



Kiel University
Christian-Albrechts-Universität zu Kiel

Technische Fakultät
Institut für Informatik
Prof. Dr.-Ing. Sven Tomforde

Proposal

zur Masterarbeit

Entwicklung eines KI-basierten
Single-Camera-Systems für das Scoring
von Steeldarts durch die synthetische
Erzeugung realistischer Daten

vorgelegt von

Justin Fürstenwerth
stu**210298**@mail.uni-kiel.de
Informatik 1-Fach Master

1 Motivation

Der Anstoß zu dieser Masterarbeit ist ein Paper von McNally et al. aus dem Mai 2021 mit dem Titel „DeepDarts: Modeling Keypoints as Objects for Automatic Scorekeeping in Darts using a Single Camera“ [12]. Wie aus dem Titel des Papers hervorgeht, adressiert es die Herausforderung, die Punktzahl eines Dartspiels mithilfe eines Single-Camera-Setups, zu dem auch Mobiltelefone zählen, zu ermitteln. Der spezifische Fokus liegt auf Steeldarts, für die bereits etablierte Multi-Camera-Systeme zum Scorekeeping existieren [6, 13]. Diese Systeme erfordern die Installation und Kalibrierung mehrerer Kameras, um Dartpfeile präzise zu erfassen. Obwohl dieses Setup in der Lage ist, die Position der Dartscheibe und der Pfeile effektiv zu bestimmen, ist es aufgrund des erheblichen finanziellen Aufwands für die Systeme und der Notwendigkeit spezieller Hardware nicht für den Privatgebrauch sowie für Gelegenheits-Spieler geeignet.

Das vorliegende Referenzpaper bildet die Grundlage dieser Masterarbeit, deren Ziel es ist, ein System zu entwickeln, das Scorekeeping von Steeldarts vereinfacht und für den (gelegentlichen) Privatgebrauch unter Verwendung einer einzigen Kamera ermöglicht, wie sie heute in nahezu jedem Mobiltelefon verbaut ist.

McNally et al. haben in ihrer Arbeit Erfolge mit ihrer Methodik erzielt, dennoch existieren einige Herausforderungen und Ungereimtheiten, bei denen ihr System ungenaue Vorhersagen trifft. Diese Schwachstellen und Grenzfälle dienen als Grundlage für diese Masterarbeit, die darauf abzielt, auf den gewonnenen Erkenntnissen aufzubauen und ein insgesamt robusteres und präziseres System zur Erkennung von Dartpfeilen in Single-Camera-Setups, insbesondere im Kontext von Handyaufnahmen, zu entwickeln.

Obwohl bereits Systeme zur Erkennung von Dartpfeilen existieren, sind diese häufig in ihrem Umfang und ihrer Funktionalität eingeschränkt und weisen zahlreiche potenzielle Fehlerquellen auf oder erfordern besondere Maßnahmen zur Kalibrierung [8, 9, 10].

2 Ziele

Diese Masterarbeit stellt eine Erweiterung und Verbesserung eines bereits bestehenden Systems dar. Der Kern der Zielstellung dieser Arbeit ist analog zu dem Referenzpaper die Etablierung eines Single-Camera-Systems für das Scorekeeping von Steeldarts. Dieses System ist darauf ausgelegt, potenziell auf mobilen Endgeräten eingesetzt zu werden, wodurch in Anspruch zu nehmende Ressourcen begrenzt sind. Diese Voraussetzung ist kein zentraler Aspekt dieser Arbeit, sollte aber bei der Wahl der eingesetzten Techniken stets im Hinterkopf behalten werden.

Das zentrale Ziel dieser Arbeit besteht darin, die Schwachstellen des Systems von McNally et al. systematisch zu identifizieren und gezielt zu adressieren.

2.1 Schwachstellen im Paper

Im Referenzpaper wurden bereits einige Schwachstellen des Systems erwähnt. Neben diesen explizit erwähnten Problemen wurden weitere Problembereiche identifiziert, die in diesem Unterkapitel aufgeführt und erläutert werden.

