

BACHELOR THESIS
Sebastian Schmidpeter

Sturzerkennung mittels Pose Estimation im Kontext von Smart Home Umgebungen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Sebastian Schmidpeter

Sturzerkennung mittels Pose Estimation im Kontext von Smart Home Umgebungen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas Lehmann
Zweitgutachter: Prof. Dr. Kai von Luck

Eingereicht am: 10.10.2022

Sebastian Schmidpeter

Thema der Arbeit

Sturzerkennung mittels Pose Estimation im Kontext von Smart Home Umgebungen

Stichworte

Sturzerkennung, Pose Estimation, Smart Home, Ambient Assisted Living

Kurzzusammenfassung

In dieser Arbeit wird eine Sturzerkennung entwickelt, die auf den, durch Pose Estimation, erzeugten Daten basiert. Die Sturzerkennung entscheidet anhand der aktuellen identifizierten Haltung einer Person und den vergangenen Haltungen, ob ein Sturz stattgefunden hat oder nicht. Die Parameter des Systems werden im Sinne einer Optimalkonfiguration untersucht. Die Sturzerkennung wird mit Aufnahmen unterschiedlichen Ursprungs getestet und bewertet.

Sebastian Schmidpeter

Title of Thesis

Fall detection using pose estimation in the context of smart home environments

Keywords

fall detection, pose estimation, smart home, ambient assisted living

Abstract

In this thesis, a fall detection system is developed based on the data generated by Pose Estimation. The fall detection system decides whether a fall has occurred or not based on the current identified pose of a person and the past poses. The parameters of the system are studied in terms of an optimal configuration. The fall detection is tested and evaluated with recordings from different sources.

Inhaltsverzeichnis

Abbildungsverzeichnis	vi
Tabellenverzeichnis	viii
Abkürzungen	ix
1 Einleitung	1
2 Stand der Technik	3
2.1 Smart Home	3
2.2 Ambient Assisted Living	4
2.3 Sturzerkennung	6
2.4 Pose Estimation	8
3 Problemanalyse	12
3.1 Definition „Sturz“	12
3.2 Definition valider Umgebungen für die Sturzerkennung mittels Pose Estimation	13
3.3 Requirements Engineering	14
4 Lösungsansätze	16
4.1 Lösungskonzepte	16
4.1.1 Auswahl Pose Estimation Bibliothek	16
4.1.2 Sturzerkennung über Beschleunigungswerte	16
4.1.3 Sturzerkennung über ein neuronales Netz	19
4.1.4 Sturzerkennung über Pose aus Keypoints	21
4.2 Systementwurf	22
4.2.1 Funktionsweise Pose-Klassifikator	22
4.2.2 Funktionsweise Sturzerkennung	24

5	Ergebnisse	28
5.1	Konfiguration MediaPipe	28
5.2	Auswertung von Webcam-Aufnahmen	30
5.2.1	Beschreibung der Aufnahmen	32
5.2.2	Auswertung	32
5.3	Ermitteln einer Optimalkonfiguration für die Schwellwerte	34
5.3.1	Test 1	35
5.3.2	Test 2 - Konstante Fallhöhe	36
5.3.3	Test 3 - Konstanter prozentualer Höhenunterschied	37
5.4	Auswertung von Aufnahmen in einer Smart Home Umgebung	38
5.4.1	Beschreibung der Aufnahmen	38
5.4.2	Auswertung	39
5.5	Auswertung von Videos Dritter	41
5.5.1	Beschreibung der Aufnahmen	41
5.5.2	Auswertung	42
6	Diskussion	44
6.1	Evaluierung der Ergebnisse	44
6.2	Ausblick	47
6.3	Fazit	49
	Literaturverzeichnis	50
	A Digitaler Anhang	54
	Glossar	55
	Selbstständigkeitserklärung	56

Abbildungsverzeichnis

2.1	Grundriss des LIVING PLACE lab. [1]	4
2.2	Tödliche Stürze in den USA 2001 nach Alter und Geschlecht. [32]	6
2.3	Die 33 Keypoints, die von MediaPipe extrahiert werden. [10]	9
2.4	Vergleich 2D und 3D Pose Estimation. Links: Verwendetes Bild. Mitte: 2D Pose Estimation. Rechts: 3D Pose Estimation. [21]	10
4.1	Visualisierung der Beschleunigungswerte während einem Sturz mittels Matplotlib.	18
4.2	Vier Bilder aus dem Datensatz, der für das nachträgliche Training des ResNet50 benutzt wurde. Links oben: Gebeugt, rechts oben: Stehend, links unten: Liegend, rechts unten: Sitzend. [18]	20
4.3	Zwei Bilder die mittels Segmentierung aufbereitet wurden und so an das ResNet50 übergeben werden. Links: Sitzend, rechts: Stehend.	21
4.4	Klassendiagramm der Sturzerkennung.	25
4.5	Aktivitätsdiagramm des Pose-Klassifikators.	26
4.6	Aktivitätsdiagramm der Sturzerkennung.	27
5.1	Einfluss des Parameters <code>min_detection_confidence</code> . Links: <code>min_detection_confidence = 0.5</code> , Rechts: <code>min_detection_confidence = 0.8</code>	31
5.2	Einfluss des Parameters <code>min_tracking_confidence</code> . Links: <code>min_tracking_confidence = 0.5</code> , Rechts: <code>min_tracking_confidence = 0.0</code>	31
5.3	Die Position des linken Fußes (grün markiert) wird missinterpretiert. Die Situation wird als Sturz gewertet.	33
5.4	Übersicht über die genutzten Domkamaseras des LP. Eigene Darstellung nach [1].	39
5.5	Oben: Nicht zugeschnittenes Bild, keine Keypoints können extrahiert werden. Unten: Zugeschnittenes Bild, Keypoints können extrahiert werden.	41

