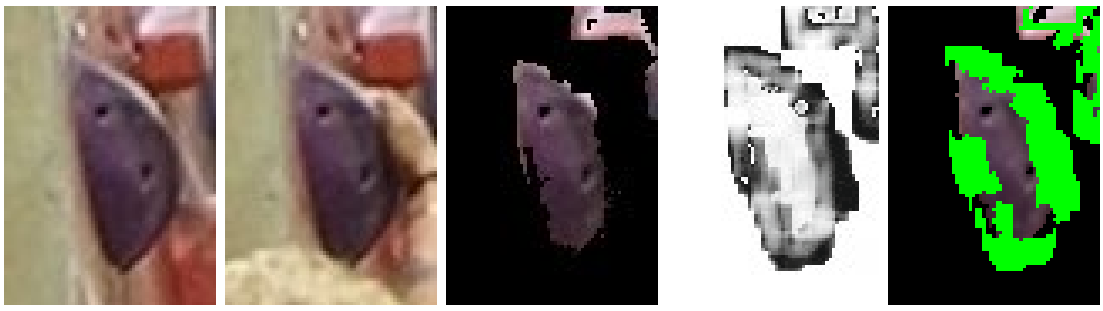


Figure 4.6 *From left to right: The "not gripped" hold; the "gripped" hold; the "gripped" hold colormasked; the raw SSIM image; the masked image with the differences highlighted; this hold is gripped in a way, that the thresholds are not exceeded and the hold is therefore not registered as "reached".*

Another potential problem that emerges with the experimental setup and the methods (section [3.1]) is the changing ambient light. Ideally, a completely controlled setting in the regards of light would have been desirable. With the recordings conducted in an climbing facility during normal opening hours, there was no way to influence the lighting. Additionally the facility has built-in skylights which also contribute to changing, not controllable lighting conditions. Although the lighting could not be influenced, some precautions were taken. The *comparison-baseline delay* for example was chosen to be relatively short to minimize the influence of changing lighting conditions on the SSIM computation. Unfortunately, its impact on the results could not be differentiated from the optical noise's.

Figure 4.7 *The hold color similar to climber's skin color causing problems with the colormasking; left: a well defined colormask; right: the climber's hand is not masked due to its color*

The changing object distance between the camera and the climber should also be taken into consideration. The system handles this in a very simple way by using an approximate of the average hand size throughout the video. More refined solutions such as changing hand sizes could be taken into consideration for increased system performance. Additionally the hand size should be adapted to each climber and climbing situation individually. One way to tackle this problem could be the integration of a hand detection which would enable an on the fly adaption of the hand size. This hand detection would also help to solve many other problems. For example, other body parts occluding the hold would not pose as large of a problem if the position of the climber's hands could be determined.

This could also help with holds that are gripped in ways, that the defined thresholds are not reached (e.g. with just a finger or two, or just on the edge of the hold as seen in Figure 4.6).

Figure 4.8 *A poorly determined colormask due to similar Saturation and Value values between background and hold color; the left and right image are only a few frames apart but show noticeably different colormasking*

The color-masking (section [3.2.2]) is also a crucial part of the system that has some downsides. While it helps immensely with the thresholding process, it can cause severe problems if not tuned and applied appropriately. Furthermore, if the wall and the holds have a very similar SV (Saturation and Value) range for example, not using a color mask at all can be beneficial (Figure 4.8). Similarly colored holds and walls can cause the colormasking to mask different areas between the comparison-baseline and the investigated hold (Figure 4.5b) leading to a decrease of the SSIM, inducing a *false positives* as thresholds are exceeded. Another problem with the colour mask can occur if the holds have similar HSV values as the climber's skin. This will lead to areas inside the bounding box not being masked properly, as soon as an exposed part of the climber enters said bounding box (Figure 4.7). Similar problems occur if the holds have a color similar to the shadows that they cast on the climbing wall (e.g. black and dark grey holds).