2.1.1 Datengrundlage

Der Ansatz von McNally et al. basiert auf neuronalen Netzen, die auf einer begrenzten und unsicheren Datengrundlage trainiert wurden. Die verwendeten Bilddaten stammen aus zwei Quellen: Ein Großteil der Daten wurde mit einem fix montierten iPhone aufgenommen, während der Rest mit einer DSLR-Kamera aus unterschiedlichen Perspektiven aufgenommen wurde [11]. Beide Quellen sind nicht repräsentativ für den vorgesehenen Einsatz des Systems, in dem mit Handaufnahmen aus stark variierenden Perspektiven in unterschiedlichen Bedingungen gearbeitet wird. Dazu wurden diese Daten manuell per Hand annotiert, was mit Ungenauigkeiten einhergeht. Diese Ungenauigkeiten wurden in dem Paper anerkannt. Die Validität des Modells ist dadurch beeinträchtigt.

2.1.2 Entzerrung der Bilder

Im Gegensatz zu einer trivialen Herangehensweise geht das Referenzpaper davon aus, dass Bilder von Dartscheiben nicht perfekt sind und mit einer perspektivischen Verzerrungen einhergehen. Um verlässliche Vorhersagen bezüglich der Dartpfeile treffen zu können, muss die Dartscheibe entzerrt werden. Der vorgeschlagene Ansatz zur Entzerrung der Bilder besteht darin, vier Fixpunkte der Dartscheibe zu identifizieren und anhand dieser eine Homographie abzuleiten, die die Dartscheibe von einer Ellipse in einen perfekt ausgerichteten Kreis transformiert. Diese Herangehensweise setzt jedoch voraus, dass alle vier Fixpunkte im Bild erkennbar sind. Sollte einer der Punkte beispielsweise durch einen Dartpfeil verdeckt oder aufgrund der Abnutzung der Dartscheibe nicht mehr zu erkennen sein, ist eine korrekte Entzerrung auf diese Weise nicht möglich. Dieser Schwachpunkt wurde im Paper anerkannt und als verbesserungswürdig hervorgehoben.

2.1.3 Generelle Verbesserung der Vorhersagen

Laut eigener Metrik hat das vorgestellte System eine Genauigkeit von 84%. Diese Genauigkeit ist nach subjektiver Einschätzung ausreichend, um einen Proof-of-Concept zu validieren, jedoch nicht für den realen Einsatz dieses Systems in einem Live-Umfeld. Teile dieser Ungenauigkeit sind auf bereits genannte Probleme zurückzuführen. Darauf hinaus sind seit der Veröffentlichung des Papers neue und fortgeschrittene Technologien im Bereich der neuronalen Netze und der Computer Vision entwickelt worden, die Möglichkeiten zur Verbesserung des Systems bieten.

2.2 Problemlösung

Das gezielte Angehen der genannten Probleme bildet den Schwerpunkt dieser Masterarbeit. Es soll ein insgesamt robusteres und genaueres System konzipiert und umgesetzt werden. Ein Kernpunkt der Umsetzung ist die Nutzung von 3D-Software, um automatisiert große Mengen synthetischer Daten zu generieren. Auf diese Weise können Verzerrungen der Daten aufgrund festgelegter Umgebungsbedingungen, Begrenzung auf wenige Kameraparameter oder fehlerhaft annotierter Daten gemindert werden. Darüber hinaus werden reale Daten aufgenommen, die im späteren Einsatz in einer App anfallen könnten. Dabei wird ein Protokoll über getroffene Felder geführt, um die Wahrscheinlichkeit fehlerhafter Erkennungen zu minimieren. Des Weiteren wird eine neue Methodik zur Erkennung der Dartscheibe implementiert, die herkömmliche Computer Vision-Techniken umfasst, um Dartscheiben analytisch zu identifizieren. Falls dies nicht zuverlässig möglich ist, wird dieser Ansatz durch den Einsatz von CNNs ergänzt. Die Verwendung von KI zur Ermittlung der Dartpfeile bleibt ein zentraler Aspekt des Systems.

Um die Systeme angemessen miteinander vergleichen zu können, ist es von Bedeutung, dass das im Paper verwendete Modell lauffähig gemacht wird und die Daten, auf denen das Modell trainiert und evaluiert wurde, verwendet werden können. Die relevanten Daten sind öffentlich verfügbar und können eingesehen und heruntergeladen werden [11]. Sobald das System funktionsfähig ist, können auch eigene Statistiken erhoben werden, um die Performance der verschiedenen Ansätze zu vergleichen, wobei eine Verbesserung der Resultate angestrebt wird.