6.1	Zeitliche Abfolge der Bilder: links oben, rechts oben, links unten, rechts unten. Einfluss der Entfernung zwischen Kamera und Person für eine erfolgreiche Pose Estimation.	45
6.2	Aufnahme derselben Situation mit den Domkamas 1 und 6, vgl. Abbildung 5.4. Links: Kamera 6, keine Sturzmeldung aufgrund fehlender Pose Estimation. Rechts: Kamera 1, Pose Estimation erfolgreich, Sturz wird erkannt.	46

Tabellenverzeichnis

3.1	Requirements einer Sturzerkennung mittels Pose Estimation.	15
4.1	Vergleich von MediaPipe und OpenPose. [9, 15, 22]	17
5.1	Verwendete Konfiguration für MediaPipe.	31
5.2	Übersicht über die Webcam-Aufnahmen.	32
5.3	Verwendete Konfigurationen für Test 1.	35
5.4	Verwendete Konfigurationen für Test 2. Konstante Fallhöhe.	37
5.5	Verwendete Konfigurationen für Test 3. Konstanter prozentualer Höhen- unterschied.	37
5.6	Vergleich einer weiteren Konfiguration mit der Konfiguration 3.1.	38
5.7	Auswertung der Konfiguration 3.3 nach Abzug der Aufnahmen, für die keine Pose Estimation erfolgen konnte.	42
5.8	Konfusionsmatrix der Auswertung für Konfiguration 3.3.	42

Abkürzungen

AAL Ambient Assisted Living.

ADL Activity of Daily Living.

AIP Aging in Place.

HAW Hochschule für Angewandte Wissenschaften.

LP LIVING PLACE lab.

NN Neuronales Netz.

WHO World Health Organization.

1 Einleitung

In einer Veröffentlichung der KOMMISSION DER EUROPÄISCHEN GEMEINSCHAFTEN von 2006 [8] werden Faktoren für den demographischen Wandel und dessen Folgen aufgezeigt. So sorgen ein Reproduktionsniveau von 1,5¹, der Babyboom der Nachkriegszeit und ein deutlicher Anstieg der Lebenserwartung dafür, dass der Abhängigkeitsquotient² bis 2050 51 % erreichen dürfte und sich damit verdoppeln würde. Das bedeutet, dass auf jede Person von mindestens 65 Jahren nur noch zwei statt vier Personen im arbeitsfähigen Alter kommen. [8]

Personen in Pflegeberufen werden sich demzufolge künftig um noch mehr pflegebedürftige Personen kümmern müssen. Dabei ist eine dauerhafte Begleitung der Person oft nicht möglich und auch gar nicht wünschenswert. Eine Unterstützung „auf Knopfdruck“ kann derzeit über einen Hausnotruf geboten werden. Diese Systeme bestehen meist aus einem Notrufarmband oder einer Notrufkette mit Knopf und einer Basisstation. Durch Drücken des Knopfes wird eine Verbindung in die Notrufzentrale des Anbieters hergestellt. [7] Das System muss somit aktiv von der pflegebedürftigen Person getragen werden – ein Vergessen ist nicht ausgeschlossen. In Situationen, in denen die Person gestürzt ist und ihr Hausnotruf außer Reichweite ist, bietet ein solches System keine Unterstützung. An diesem Problem knüpft die vorliegende Arbeit an. Es wird ein System entwickelt, das ohne Zutun einer Person in der Lage ist einen Sturz zu erkennen. Für die Erkennung des Sturzes werden hier Kamerabilder genutzt. Diese Bilder können beispielsweise aus Webcams oder aus Domkameras stammen, wie sie vermehrt in Smart Home Umgebungen anzufinden sind. Für die Kamerabilder wird dabei zunächst eine Pose Estimation durchgeführt. Die Sturzerkennung arbeitet dann auf den Daten, die durch die Pose Estimation erzeugt wurden.

¹Ein Reproduktionsniveau von 2,1 wäre mindestens erforderlich, um die Bevölkerung aufrecht zu erhalten.

²Anzahl der Personen von mindestens 65 Jahren, bezogen auf die Personen von 15 bis 64 Jahren.

In dieser Arbeit wird zunächst das vorliegende Problem analysiert. Es werden drei Lösungsansätze vorgestellt und auf ihre Tauglichkeit geprüft. Der Lösungsansatz, der die größten Erfolgschancen verspricht, wird detailliert beschrieben, konfiguriert und mit unterschiedlichen Aufnahmen von Stürzen und Alltagssituationen getestet. Die so erzeugten Ergebnisse werden schließlich evaluiert und die Einbindung dieses Systems in ein reales Zuhause diskutiert.

2 Stand der Technik

2.1 Smart Home

Unter dem Begriff Smart Home ist ein Zuhause zu verstehen, welches mit Rechner, Sensoren und Aktuatoren ausgestattet ist. Verbaute Komponenten in einem Smart Home sind dabei teilweise oder vollständig untereinander und nach außen vernetzt [5]. Zu den Komponenten, welche häufig in einem Smart Home zu finden sind, zählen Fenster- und Türsensoren, Luftfeuchtigkeits- und Temperatursensoren, per Fernzugriff steuerbare Lichtquellen und Steckdosen, Kameras ...

Denkbare Services, die durch ein Smart Home dem Bewohnenden zur Verfügung gestellt werden können sind beispielsweise:

- Kontrolle über den Energieverbrauch,
- verbesserte Sicherheit,
- Unterhaltung,
- private Tagesplanung.