As stated in section [3.2] an automatic hold detection was originally planned. It quickly became obvious that, with the given resources, a manual approach would be the appropriate solution. The automatic detection of climbing holds is a very hard task given the large variety of shapes and sizes. This makes it very difficult to create e.g. HAAR-Cascades, that detect the different holds sufficiently without creating *false positives*. In addition, sometimes common shapes such as faces, cars or even small mammoths are used as templates for climbing holds. These factors ultimately led to the decision to take a manual approach for delineating holds. Additionally, the referee has to assign scores to the holds anyways, so a manual hold delineation could be integrated into the workflow easily. In this specific case, the lack of data and computational capabilities supported the decision to switch strategies. These factors resulted in a HAAR-Cascade that worked reasonably well for detecting holds, but had such a low efficiency that it rendered the used PC unresponsive. As also already mentioned in section [3.2] an approach with YOLOv3 was tested, even though YOLO generally struggles with detecting many small

objects close to each other (Redmon et al., 2016b), a situation that occurs regularly on climbing walls. The results from this tests were not usable as expected.

The use of a pre-trained YOLOv3 detector for the climber detection (section 3.3) was another decision that arose in the course of initial tests. Multiple attempts to create a custom detector were made, but none was performing sufficiently. This failure can be attributed to the lack of training data and the lack of computing capabilities for sensible neural network training. Separate training attempts took up to 12 days even though two relatively capable PCs (section [3.1]) were used in parallel. Additionally, the approximately 1400 images proved to be insufficient as at least 2000 images are recommended for this kind of training (Bochkovskiy et al., 2020). Fortunately, the YOLOv3 on COCO pre-trained set of weights by Redmon and Farhadi (2018) was able to detect the climbers on the wall as persons relatively reliably.

Another point that has to be discussed is the choice of metric to asses the image differences between the comparison-baseline and the image that is currently investigated by the system. For this task, the structured simililarity (SSIM) index was chosen due to its wide use and its claim to represent perceived differences very well. While this index was initially developed to assess the quality of image compression (Z. Wang et al., 2004a), it can be utilized to calculate differences between any two images. Some sources claim, that the SSIM is in fact not a mathematical metric and cannot determine image similarities correctly and the Pearson correlation coefficient is proposed instead(Starovoitov et al., 2020), but even there it is stated, that the SSIM works very well for images that are visually very close to each other, which is certainly the case in this thesis. Nonetheless, this method is used widely (e.g. Sara et al., 2019; Rehman and Z. Wang, 2011; Rehman and Z. Wang, 2012; S. Wang et al., 2012; Z. Wang et al., 2004b; Ou et al., 2011; Liu et al., 2022; Setiadi, 2021). Furthermore, the SSIM worked well for determining image differences during tests (cf. Figure 3.10).

A general point for discussion is if the videos of the two resolutions would even be comparable for assessing the *performance* and *integrity* of the system as they cover different routes with different climbers. While a direct comparison of individual videos would not make much sense in this context, a general similarity in the experimental setups was taken into consideration. Based on this, it is assumed that a statistical comparison of the two resolutions as a whole makes sense (Figures 4.1 and 4.2) and was conducted as such in this thesis.

Chapter 5

Conclusion and Outlook

Following the concept presented in this thesis, a system was built that uses pre-recorded videos showing the climber and the wall from behind the climber. After initial testing with HAAR-Cascade for automatic hold detection failed, it was opted to mark the holds manually by defining bounding boxes and further delineating them by HSV color masking in a simple UI. For the climber detection and tracking, a pretrained YOLOv3 detector proved to be the best available option. Declaring a hold as "reached" was done by comparing the hold bounding box area of the current frame to the same area a number of frames before it and calculating the SSIM between the areas (Figure 3.11). From the SSIM, two values are calculated (Figures 3.10a and 3.10b) and checked as to whether the values and the SSIM itself surpass three empirically derived thresholds. If any of those thresholds are surpassed, the hold is declared as "reached". The system was tested on seven videos in 4k and seven videos in 1080p resolution. As ground truth for the evaluation of the results, a manual review of every hold was conducted to check whether the hold is declared correctly.