3 Methodik

In diesem Kapitel wird die Methodik dargestellt, die zur Erreichung der gesetzten Ziele eingesetzt wird. Die Arbeit gliedert sich in verschiedene Teilbereiche, die im Folgenden isoliert betrachtet werden.

3.1 Datensammlung

Die Datensammlung bildet den ersten zentralen Punkt dieser Arbeit. Die Datengrundlage der Arbeit von McNally et al. weist mehrere verbesserungswürdige Aspekte auf, die durch eine neue Herangehensweise angegangen werden sollen. In dieser Arbeit wird mit zwei unterschiedlichen Arten von Daten gearbeitet: realen Daten und synthetischen Daten.

Der erste Teil der Gesamtdatenmenge umfasst die realen Daten. Hierfür werden Steeldarts-Runden live gespielt, wobei dokumentiert wird, in welchen Feldern die Pfeile landen, um fehlerhafte Annotierungen zu unterbinden. Es kommen mehrere verschiedene Handykameras zum Einsatz, um eine möglichst vielfältige und realitätsnahe Datenbasis zu schaffen. Diese realen Daten stellen lediglich einen kleinen Teil der Gesamtdaten dar, sollten jedoch ausreichend groß

und divers sein, um Rückschlüsse auf die tatsächliche Performance zu ermöglichen.

Der Hauptanteil der Datengrundlage dieser Arbeit besteht nicht aus manuell annotierten und damit potenziell fehleranfälligen Daten, sondern wird mittels 3D-Software generiert. Die synthetischen Daten werden unter Verwendung von Ray Tracing und fotorealistischen Texturen erstellt. Dieser Teil der Daten wird den Großteil der Gesamtdatenmenge ausmachen, da er automatisiert erstellt und korrekt annotiert werden kann, was eine solide Datengrundlage für das System bietet.

Zur Erstellung der synthetischen Daten werden typische Parameter der Bildaufnahme durch Metadatenanalyse und Online-Recherchen ermittelt und nachgestellt, um eine möglichst realistische Abbildung der zu erwartenden Daten zu erzielen.

3.2 Dartscheiben-Identifizierung

Eine Methode, die im Referenzpaper angewandt wurde und in dieser Arbeit übernommen werden soll, ist die Entzerrung der Eingabebilder in Bezug auf die Dartscheibe. Es muss davon ausgegangen werden, dass Aufnahmen nicht frontal, sondern in einem Winkel $\neq 90^\circ$ zur Dartscheibe gemacht werden. Daher ist eine Entzerrung der Bilder erforderlich, um die Feldwerte korrekt anhand der Pfeilpositionen einordnen zu können.

Die im Paper eingesetzte Methode ist fehleranfällig und nicht zuverlässig. Daher wird auf Algorithmen der Computer Vision zurückgegriffen, um eine möglichst robuste Erkennung der Dartscheibe zu gewährleisten und darauf aufbauend eine Entzerrung des Bildes durchzuführen. Hierbei stehen verschiedene Techniken wie Farbraum-Transformation, Thresholding, Filterung sowie Blob- und Keypoint-Detektion zur Verfügung. Darüber hinaus können fortgeschrittene Verarbeitungsschritte wie Hough-Transformationen und Ellipsenerkennung eingesetzt werden. Es ist ebenfalls denkbar, CNN-Architekturen zu integrieren, um die Keypoint-Detektion zu verbessern.

Ein Beispiel für den Ablauf einer Entzerrung eines Bildes ist in Abbildung 1 dargestellt. Diese Abbildung dient lediglich als schematische Darstellung und kann von der tatsächlichen Methodik abweichen.

3.3 KI-Training / Dartpfeil-Identifizierung

Wie im Referenzpaper wird die Identifizierung und Lokalisierung der Dartpfeile auf neuronalen Netzen basieren. Dieser Aspekt ist hinsichtlich des Arbeitsaufwands der größte Teil dieser Arbeit. Es wird auf Transfer Learning von Objekterkennungs-KIs zurückgegriffen, um auf aktuellen, leistungsstarken KIs aufzubauen. Zur Steuerung des Trainings werden eigene Methoden entwickelt, die teilweise auf der Keypoint-Detektion von McNally et al. basieren. Bei der Implementierung dieser KI-Methoden kommen aktuelle KI-Frameworks zum Einsatz.