Die Services zielen dabei darauf ab, menschliche Interaktion mit dem eigenen Zuhause zu reduzieren, den Komfort der Bewohnenden zu erhöhen und einen zentralisierten Zugang für die Automation des Zuhauses bereitzustellen. Die gesammelten Informationen sollen es den Bewohnenden außerdem erleichtern, fundierte Entscheidungen über ihren Haushalt zu treffen. [4]

Die steigende Relevanz von Smart Home Konzepten zeigt eine Analyse der ZION MARKET RESEARCH. ZION MARKET RESEARCH schätzt die globalen Einnahmen, welche 2021 mit Komponenten für Smart Homes erwirtschaftet wurden, auf 85,6 Mrd. \$. Bis 2026 wird ein Anstieg dieser Summe auf 137,9 Mrd. \$ erwartet. [34]

Ebenso investiert die Forschung in diesen Bereich. Die HAW Hamburg hat für Forschungsfragen im Kontext von Smart Home Umgebungen das LIVING PLACE lab (LP) eingerichtet. Das LP ist eine Smart Home Umgebung, die von Lehrenden und Studierenden der HAW für Forschungsfragen genutzt wird [1]. Ein Teil dieser Arbeit findet in dem LP statt. Es werden vier der insgesamt sechs Domkameran des LP genutzt. Die Domkameran überblicken das Schlafzimmer, den Flur, die Küche und das Wohnzimmer (vgl. Grundriss in Abbildung 2.1).

Ein Forschungsbereich, der sich einige der Smart Home-Komponenten zu nütze macht und eng mit dem hier entwickelten System in Verbindung steht ist das Ambient Assisted Living (AAL).



Abbildung 2.1: Grundriss des LIVING PLACE lab. [1]

2.2 Ambient Assisted Living

„Ambient Assisted Living“ (AAL) steht für Konzepte, Produkte und Dienstleistungen, die neue Technologien in den Alltag einführen um die Lebensqualität für Menschen in allen Lebensphasen, vor allem im Alter, zu erhöhen. Ins [D]eutsche übersetzt steht AAL für Altersgerechte Assistenzsysteme für ein gesundes und unabhängiges Leben. [12]

Wie in Kapitel 1 bereits aufgeführt, birgt der demographische Wandel Herausforderungen für die Zukunft. Es ist abzusehen, dass immer mehr Menschen im Alter alleine leben werden und immer weniger jüngere Personen für ihre Pflege zur Verfügung stehen werden. Weiterhin zeigt eine Studie im *European Heart Journal* von 2019, dass das Alleinleben für Personen in den unteren und mittleren sozioökonomischen Schichten signifikant mit einem erhöhten Risiko, sowohl für die Gesamt mortalität, als auch für die kardiovaskuläre Mortalität verbunden ist. [14]

AAL zielt darauf alleine lebende Personen durch Technologien zu unterstützen, sodass sie möglichst lange und sicher in ihrem eigenen Zuhause leben können. Die Art der Unterstützung kann dabei unterschiedliche Formen annehmen. So werden in einer Broschüre des Bundesministerium für Familie, Senioren, Frauen und Jugend hilfreiche Technik aufgelistet, welche von einer Fenstergriffverlängerung über eine Herdüberwachung für Elektroherde bis hin zum Badewannenlift reicht. [6]

Aktuelle Forschung setzt sich unter anderem mit kognitiven Assistenzsystemen auseinander. Hier besteht das Ziel darin, Bewohnende bei der korrekten Ausführung von alltäglichen Aufgaben zu unterstützen. Insbesondere wird dabei darauf geachtet, dass die Systeme die Personen nur dann unterstützen, wenn Unterstützung benötigt wird und die Personen so weiterhin möglichst eigenständig arbeiten müssen. So werden in [3] Assistenzsysteme aufgezählt, die beispielsweise mittels eines Dialogsystems und Ja/Nein-Fragen verifizieren, ob ein Schritt einer Aufgabe ausgeführt wurde, bevor ein Hinweis für den nächsten Schritt gegeben wird. Ein weiteres Beispiel ist ein Kochassistent, welcher automatisch akustische oder visuelle Hinweise für jeden Schritt eines Rezepts gibt.

Ebenso wird an Systemen geforscht, die auf ein bestimmtes Krankheitsbild zugeschnitten sind. So ist DEM@HOME ein pervasives System das speziell auf die Bedürfnisse von Patienten mit Demenz abgestimmt ist. Eine Kombination aus Umgebungssensoren und tragbaren Sensoren liefert Daten, die einer semantischen Analyse oder Bildanalyse unterzogen werden. Zu den Sensoren zählen Tiefenkameras, Steckdosen-Sensoren, Tags für Objekte von besonderem Interesse (z.B. Medikamentdosen), Bewegungssensoren, Smartwatches und Druckmatten, welche unterhalb der Matratze installiert wird und so Daten über das Schlafverhalten sammelt. Diese Daten werden verarbeitet und genutzt, um Anomalien und Probleme im Verhalten der Patienten festzustellen und vorauszusagen. [2]

2.3 Sturzerkennung

Zwischen 28 und 35 % aller Personen im Alter von 65 bis 70 Jahren stürzen jährlich mindestens ein Mal. Dieser Prozentsatz vergrößert sich für Personen ab 70 Jahren auf 32-42 %. Das geht aus einem Bericht [32] der World Health Organization (WHO) von 2008 hervor.

Weiterhin schreibt die WHO, dass Stürze für 20-30 % der leichten bis schweren Verletzungen, für 10-15 % der Aufenthalte in der Notaufnahme und für 40 % aller Todesfälle durch Verletzungen verantwortlich sind. In Abbildung 2.2 wird der exponentielle Zusammenhang von Stürzen mit tödlichem Verlauf und dem Alter der stürzenden Person deutlich. Eine Sturzerkennung hat das Ziel, den Sturz einer Person ohne die Hilfe einer weiteren Person zu identifizieren. Wurde ein Sturz erfolgreich erkannt, können automatisiert Ersthelfer*innen informiert werden, die dann die gestürzte Person versorgen können.

Eine Sturzerkennung arbeitet dabei typischerweise mit Hilfe von Sensoren, die an der zu überwachenden Person angebracht sind oder von der zu überwachenden Person getragen werden oder mit Sensoren, die in der Umgebung angebracht sind oder mit Computer Vision.