The overall mean *integrity* of the results is 82.6% for 4k and 76.6% for 1080p (Figure 4.2) showing great potential for improvement of the *performance*, which is at 48.7% and 40.4% correctly declared holds, respectively (Figure 4.1). A simple way to increase the *performance* could be using higher resolutions, which would in return increase computational complexity. This increased complexity would arise the need for more capable processing resources which would as a consequence raise the system's cost. As a low cost system is the aim of this thesis, this would contradict the intended purpose. Other potential improvements include further experiments with different thresholds, *comparison-baseline delays* and *(*frame-reduction) parameters. Additionally, as mentioned in section [4.2], the system would greatly benefit from an automated hand detection to improve the robustness of the "reached" declaration process and to improve the handling of the different climber hand sizes throughout a video. This could also increase the system's speed, as the search area in both spatial and time domain for potential "reached" holds could be further narrowed down in comparison to the simple climber detection. Other than that, the code that underlies the system, is not optimized for computational speed and leaves room for improvement.

The total estimated cost of the equipment used for the final testing and processing is about 3280€. This could be greatly reduced as the processing equipment's cost is estimated at 1080€ and 4k cameras available at as low prices as 200€ or less. The used filming equipment adds up to about 2200€ and was utilized because it was easily available for this thesis. In comparison to other systems that use more sophisticated capturing equipment (section [1.3]) the proposed system can be considered relatively low-cost.

All in all the, intended goal of the thesis was reached as all parts of the original concept were implemented and assembled into a working system. While many new developments and publications are made in the wake of the increasing popularity of climbing sports in general (Olhorst, 2019), a surprisingly low amount of research was made on the topic of automated scoring systems. This is were the proposed ideas and the ultimately developed system serves as a proof of concept for building low cost automated scoring systems for lead climbing competitions and to show that it is possible to create a working scoring system with only one camera perspective and state of the art technology. Even though the performance doesn't seem great on paper, the envisioned system was implemented in a way, that it is sufficient for follow-up analysis of lead climbing competitions and to serve as a basis for future developments in this field of research.

List of Figures

List of Tables

Bibliography

Ahmadi, A. (2017). *Cascade Trainer GUI - Amin*. URL: https://amin-ahmadi.com/cascade-trainer-gui/ (visited on 12/20/2021).

Aladdin, R. and P. G. Kry (2012). "Static pose reconstruction with an instrumented bouldering wall". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. New York, New York, USA: Association for Computing Machinery, pp. 177–184. DOI: 10.1145/2407336.2407369. URL: http://dl.acm.org/citation.cfm?doid=2407336.2407369.

Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao (Apr. 2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: arXiv: 2004.10934. URL: http://arxiv.org/abs/2004.10934.

Cordier, P., M. M. France, P. Bolon, and J. Pailhous (Sept. 1994). "Thermodynamic study of motor behaviour optimization". In: *Acta Biotheoretica* 42.2-3, pp. 187–201. DOI: 10.1007/BF00709490. URL: https://link.springer.com/article/10.1007/BF00709490.

Dovgalecs, V., J. Boulanger, D. Orth, R. Hérault, J. F. Coeurjolly, K. Davids, and L. Seifert (Oct. 2014). "Movement phase detection in climbing*". In: *Sports Technology* 7.3-4, pp. 174–182. DOI: 10.1080/19346182.2015.1064128. URL: https://www.tandfonline.com/doi/abs/10.1080/19346182.2015.1064128.

Ebert, A., K. Schmid, C. Marouane, and C. Linnhoff-Popien (Oct. 2018). "Automated Recognition and Difficulty Assessment of Boulder Routes". In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Vol. 225. Springer Verlag, pp. 62–68. DOI: 10.1007/978-3-319-76213-5_9. URL: https://doi.org/10.1007/978-3-319-76213-5_9.

Efstratiou, P. (2021). *SKELETON TRACKING FOR SPORTS USING LiDAR DEPTH CAMERA KTH Thesis Report Panagiotis Efstratiou KTH ROYAL INSTITUTE OF TECHNOLOGY ENGINEERING SCIENCES IN CHEMISTRY, BIOTECHNOLOGY AND HEALTH*. Tech. rep. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-297536.