Abbildung 1: Veranschaulichung möglicher Schritte eines Ablaufes zur Entzerung einer Dartscheibe

3.4 System-Statistiken

Um einen Überblick über die Performance des Systems zu erhalten, werden im Verlauf der Arbeit Statistiken erstellt, die eine umfangreiche Auswertung des Systems ermöglichen. Diese Statistiken beziehen sich auf die Trefferquote bestimmter Felder und übernehmen Metriken aus dem Referenzpaper, um einen objektiven Vergleich der unterschiedlichen Herangehensweisen zu gewährleisten. Des Weiteren werden Statistiken über die Art des Feldes - ob einfache, Double oder Triple - sowie Feldwerte ohne Vervielfältigungen, Genauigkeit der Vorhersagen in Abhängigkeit von den Pfeilabständen auf dem Board oder das Verhalten des Systems bei Änderungen des Kamerawinkels erhoben. Der genaue Umfang der Statistiken wird im Verlauf der Arbeit in Bezug auf Aussagekraft und Umsetzbarkeit ermittelt.

4 Implementierung

Die Implementierung gliedert sich in mehrere Schritte, die in den folgenden Unterkapiteln jeweils detailliert dargestellt werden, um den Entwicklungsprozess präzise zu beschreiben.

4.1 Datenerstellung

Zur Generierung synthetischer Daten wird die Open-Source-Software Blender verwendet, die sich durch ihre umfangreichen Möglichkeiten zur Erstellung fotorealistischer 3D-Szenen auszeichnet [3]. Mit Blender lassen sich dataillierte Szenen modellieren, die Dartscheiben und Pfeile in verschiedenen Umgebungen darstellen. Ein Beispiel für eine erstellte Szene ist in Abbildung 2 zu sehen, das Ergebnis eines Renderings in Abbildung 3. Durch parametrisierbare Erstellung

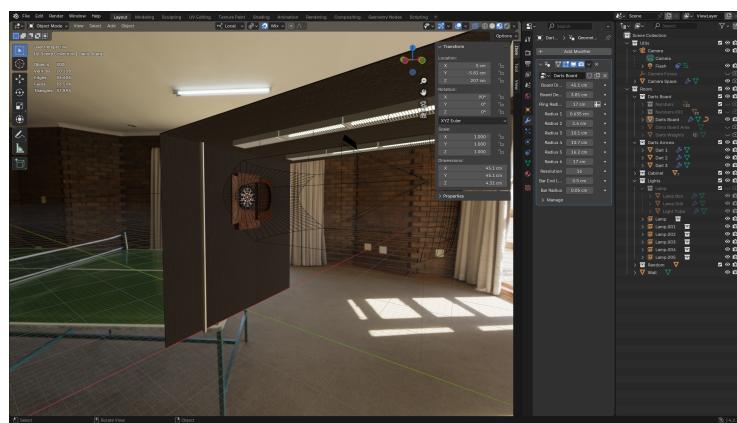


Abbildung 2: Blender-Projekt zur Generierung von Dartscheiben

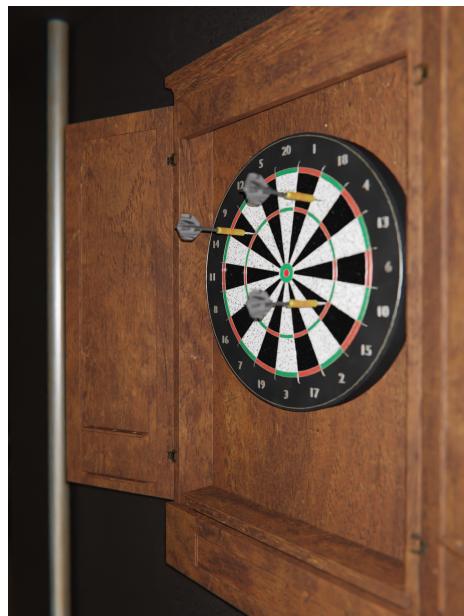


Abbildung 3: Rendering einer Dartscheibe aus dem Blender-Projekt

von Objekten und Szenen ist eine modulare und vielfältige Variation der Renderings möglich, durch welche eine große Spanne verschiedener Szenarien simuliert werden kann.