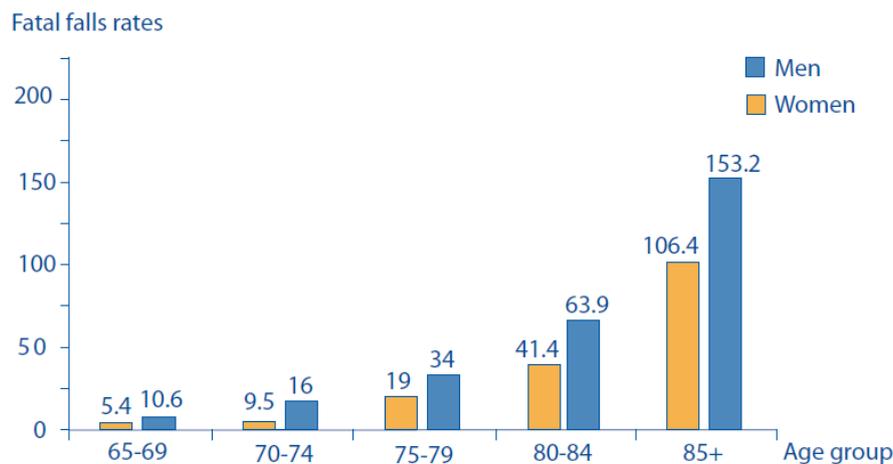


Abbildung 2.2: Tödliche Stürze in den USA 2001 nach Alter und Geschlecht. [32]

Sturzerkennung mit Hilfe von tragbaren Sensoren

Für eine Sturzerkennung mit Hilfe tragbarer Sensoren werden Kleidungsstücke oder kleine Geräte entwickelt, die die Person am Körper trägt. Häufig verwendet werden Beschleuni-

gungssensoren - die plötzlich auftretenden großen Beschleunigungswerte bei einem Sturz, bzw. bei dem Aufkommen auf dem Boden, können für die Sturzerkennung genutzt werden - und Luftdrucksensoren - hier kann die Änderung des Luftdrucks aufgrund einer Höhenänderung als Indikator für einen Sturz herangezogen werden - oder eine Kombination der beiden. [23]

Frühe Forschungen dieser Art gehen bis in das Jahr 1998 zurück. In [31] wird ein tragbares Gerät beschrieben, das zunächst über Beschleunigungssensoren einen Aufprall identifiziert und anschließend über einen Quecksilberschalter die Orientierung der Person überprüft. Je nach Orientierung wird anschließend darüber entschieden, ob die Person gestürzt ist oder nicht. Vorteile dieser Methode sind die Unabhängigkeit der Sturzerkennung von der momentanen Umgebung des Stürzenden und die geringen Kosten der verwendeten Sensoren. Ein Nachteil ist, dass die Person aktiv an das Tragen des Gerätes denken muss.

Sturzerkennung mit Hilfe von nicht-tragbaren Sensoren

In [29] werden mehrere Sturzerkennungssysteme vorgestellt, die auf nicht tragbaren Sensoren basieren. Hierzu werden Systeme gezählt, die über im Boden installierte Drucksensoren Vibrationen messen und damit Rückschlüsse auf einen Sturz ziehen.

Ebenso werden Lösungen aufgezeigt, die Mikrofone als Sensoren nutzen und auf Geräusche achten, die für einen Sturz typisch sind. Beide Systeme befreien die Person davon ein Gerät direkt am Körper tragen zu müssen, binden das System allerdings fest an den Ort, an dem die Sensoren installiert wurden. Zusätzlich stellten sich Systeme, die auf akustischen Signalen beruhen, als zu sensibel gegenüber Alltagsgeräuschen heraus.

Sturzerkennung mit Hilfe von Computer Vision

Für eine Sturzerkennung mittels Computer Vision gibt es unterschiedliche Ansätze.

In [25] werden die Bilder einer Deckenkamera dazu genutzt, eine Ellipse um die Person zu legen. Die Ellipse wird dabei durch ihren Mittelpunkt, ihre Orientierung, ihre Größe und ihre Exzentrizität beschrieben. Durch diese, möglichst grobe, Beschreibung der Person sollte die Erkennung möglichst unabhängig von der Haltung und Kleidung der Person gestaltet werden. Ferner wird der überwachte Raum in Inaktivitätszonen (Sessel, Sofa) und Eintrittszonen (Ein- und Ausgänge des Raums) unterteilt. Diese Informationen werden anschließend genutzt, um Inaktivität in Bereichen, die nicht als Inaktivitätszone

ausgezeichnet sind, als unübliche Aktivität zu klassifizieren.

Auch in [19] werden die Bilder einer Deckenkamera und Inaktivitätszonen genutzt. Im Gegensatz zu dem in [25] vorgestelltem System, werden hier simplere Algorithmen genutzt und es ist kein Trainingsdatensatz nötig. Das System nutzt den Umfang, den Durchmesser und die Geschwindigkeit des Mittelpunkts einer Bounding Box¹, welche um die Person konstruiert wird, nachdem der Hintergrund des Bildes entfernt wurde. Mittels dieser Werte wird die Haltung und das Aktivitätslevel einer Person bestimmt. Ein Sturz wird dann erkannt, wenn Schwellwerte für den Umfang und den Durchmesser der Bounding Box überschritten werden und die Aktivität außerhalb einer Inaktivitätszone zu gering ist. Da die Bounding Box von der Größe der Person abhängig ist, müssen Schwellwerte bezüglich des Durchmessers und des Umfangs vor eine Überwachung für die Größe der Person konfiguriert werden.