Felzenszwalb, P. F., R. B. Girshick, D. McAllester, and D. Ramanan (2010). "Object detection with discriminatively trained part-based models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9, pp. 1627–1645. DOI: 10.1109/TPAMI.2009.167.

Fischler, M. A. and R. C. Bolles (June 1981). "Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* 24.6, pp. 381–395. DOI: 10.1145/358669.358692. URL: https://dl.acm.org/doi/10.1145/358669.358692.

Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)*. Tech. rep. arXiv: 1311.2524v5. URL: http://www.cs.berkeley.edu/%CB%9Crbg/rcnn..

Haar, A. (Sept. 1910). "Zur Theorie der orthogonalen Funktionensysteme - Erste Mitteilung". In: *Mathematische Annalen* 69.3, pp. 331–371. DOI: 10.1007/BF01456326. URL: https://link.springer.com/article/10.1007/BF01456326.

IFSC (2021). “2021 Rules”. In: May, pp. 1–91. URL: https://cdn.ifsc-climbing.org/images/World_Competitions/2021_IFSC_Rules_v176.pdf.

Iguma, H., A. Kawamura, and R. Kurazume (Jan. 2020). “A New 3D Motion and Force Measurement System for Sport Climbing”. In: *Proceedings of the 2020 IEEE/SICE International Symposium on System Integration, SII 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 1002–1007. DOI: 10.1109/SII46433.2020.9026213.

IROZHLAS (2019). *Adam Ondra hung with sensors.* URL: https://www.irozhlas.cz/sport/ostatni-sporty/czech-climber-adam-ondra-climbing-data-sensors_1809140930_jab (visited on 01/04/2022).

Kajastila, R. and P. Hämäläinen (Apr. 2014). “Augmented climbing: Interacting with projected graphics on a climbing wall”. In: *Conference on Human Factors in Computing Systems - Proceedings*. New York, NY, USA: Association for Computing Machinery, pp. 1279–1284. DOI: 10.1145/2559206.2581139. URL: https://dl.acm.org/doi/10.1145/2559206.2581139.

Kajastila, R., L. Holsti, and P. Hamalainen (May 2016). “The augmented Climbing wall: High-exertion proximity interaction on a wall-sized interactive surface”. In: *Conference on Human Factors in Computing Systems - Proceedings*. New York, NY, USA: Association for Computing Machinery, pp. 758–769. DOI: 10.1145/2858036.2858450. URL: https://dl.acm.org/doi/10.1145/2858036.2858450.

Kalyanaraman, A., J. Ranjan, and K. Whitehouse (Sept. 2015). “Automatic rock climbing route inference using wearables”. In: *UbiComp and ISWC 2015 - Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the Proceedings of the 2015 ACM International Symposium on Wearable Computers*. New York, New York, USA: Association for Computing Machinery, Inc, pp. 41–44. DOI: 10.1145/2800835.2800856. URL: http://dl.acm.org/citation.cfm?doid=2800835.2800856.

Kosmalla, F., F. Daiber, F. Wiehr, and A. Krüger (Oct. 2017). “Climbvis — Investigating in-situ visualizations for understanding climbing movements by demonstration”. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces, ISS 2017*. Vol. 17. New York, NY, USA: Association for Computing Machinery, Inc, pp. 270–279. DOI: 10.1145/3132272.3134119. URL: https://dl.acm.org/doi/10.1145/3132272.3134119.

Kosmalla, F., F. Wiehr, F. Daiber, A. Krüger, and M. Löchtefeld (May 2016). “ClimbAware - Investigating perception and acceptance of wearables in rock climbing”. In: *Conference on Human Factors in Computing Systems - Proceedings*. New York, NY, USA: Association for Computing Machinery, pp. 1097–1108. DOI: 10.1145/2858036.2858562. URL: https://dl.acm.org/doi/10.1145/2858036.2858562.

Lin, T. Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (May 2014). “Microsoft COCO: Common objects in context”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8693 LNCS. PART 5. Springer Verlag, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48. arXiv: 1405.0312. URL: https://arxiv.org/abs/1405.0312v3.