Der Zugriff auf die Blender-Szene wird durch die Python-Bibliothek *bpy* ermöglicht [4]. Diese erlaubt eine vollständige Automatisierung der Datenproduktion ohne grafische Benutzeroberfläche. Mithilfe von Skripten können Parameter wie Position, Beleuchtung und Texturen von Objekten sowie Kameraparameter manipuliert werden. Besonders vorteilhaft ist, dass dieser Prozess keine lokale Blender-Installation erfordert und dadurch auf GPU-Servern durchgeführt werden kann, wodurch eine flexible Skalierung und Verarbeitung großer Datens Mengen möglich ist.

Das Variieren von Parametern, wie der Lichtverhältnisse, der Oberflächenbeschaffenheit oder Kameraposition, erlaubt die realistische Nachbildung unterschiedlicher Aufnahmebedingungen. Eine hohe Diversität wird durch die Simulation verschiedener Lichtquellen, Raumkonfigurationen und Texturen erreicht. Parameter können dabei sowohl zufällig aus einer vordefinierten Spanne an Werten, wie auch anhand von Heatmaps und Wahrscheinlichkeitsverteilungen gesetzt werden. Ein weiterer wichtiger Aspekt der Datenerstellung ist die korrekte Annotation der synthetischen Bilder. Da die Daten aus einer virtuellen Umgebung generiert werden, ist sichergestellt, dass jede Bild-Pixel-Annotation präzise ist. Dies ermöglicht eine effiziente und fehlerfreie Vorbereitung der Daten für das Training.

4.2 Erkennung der Dartscheibe

Die Dartscheibe stellt durch ihren strukturierten und kontrastreichen Aufbau eine ideale Grundlage für Bildverarbeitungsalgorithmen dar. Der universelle Aufbau aus radial angeordneten Segmenten und Ringen, die klar unterscheidbare Farben und Formen aufweisen, erlaubt es, robuste Verfahren zur Dartscheiben-Detektion einzusetzen [2, 7].

Für die Erkennung wird auf klassische Methoden der Computer Vision zurückgegriffen, wie etwa Kantendetektion und die Hough-Transformation zur Kreis- und Ellipsenerkennung. Durch diese Verfahren wird die Geometrie der Dartscheibe extrahiert und anschließend eine Entzerrung der perspektivischen Aufnahme vorgenommen. Diese klassischen Verfahren können durch den Einsatz neuronaler Netze, insbesondere CNNs, ergänzt werden, um eine robuste Erkennung auch unter schwierigen Bedingungen zu gewährleisten, beispielsweise wenn Teile der Dartscheibe verdeckt sind. Für die Implementierung dieser Funktionen wird die Bibliothek *OpenCV* verwendet [5].

4.3 Training neuronaler Netze zur Erkennung von Dartpfeilen

Für die Lokalisierung der Dartpfeile wird ein neuronales Netz entwickelt, das mit synthetischen und realen Daten trainiert wird. Die Implementierung erfolgt

mit dem *TensorFlow*-Framework, das umfassende Möglichkeiten zur Datenverarbeitung bietet und insbesondere durch seine Optimierung für das Training auf GPUs eine effiziente Handhabung großer Datenmengen erlaubt [1].

Die Trainingsdaten werden in drei Hauptkategorien unterteilt: Trainings-, Validierungs- und Testdaten. Ein Großteil der Trainingsdaten wird aus synthetischen Daten bestehen, während zur Validierung hauptsächlich reale Daten verwendet werden. Für die abschließende Testphase wird auf Datensätze zurückgegriffen, die im Referenzpaper verwendet wurden. Diese strukturierte Aufteilung der Daten gewährleistet einen aussagekräftigen Vergleich zwischen dem entwickelten System und dem Referenzmodell von McNally et al.