Ein weiterer Ansatz wird in [28] vorgestellt. Hier wird Pose Estimation (genauer in Kapitel 2.4) genutzt, um die relativen Positionen der Gelenke einer Person in einem Bild zu extrahieren. Diese Gelenkpositionen werden anschließend von einem neuronalen Netz ausgewertet und gegebenenfalls als Sturz klassifiziert.

2.4 Pose Estimation

Pose Estimation beschreibt die Extraktion von Körper-Keypoints² aus Bildern. Typische Keypoints sind dabei die Augen, die Nase, der Mund, die Ohren und Gelenke (siehe Beispiel in Abbildung 2.3). Weiterhin umfassen einige Bibliotheken für Pose Estimation auch fein aufgelöste Keypoints für die Hände und für Gesichtszüge. Generell können Pose Estimation Lösungen nach folgenden Kriterien unterschieden werden:

- 2D oder 3D Pose Estimation,
- Erkennung von Einzelpersonen oder Personengruppen,
- Top-Down oder Bottom-Up Ansatz.

In einer 2D Pose Estimation werden die Keypoints über Koordinaten in der horizontalen und vertikalen Achse des Bildes angegeben. Die 3D Pose Estimation fügt den Keypoints noch eine Koordinate hinzu, die die Tiefe des Punktes widerspiegelt. Der Unterschied ist in Abbildung 2.4 dargestellt.

¹Das kleinste mögliche Rechteck, welches alle Punkte eines Interessengebiets umschließt.

²Datenpunkt, der durch zwei bis drei Koordinaten spezifiziert wird.

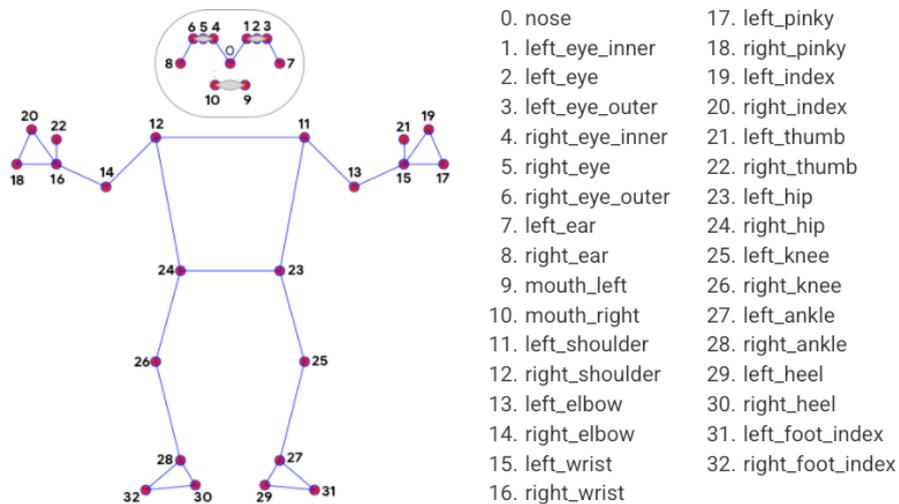


Abbildung 2.3: Die 33 Keypoints, die von MediaPipe extrahiert werden. [10]

Das Erkennen von Personengruppen stellt dabei eine besondere Herausforderung dar, da sich Personen im Bild überschneiden können, unterschiedlich groß dargestellt werden und miteinander interagieren können. [20]

Pose Estimation Lösungen für Personengruppen arbeiten dabei generell nach zwei unterschiedlichen Ansätzen: Top-Down Ansatz oder Bottom-Up Ansatz.

Top-Down Ansatz

Im Top-Down Ansatz wird zunächst das gesamte Bild analysiert und jeweils eine Bounding Box um erkannte Personen gelegt. Anschließend wird innerhalb dieser Bounding Box die Pose der Person bestimmt. Ein bekanntes Problem dieses Ansatzes ist das sogenannte *Early Commitment*, das bedeutet, dass in diesem Ansatz früh entschieden wird, ob eine Person erkannt wurde oder nicht. Wurde im ersten Schritt keine Person erkannt, so gibt es keine Möglichkeit nachträglich Keypoints zu extrahieren.

Ferner hängt der Rechenaufwand im Top-Down Ansatz proportional von der Anzahl der Personen im Bild ab, da für jede erkannte Person ein eigener Pose Estimator gestartet wird. [24]

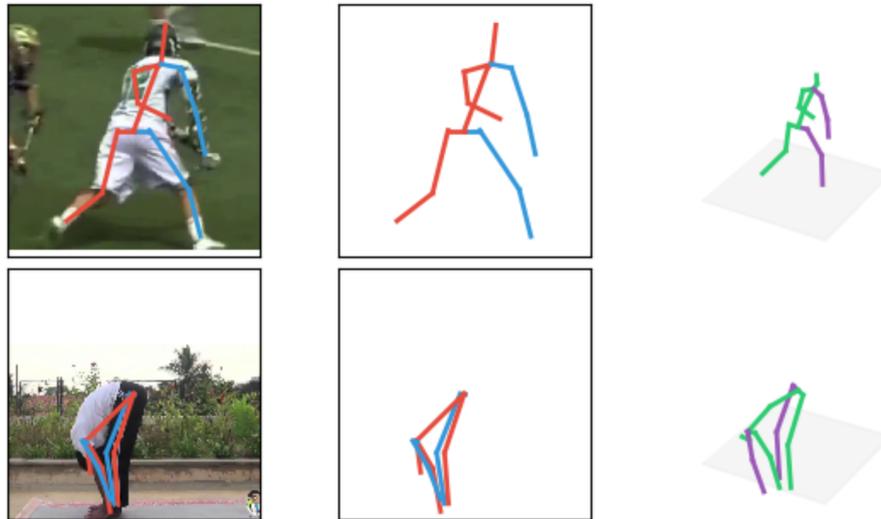


Abbildung 2.4: Vergleich 2D und 3D Pose Estimation. Links: Verwendetes Bild. Mitte: 2D Pose Estimation. Rechts: 3D Pose Estimation. [21]