Liu, N., L. Wu, J. Wang, H. Wu, J. Gao, and D. Wang (2022). “Seismic Data Reconstruction via Wavelet-based Residual Deep Learning”. In: *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–1. DOI: 10.1109/TGRS.2022.3152984. URL: https://ieeexplore.ieee.org/document/9716927/.

Okabe, T., I. Sato, and Y. Sato (2004). “Spherical harmonics vs. haar wavelets: Basis for recovering illumination from cast shadows”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. DOI: 10.1109/cvpr.2004.1315013.

Olhorst, T. (2019). *Rock Climbing Grows in Popularity - The National Digest*. URL: https://thenationaldigest.com/rock-climbing-grows-in-popularity/ (visited on 03/11/2022).
Olympus (2021b). *Specifications E-M1 Mark III | OM-D | Olympus: cameras, audio and binoculars*. URL: https://asia.olympus-imaging.com/product/dslr/em1mk3/spec.html (visited on 01/06/2022).
— (2021a). *Specifications E-M5 Mark II | OM-D | Olympus: cameras, audio and binoculars*. URL: https://asia.olympus-imaging.com/product/dslr/em5mk2/spec.html (visited on 11/10/2021).
— (2021c). *Specifications M.ZUIKO DIGITAL ED 12-100mm F4.0 IS PRO | M.ZUIKO PRO | Olympus: cameras, audio and binoculars*. URL: https://asia.olympus-imaging.com/product/dslr/mlens/12-100_4ispro/spec.html (visited on 11/10/2021).
— (2021d). *Specifications M.ZUIKO DIGITAL ED 12-40mm F2.8 PRO | M.ZUIKO PRO | Olympus: cameras, audio and binoculars*. URL: https://asia.olympus-imaging.com/product/dslr/mlens/12-40_28pro/spec.html (visited on 01/06/2022).
OpenCV (2018a). *OpenCV: cv::ximgproc::StructuredEdgeDetection Class Reference*. URL: https://docs.opencv.org/3.4.2/d8/d54/classcv_1_1ximgproc_1_1StructuredEdgeDetection.html (visited on 02/03/2022).
— (2018b). *OpenCV: Image Processing (imgproc module)*. URL: https://docs.opencv.org/3.4.2/d7/da8/tutorial_table_of_content_imgproc.html (visited on 01/13/2022).
— (2021). *About - OpenCV*. URL: https://opencv.org/about/ (visited on 10/27/2021).
Ou, T. S., Y. H. Huang, and H. H. Chen (May 2011). "SSIM-based perceptual rate control for video coding". In: *IEEE Transactions on Circuits and Systems for Video Technology* 21.5, pp. 682–691. DOI: 10.1109/TCSVT.2011.2129890.
Pandurevic, D., A. Sutor, and K. Hochradel (Nov. 2019). "Methods for quantitative evaluation of force and technique in competitive sport climbing". In: *Journal of Physics: Conference Series*. Vol. 1379. 1. Institute of Physics Publishing, p. 012014. DOI: 10.1088/1742-6596/1379/1/012014. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1379/1/012014%20https://iopscience.iop.org/article/10.1088/1742-6596/1379/1/012014/meta.
Pandurevic, D., A. Sutor, and K. Hochradel (2020). "Introduction of a Measurement System for Quantitative Analysis of Force and Technique in Competitive Sport Climbing". In: *Proceedings of the 8th International Conference on Sport Sciences Research and Technology Support*. SCITEPRESS - Science and Technology Publications, pp. 173–177. DOI: 10.5220/0010010001730177. URL: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010010001730177.
Parsons, C. P., I. C. Parsons, and N. H. Parsons (July 2013). "Interactive climbing wall system using touch sensitive, illuminating, climbing hold bolts and controller". In:
Qt | Cross-platform software development for embedded desktop (2022). URL: https://www.qt.io/ (visited on 03/20/2022).
Quaine, F. and L. Martin (Dec. 1999). "A biomechanical study of equilibrium in sport rock climbing". In: *Gait and Posture* 10.3, pp. 233–239. DOI: 10.1016/S0966-6362(99)00024-7.
Quaine, F., L. Martin, and J. P. Blanchi (Apr. 1997a). "Effect of a leg movement on the organisation of the forces at the holds in a climbing position 3-D kinetic analysis". In: *Human Movement Science* 16.2-3, pp. 337–346. DOI: 10.1016/S0167-9457(96)00060-7.
Quaine, F., L. Martin, and J. P. Blanchi (Feb. 1997b). "The effect of body position and number of supports on wall reaction forces in rock climbing". In: *Journal of*

Applied Biomechanics 13.1, pp. 14–23. DOI: 10.1123/jab.13.1.14. URL: https://journals.humankinetics.com/view/journals/jab/13/1/article-p14.xml.