Ein zentraler Punkt zur Verbesserung der Modellleistung ist die Datenaugmentierung. Diese umfasst sowohl pixelbasierte Transformationen der Eingabedaten, wie das Hinzufügen von Rauschen oder die Anpassung von Helligkeit und Kontrast, als auch geometrische Transformationen der Eingabe- und Ausgabedaten, wie Spiegeln, Rotieren oder Skalieren der Bilder. Wichtig hierbei ist, dass Eingangs- und Ausgangsdaten stets synchron transformiert werden, um eine Verfälschung der Daten zu verhindern. Durch die Einführung solcher Augmentierungen wird das Modell auf eine Vielzahl von realistischen Szenarien vorbereitet, was das Risiko von Overfitting, dem Auswendiglernen von Daten, erheblich reduziert.

Diese Ansätze gewährleisten, dass das entwickelte Modell in der Lage ist, robuste und präzise Vorhersagen zur Position der Dartpfeile zu treffen. Darüber hinaus legen sie den Grundstein für eine systematische Verbesserung gegenüber dem ursprünglichen Ansatz.

5 Evaluation

Diese Arbeit stellt eine Erweiterung eines bestehenden Systems dar und muss daher im Kontext der ursprünglichen Ergebnisse evaluiert werden. In dem Referenzpaper wurden bereits Metriken zur Messung der Genauigkeit und Leistungsfähigkeit des Systems etabliert, die für die Evaluation dieser Arbeit übernommen werden, um eine konsistente Vergleichbarkeit der Systeme zu gewährleisten. Zusätzlich besteht die Möglichkeit, eigene, falsifizierbare Metriken zu entwickeln, um spezifische Aspekte der Systemleistung weiter zu untersuchen. Die Bewertung des Systems wird nicht ausschließlich anhand eigener Daten erfolgen, sondern insbesondere durch den Einsatz der in Referenzpaper von McNally et al. verwendeten Datensätze, um die Vergleichbarkeit der Ergebnisse zu gewährleisten.

Hinsichtlich der Systemperformance werden umfangreiche Statistiken gesammelt, die durch die synthetische Datenerstellung vielseitig ausfallen. Eine denkbare Auswertung betrifft die Genauigkeit der Vorhersagen in spezifischen Bereichen der Dartscheibe. Zudem kann untersucht werden, in welchen Szenarien die Methodik von McNally et al. Ungenauigkeiten aufweist und wie sich das in dieser Arbeit entwickelte System in diesen Fällen verhält. Im Gegenzug lassen sich auch Schwachstellen des hier präsentierten Ansatzes identifizieren und

vergleichen. Weitere statistische Auswertungen könnten sich auf die Trefferart (einfach, doppelt oder dreifach), die getroffenen Felder oder die Genauigkeit der Vorhersagen in Abhängigkeit vom Abstand der Dartpfeile zueinander beziehen.

Da das System für einen möglichen Einsatz in mobilen Endgeräten vorgesehen ist, bietet sich zudem ein Effizienzvergleich der Systeme an. Hierbei könnte die Erkennungsdauer auf identischer Hardware als Maßstab herangezogen werden, um die Laufzeitoptimierung und Eignung für mobile Anwendungen zu bewerten.