Bottom-Up Ansatz

Der Bottom-Up Ansatz beginnt damit Keypoints unabhängig ihrer Zugehörigkeit zu einer Person zu identifizieren und ordnet diese anschließend einer Person zu. Der Vorteil dieses Ansatzes ist, dass das Problem des *Early Commitment* vermieden wird und es keinen proportionalen Zusammenhang zwischen der Anzahl der Personen und der Rechenzeit gibt. Allerdings kann es zu Ungenauigkeiten in der Zuordnung der Keypoints kommen, wenn sich mehrere Personen im Bild überschneiden. [20, 24]

Anwendungsgebiete

Pose Estimation findet in unterschiedlichen Gebieten Anwendung – [13] bietet eine Übersicht über typische Anwendungsgebiete. Aufgezählt werden hier unter anderem:

- Gestensteuerung von Computern,
- Unterstützung und Überprüfung von manuellen Schritten in der Fertigung,
- Assistenzsysteme, die über die Handstellung und Kopfposition der fahrenden Person, Rückschlüsse über deren Aufmerksamkeit ziehen,
- Übertragung von Bewegungen für Animationen in Filmen,

- Bewegungsanalyse in Sportarten,
- Videoüberwachungssysteme, die automatisch gefährliche Situationen erkennen können.

Die unterschiedlichen Anwendungsgebiete stellen auch unterschiedliche Anforderungen an die Qualität und Geschwindigkeit der Pose Estimation. So wurde beispielsweise *PifPaf* für den Einsatz in autonomen Fahrzeugen und Lieferrobotern entwickelt. Hier geht es vorrangig nicht um das Bestimmen einer menschlichen Pose, sondern viel mehr darum eine niedrigdimensionale und interpretierbare Repräsentation eines Menschen zu erstellen, um kritische Aktionen der Personen früh genug erkennen zu können. Mittels eines neu entwickelten Bottom-Up Ansatzes wurde *PifPaf* auf genau dieses Anwendungsgebiet getrimmt. [17]

3 Problemanalyse

Die Aufgabe ist es, ein System zu entwerfen und zu entwickeln, das mittels Pose Estimation menschliche Stürze in einem Zuhause erkennt. Um abzugrenzen, welche Anforderungen das System erfüllen muss, wird die Problemstellung zunächst in mehreren Schritten analysiert.

3.1 Definition „Sturz“

Die WHO definiert einen Sturz folgendermaßen:

Falls are commonly defined as “inadvertently coming to rest on the ground, floor or other lower level, excluding intentional change in position to rest in furniture, wall or other objects”. [32]

Für die Bewertung der Qualität des hier entwickelten Systems, wird die Definition für einen Sturz weiter quantifiziert. In einer Veröffentlichung über Sturzerkennungssysteme für ältere Personen von 2008 [33] werden drei Typen von Stürzen unterschieden:

1. Stürze aus einer liegenden Haltung,
2. Stürze aus einer sitzenden Haltung und
3. Stürze aus einer stehenden Haltung.

Dabei wird Stürzen aus einer stehenden Haltung eine Dauer von 1-2s und den Stürzen aus einer sitzenden oder liegenden Haltung eine Dauer von 1-3s zugeschrieben. Die kürzere Dauer für Stürze aus einer stehenden Haltung wird damit begründet, dass hier keine Möbelstücke, wie z.B. ein Bett oder ein Stuhl, vorhanden sind, die den Fall verlangsamen würden.

Eine Charakteristik, die die Stürze aus allen Haltungen aufweisen, ist, dass der Kopf der stürzenden Person währenddessen an Höhe verliert und zeitweise im freien Fall ist. [33]

Die oben aufgeführte Definition eines Sturzes wird nachfolgend um quantifizierbare Merkmale erweitert, sodass eine objektive Bewertung der entwickelten Lösung möglich ist. Stürze, die von dem System erkannt werden sollen, müssen folgende Merkmale aufweisen – für Sturzbewegungen, die eines oder mehrere dieser Merkmale nicht erfüllen, kann keine zuverlässige Erkennung garantiert werden:

- Der Sturz startet aus einer sitzenden oder stehenden Haltung,
- der Sturz dauert maximal 3 s,
- während des Sturzes verringert sich der Höhenunterschied zwischen dem Kopf und den Füßen,
- der Sturz endet in einer liegenden Haltung,
- der Sturz endet mit einem Höhenunterschied von weniger als 0,4 m zwischen dem Kopf und den Füßen und
- die gestürzte Person nimmt nicht innerhalb von 10 s wieder eine stehende Haltung ein.

Weiterhin wird vorausgesetzt, dass das System zu Beginn des Sturzes und direkt nach dem Sturz die Haltung der Person mittels Pose Estimation erkennen können muss. Ist dies nicht der Fall, trifft es keine Annahmen über mögliche Stürze.

3.2 Definition valider Umgebungen für die Sturzerkennung mittels Pose Estimation

Um den Erfolg einer korrekten Sturzerkennung mittels Pose Estimation zu erhöhen, müssen die Kamerabilder folgende Merkmale erfüllen:

- Die Auflösung des Kamerabilds muss mindestens $640 * 480$ Pixel betragen,
- das Seitenverhältnis des Kamerabilds muss $4 : 3$ betragen,
- die Kamera/das Video muss eine Bildfrequenz von 20 fps vorweisen,
- in einem Kamerabild darf sich maximal eine Person befinden,
- das Kamerabild darf nicht aus der Vogelperspektive auf die Person blicken,

- der Raum, aus dem das Kamerabild stammt, sollte mit mindestens $100 \frac{\text{lm}}{\text{m}^2}$ ¹ beleuchtet sein,
- die Kamerabilder müssen von einer stationär montierten Kamera aufgenommen werden.