Rätsch, M., S. Romdhani, and T. Vetter (2004). "Efficient face detection by a cascaded support vector machine using haar-like features". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3175, pp. 62–70. DOI: 10.1007/978-3-540-28649-3_8. URL: https://link.springer.com/chapter/10.1007/978-3-540-28649-3_8.

Redmon, J. (2016a). *Darknet: Open Source Neural Networks in C*. URL: https://pjreddie.com/darknet/ (visited on 02/02/2022).

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi (2016b). *You Only Look Once: Unified, Real-Time Object Detection*. Tech. rep. URL: http://pjreddie.com/yolo/.

Redmon, J. and A. Farhadi (Apr. 2018). "YOLOv3: An Incremental Improvement". In: *arXiv*. arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

Rehman, A. and Z. Wang (2011). "SSIM-based non-local means image denoising". In: *Proceedings - International Conference on Image Processing, ICIP*, pp. 217–220. DOI: 10.1109/ICIP.2011.6116065.

— (2012). "Reduced-reference image quality assessment by structural similarity estimation". In: *IEEE Transactions on Image Processing* 21.8, pp. 3378–3389. DOI: 10.1109/TIP.2012.2197011.

Reveret, L., S. Chapelle, F. Quaine, and P. Legreneur (Sept. 2020). "3D Visualization of Body Motion in Speed Climbing". In: *Frontiers in Psychology* 11, p. 2188. DOI: 10.3389/fpsyg.2020.02188. URL: /pmc/articles/PMC7549505/%20/pmc/articles/PMC7549505/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7549505/.

Richter, J., R. Beltrán Beltrán, G. Köstermeyer, and U. Heinkel (2020). *Human Climbing and Bouldering Motion Analysis: A Survey on Sensors, Motion Capture, Analysis Algorithms, Recent Advances and Applications*. Tech. rep. URL: https://orcid.org/0000-0002-2681-5801.

Rosebrock, A. (2021). *OpenCV Getting and Setting Pixels - PyImageSearch*. URL: https://www.pyimagesearch.com/2021/01/20/opencv-getting-and-setting-pixels/ (visited on 02/02/2022).

Sara, U., M. Akter, and M. S. Uddin (Mar. 2019). "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study". In: *Journal of Computer and Communications* 07.03, pp. 8–18. DOI: 10.4236/jcc.2019.73002. URL: http://www.scirp.org/journal/jcc.

Setiadi, D. R. I. M. (Mar. 2021). "PSNR vs SSIM: imperceptibility quality assessment for image steganography". In: *Multimedia Tools and Applications* 80.6, pp. 8423–8444. DOI: 10.1007/s11042-020-10035-z. URL: https://doi.org/10.1007/s11042-020-10035-z.

Starovoitov, V. V., E. E. Eldarova, and K. T. Iskakov (2020). "Comparative analysis of the ssim index and the pearson coefficient as a criterion for image similarity". In: *Eurasian Journal of Mathematical and Computer Applications* 8.1, pp. 76–90. DOI: 10.32523/2306-6172-2020-8-1-76-90.

Strickler, J. H., P. Alto, and B. R. Kusse (Dec. 1994). *United States Patent (19) Strickler et al. 54) ROUTE RECORDING, MARKING, AND*. Tech. rep.

Tejas R. Phase (2020). "Building Custom HAAR-Cascade Classifier for face Detection". In: *International Journal of Engineering Research and* V8.12, pp. 881–886. DOI: 10.17577/ijertv8is120350.

TensorFlow (2021). URL: https://www.tensorflow.org/ (visited on 10/27/2021).