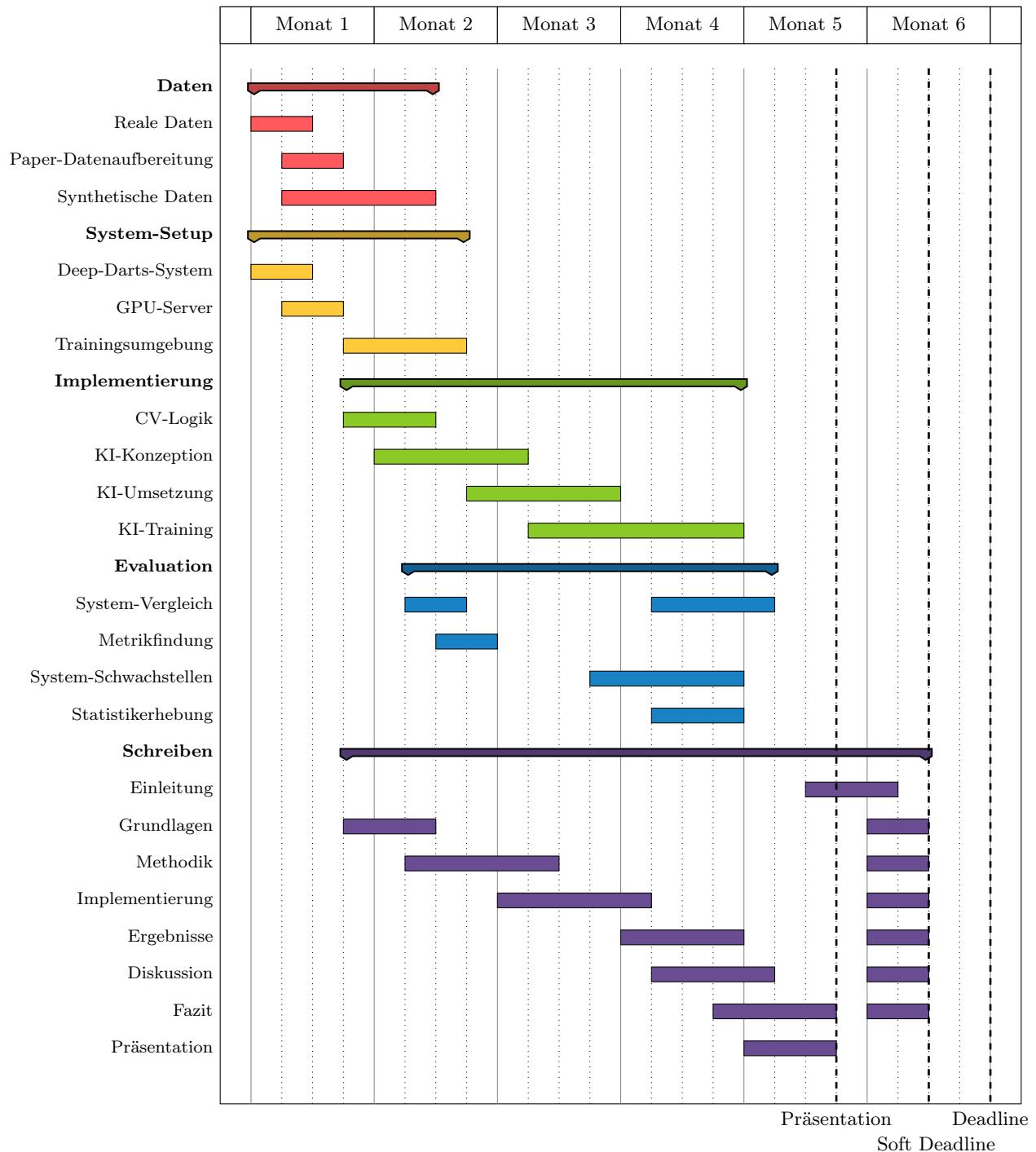
6 Zeitplanung

In diesem Abschnitt wird die geplante zeitliche Strukturierung der Masterarbeit dargestellt. Die verschiedenen Aufgabenbereiche wie Systemaufbau, Datengenerierung, KI-Training, Evaluation und das Schreiben der Abschlussarbeit wurden in einem Gantt-Chart¹ organisiert, um einen Überblick über die Projektfortschritte zu gewährleisten. Dieses Chart stellt die voraussichtlichen Zeiträume dar, in denen die jeweiligen Aufgaben bearbeitet werden sollen.

Die Struktur des Gantt-Charts erlaubt die einfache Visualisierung der parallel laufenden Arbeitsschritte. Es zeigt, wie Datengenerierung, System-Setup, Implementierung, Evaluation und Schreiben der Arbeit ineinander greifen. Diese Abbildung erleichtert die kontinuierliche Anpassung des Zeitplans, falls unvorhergesehene Herausforderungen auftreten. Der Schreibprozess findet parallel zur Umsetzung des Projektes statt und ist so geplant, dass eine Phase direkt nach ihrer Vollendung mit aufgenommen werden kann. Zudem wurde ausreichend Zeit für etwaige Verzögerungen eingeplant.

Die detaillierte Planung erlaubt es, wichtige Meilensteine und Deadlines zu definieren und gibt einen klaren Überblick über die zeitliche Abfolge der Schritte. Der Fortschritt wird im Laufe der Arbeit anhand dieser Grafik gemessen und eingeordnet, um sicherzustellen, dass das Projekt im vorgesehenen Rahmen abgeschlossen wird.