Ferner ist es förderlich, wenn die Sicht der Kamera auf die Person durch möglichst wenig Objekte verdeckt wird.

3.3 Requirements Engineering

Aus den in Kapitel 3.1 und 3.2 erarbeiteten Definitionen lassen sich damit folgende Requirements, vgl. Tabelle 3.1, für das System ableiten.

Anhand dieser Requirements werden nachfolgend Lösungsansätze aufgestellt und diskutiert.

¹Typischer Wert für die Grundbeleuchtung in einem Wohnzimmer. [27]

Tabelle 3.1: Requirements einer Sturzerkennung mittels Pose Estimation.

Nummer	Requirement
R01	Das System muss eine Pose Estimation verwenden.
R02	Das System muss die Haltung einer Person zwischen stehend, sitzend und liegend unterscheiden können.
R03	Das System muss die Haltung einer Person als undefiniert erkennen, wenn unzureichende Daten für die Erkennung vorliegen.
R04	Das System muss Videostreams und Videoaufnahmen mit einem Seitenverhältnis von 4 : 3 und einer Auflösung von mindestens 640 * 480 Pixel bei einer Bildfrequenz von 20 fps verarbeiten können.
R05	Das System muss Kamerabilder auswerten können, die in einem Raum aufgenommen wurden, der mit mindestens $100 \frac{\text{lm}}{\text{m}^2}$ beleuchtet ist.
R06	Das System muss Stürze erkennen können, die aus einer stehenden oder sitzenden Haltung beginnen.
R07	Das System muss Stürze erkennen können, die in einer liegenden Haltung enden.
R08	Das System muss keine Stürze erkennen können, die länger als 3s dauern.
R09	Das System muss Stürze erkennen können, bei denen sich während des Sturzes der Höhenunterschied zwischen dem Kopf und den Füßen verringert.
R10	Das System muss Stürze erkennen können, die mit einem Höhenunterschied von weniger als 0,4 m zwischen dem Kopf und den Füßen enden.
R11	Das System soll ein Ereignis nicht als Sturz werten, wenn es für die gestürzte Person innerhalb von 10s erneut eine stehende Haltung erkennt.
R12	Das System soll keine Annahmen über Stürze treffen, wenn zu Beginn des Sturzes oder direkt nach dem Sturz die Haltung der Person undefiniert ist.
R13	Das System muss Kamerabilder, die mehr als eine Person enthalten, nicht korrekt klassifizieren können.
R14	Das System muss Kamerabilder, die aus der Vogelperspektive auf die Person blicken, nicht korrekt klassifizieren können.
R15	Das System muss Kamerabilder, die von einer nicht stationären Kamera erstellt wurden, nicht korrekt klassifizieren können.

4 Lösungsansätze

4.1 Lösungskonzepte

4.1.1 Auswahl Pose Estimation Bibliothek

Zunächst wird eine Pose Estimation Bibliothek ausgewählt, die folgend für die Bearbeitung der Aufgabenstellung genutzt werden soll. Zur Auswahl stehen zwei weitverbreitete Pose Estimation Bibliotheken: OpenPose (7300 Forks auf github.com [9]) und MediaPipe (4000 Forks auf github.com [15]). Tabelle 4.1 enthält einen Vergleich der technischen Daten der beiden Bibliotheken.

Da das System dafür gedacht ist, Personen zu unterstützen die alleine leben, ist die Anzahl der unterstützten Personen ein vernachlässigbares Merkmal. Auch die durchweg gleiche oder größere Anzahl an Keypoints für Körper, Hände und Gesicht in MediaPipe sind für diesen Anwendungsfall kein ausschlaggebendes Argument. Bemerkenswert ist die Vielzahl der angebotenen APIs bei MediaPipe. Gerade in einem Smart Home, in dem die verfügbare Hardware stark variieren kann, ist es sinnvoll flexibel zu sein.

Nach ersten Tests erwies sich MediaPipe mit seiner Python API als ein gut handhabbares Werkzeug für diese Aufgabe. Alle folgenden Entwicklungen des Systems arbeiten daher mit der MediaPipe Python API.

In den folgenden drei Unterkapiteln werden unterschiedliche Lösungsansätze diskutiert und auf ihre Eignung für die Lösung der Aufgabenstellung getestet.

4.1.2 Sturzerkennung über Beschleunigungswerte

Idee und Umsetzung

Der erste Ansatz beruht auf dem Gedanken, dass der Kopf einer Person bei einem Sturz zunächst in Richtung Boden beschleunigt wird und anschließend, wenn die Person sich

Tabelle 4.1: Vergleich von MediaPipe und OpenPose. [9, 15, 22]

	MediaPipe	OpenPose
Verwendeter Ansatz	Top-Down	Bottom-Up
Anzahl Personen	Einzelperson	Personengruppen
Körper-Keypoints	33	25
Hand-Keypoints	2 * 21	2 * 21
Gesicht-Keypoints	468	70
Verfügbare APIs	Android, iOS, Python, JavaScript, C++	C++, Python

auf dem Boden abstützt oder auf dem Boden aufkommt, abgebremst wird.

Um die Beschleunigung des Kopfes zu berechnen, wird in einem ersten Schritt überprüft, ob die Nase der Person sichtbar ist. Hierfür ermittelt MediaPipe pro Keypoint, neben den x-, y- und z-Koordinaten¹ auch, ein *Visibility*-Attribut², das angibt, wie sicher die Position des Keypoints bestimmt werden konnte.

Für diesen Ansatz wird eine minimale *Visibility* von 0,8 erwartet. Wird dieser Schwellwert nicht erreicht, wird auch keine Geschwindigkeits- und Beschleunigungsberechnung für den Frame ausgeführt. Ist die *Visibility* des Keypoints der Nase größer als 0,8, werden die ersten elf Keypoints³ extrahiert und zurückgegeben.