Tensorflow (2022a). *Distributed training with TensorFlow | TensorFlow Core*. URL: https://www.tensorflow.org/guide/distributed_training (visited on 02/03/2022).

Tensorflow (2022b). *Transfer learning and fine-tuning | TensorFlow Core*. URL: https://www.tensorflow.org/guide/keras/transfer_learning (visited on 02/03/2022).

Thulasya Naik, B., M. Farukh Hashmi, C. Author, and M. Farukh Hashmi mdfarukh (2021). "Ball and Player Detection Tracking in Soccer Videos Using Improved YOLOV3 Model". In: DOI: 10.21203/rs.3.rs-438886/v1. URL: https://doi.org/10.21203/rs.3.rs-438886/v1.

Tiator, M., B. Fischer, C. Geiger, L. Gerhardt, H. Preu, B. Dewitz, and D. Nowottnik (July 2018). "Venga!" In: *ACM International Conference Proceeding Series*. Vol. Part F1376. New York, New York, USA: Association for Computing Machinery, pp. 1–8. DOI: 10.1145/3210299.3210308. URL: http://dl.acm.org/citation.cfm?doid=3210299.3210308.

Torino, P. di, D. Maffiodo Raffaella Sesana, and M. Donno (2020). *Performance evaluation in indoor sport climbing*. Tech. rep.

Van der Walt, S., J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu (June 2014). "Scikit-image: Image processing in python". In: *PeerJ* 2014.1, e453. DOI: 10.7717/peerj.453. URL: https://developers.google.com/.

Viola, P. and M. Jones (2001a). "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. DOI: 10.1109/cvpr.2001.990517. URL: http://www.merl.com.

— (2001b). *Robust Real-time Object Detection*. Tech. rep.

Wang, S., A. Rehman, Z. Wang, S. Ma, and W. Gao (Apr. 2012). "SSIM-motivated rate-distortion optimization for video coding". In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.4, pp. 516–529. DOI: 10.1109/TCSVT.2011.2168269.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (Apr. 2004a). "Image quality assessment: From error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

Wang, Z., L. Lu, and A. C. Bovik (Feb. 2004b). "Video quality assessment based on structural distortion measurement". In: *Signal Processing: Image Communication*. Vol. 19. 2, pp. 121–132. DOI: 10.1016/S0923-5965(03)00076-6.

Weisberg, S. (2005). "Applied Linear Regression". In: *Classical Methods of Statistics*, pp. 113–160. DOI: 10.1007/3-540-29288-8_3.

Wiehr, F., F. Daiber, F. Kosmalla, and A. Krüger (May 2016). "BetaCube - Enhancing training for climbing by a self-calibrating camera-projection unit". In: *Conference on Human Factors in Computing Systems - Proceedings*. Vol. 07-12-May-2016. New York, NY, USA: Association for Computing Machinery, pp. 1998–2004. DOI: 10.1145/2851581.2892393. URL: https://dl.acm.org/doi/10.1145/2851581.2892393.

Wojke, N., A. Bewley, and D. Paulus (Feb. 2018). "Simple online and realtime tracking with a deep association metric". In: *Proceedings - International Conference on Image Processing, ICIP*. Vol. 2017-Septe. IEEE Computer Society, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962. arXiv: 1703.07402.

Zhang, S., S. Lan, Q. Bu, and S. Li (June 2019). "YOLO based intelligent tracking system for curling sport". In: *Proceedings - 18th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2019*. Institute of Electrical and Electronics Engineers Inc., pp. 371–374. DOI: 10.1109/ICIS46139.2019.8940229.

Zhang, Y., Z. Chen, and B. Wei (Dec. 2020). "A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement". In: *2020 IEEE 6th International Conference on Computer and Communications, ICCC 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 1312–1316. DOI: 10.1109/ICCC51575.2020.9345010.

Zhang, Z. (2019). *yolov3-tf2/training_voc.mdatmasterzzh8829/yolov3 – tf2GitHub*. URL: https://github.com/zzh8829/yolov3-tf2/blob/master/docs/training_voc.md (visited on 01/12/2022).