¹Paketdokumentation: <https://ftp.rrze.uni-erlangen.de/ctan/graphics/pgf/contrib/pgfgantt/pgfgantt-doc.pdf>



HALLO

Literatur

- [1] ABADI, Martín ; AGARWAL, Ashish ; BARHAM, Paul ; BREVDO, Eugene ; CHEN, Zhifeng ; CITRO, Craig ; CORRADO, Greg S. ; DAVIS, Andy ; DEAN, Jeffrey ; DEVIN, Matthieu ; GHEMAWAT, Sanjay ; GOODFELLOW, Ian ; HARP, Andrew ; IRVING, Geoffrey ; ISARD, Michael ; JIA, Yangqing ; JOZEFOWICZ, Rafal ; KAISER, Lukasz ; KUDLUR, Manjunath ; LEVENBERG, Josh ; MANÉ, Dandelion ; MONGA, Rajat ; MOORE, Sherry ; MURRAY, Derek ; OLAH, Chris ; SCHUSTER, Mike ; SHLENS, Jonathon ; STEINER, Benoit ; SUTSKEVER, Ilya ; TALWAR, Kunal ; TUCKER, Paul ; VANHOUCKE, Vincent ; VASUDEVAN, Vijay ; VIÉGAS, Fernanda ; VINYALS, Oriol ; WARDEN, Pete ; WATTENBERG, Martin ; WICKE, Martin ; YU, Yuan ; ZHENG, Xiaoqiang: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. – URL <https://www.tensorflow.org/>. – Software available from tensorflow.org
- [2] AUTHORITY, Darts R.: *Rule Book*. 2023. – URL https://www.thedra.co.uk/_files/ugd/20bb2f_fbc16a1efdf34cf9b8d377a5657a4daa.pdf. – Accessed: 2024-10-23
- [3] BLENDER ONLINE COMMUNITY: *Blender - a 3D modelling and rendering package*. Amsterdam, The Netherlands: Blender Foundation (Veranst.), 2024. – URL <http://www.blender.org>. – Version 4.2.1
- [4] BLENDER ONLINE COMMUNITY: *bpy: The Blender Python API*. 2024. – URL <https://docs.blender.org/api/current/>. – Version 4.2.0
- [5] BRADSKI, Gary ; KAEHLER, Adrian: *OpenCV: Open Source Computer Vision Library*. 2024. – URL <https://opencv.org/>. – Accessed: 2024-10-23
- [6] DARTS, Auto: *Auto Darts - Official Website*. 2024. – URL <https://autodarts.io>. – Accessed: 2024-10-23
- [7] FEDERATION, World D.: *Playing and Tournament Rules*. 2018. – URL https://dartsfdf.com/storage/uploads/fb05b306-c92c-4f08-b512-affb092a1b3d/2018-02-28_WDF_Playing_and_Tournament_Rules_rev20.pdf. – Accessed: 2024-10-23
- [8] GUDJONS, Lars: *Darts Project*. GitHub repository. 2023. – URL https://github.com/LarsG21/Darts_Project. – Accessed: 2024-10-23
- [9] HOETTINGER, Hannes: *opencv-steel-darts*. GitHub repository. 2017. – URL <https://github.com/hanneshoettinger/opencv-steel-darts>. – Accessed: 2024-10-23
- [10] KENTO kawasaki: *darts-score-detection*. GitHub repository. 2021. – URL <https://github.com/kawasaki-kento/darts-score-detection>. – Accessed: 2024-10-16

- [11] McNALLY, William: *DeepDarts Dataset*. 2021. – URL <https://dx.doi.org/10.21227/05e7-xs69>
- [12] McNALLY, William ; WALTERS, Pascale ; VATS, Kanav ; WONG, Alexander ; MCPHEE, John: *DeepDarts: Modeling Keypoints as Objects for Automatic Scorekeeping in Darts using a Single Camera*. 2021. – URL <https://arxiv.org/abs/2105.09880>
- [13] TECHNOLOGIES, SCOLIA: *Scolia - Official Website*. 2024. – URL <https://scoliadarts.com>. – Accessed: 2024-10-23