Anschließend werden die x-, y- und z-Koordinaten der Kopf-Keypoints arithmetisch gemittelt und mit einem Zeitstempel versehen. Dieser Zeitstempel wird für die nachfolgende Geschwindigkeitsberechnung benötigt. Das so ermittelte Quadrupel wird in einer Queue mit fester Länge abgespeichert.⁴

Mit Hilfe der abgespeicherten Werte des momentanen Frames und der vergangenen Frames, wird die Geschwindigkeit des Kopfes in der x-, y- und z-Achse berechnet. Wiederum kann in einem nächsten Schritt aus der eben ermittelten Geschwindigkeit und der Geschwindigkeit des letzten Frames die Beschleunigung ermittelt werden.

Zur Visualisierung und Hilfe beim Debugging wurde die Matplotlib für Python genutzt.

¹Die x-Achse geht in diesem Fall von links nach rechts, die y-Achse von oben nach unten und die z-Achse vom Mittelpunkt der Hüfte zur Kameralinse.

²Der Wert des *Visibility*-Attributs liegt zwischen 0,0 und 1,0. Wobei 1,0 für eine hohe Sichtbarkeit des Keypoints steht.

³Die ersten elf Keypoints umfassen alle Punkte, die von MediaPipe mit dem Kopf in Verbindung gebracht werden, vgl. Abbildung 2.3.

⁴Ist die Queue voll und es werden neue Werte hinzugefügt, werden die Werte, die am längsten in der Queue gespeichert waren, gelöscht.

In Abbildung 4.1 sind die Beschleunigungswerte der letzten zehn Frames für die x-, y- und z-Achse dargestellt. Der Plot wurde für einen Sturz aufgezeichnet, der in Richtung der Kamera ausgeführt wurde. Gut zu erkennen ist, wie negative Werte für die Beschleunigung in y-Richtung (der Kopf beschleunigt Richtung Boden) ermittelt wurde. Der Graph der z-Richtung kann in diesem Beispiel vernachlässigt werden, da MediaPipe, durch den Sturz in Richtung Kamera, Probleme hatte die Position der Hüfte korrekt zu bestimmen.

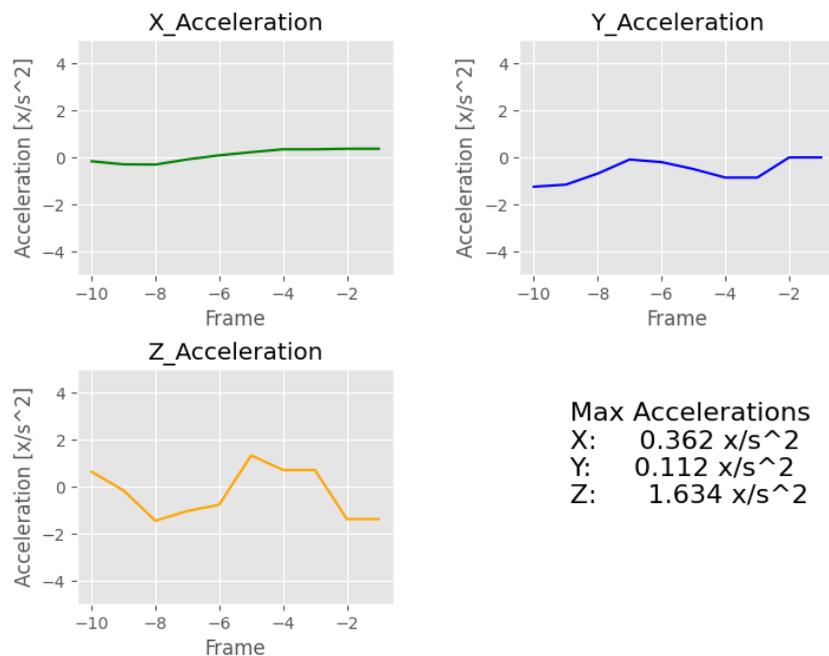


Abbildung 4.1: Visualisierung der Beschleunigungswerte während einem Sturz mittels Matplotlib.

Bewertung des Ansatzes

Die so ermittelten Geschwindigkeits- und Beschleunigungswerte sind als Pseudowerte anzusehen.

Das Problem an diesem Ansatz ist, dass die Koordinaten, mit denen die Berechnung ausgeführt wird, relative Koordinaten sind und damit keine echten Distanzen berechnet werden können. Die x-Koordinate beginnt an dem linken Bildrand bei 0,0 und endet an dem rechten Bildrand mit 1,0. Analog beginnt die y-Koordinate an dem oberen Bildrand

bei 0,0 und endet am unteren Bildrand mit 1,0.

Das bedeutet, dass die Größe der ermittelten Pseudobeschleunigung davon abhängig ist, wie groß die Distanz zwischen Kameralinse und der Person ist. Somit kann kein allgemeingültiger Schwellwert festgelegt werden, der, wenn er überschritten wird, einen Sturz signalisieren würde.

4.1.3 Sturzerkennung über ein neuronales Netz

Idee und Umsetzung

In einem zweiten Ansatz soll die Sturzerkennung nicht von Beschleunigungswerten abhängig sein. Hier wird ein möglicher Sturz über eine bestimmte Abfolge von festgelegten Posen der Person identifiziert.

Um diesen Ansatz verfolgen zu können, muss bestimmt werden können, in welcher Haltung sich die Person befindet. Dabei soll zwischen folgenden Haltungen unterschieden werden:

- stehend,
- sitzend,
- gebeugt und
- liegend.

Für die Klassifizierung der Pose soll in diesem Ansatz ein künstliches Neuronales Netz (NN) genutzt werden. Da der Fokus dieser Arbeit nicht auf dem Erstellen eines solchen Netzes liegt, wird hierfür ein NN ausgewählt, welches seine Leistungsfähigkeit in Computer Vision Aufgaben bereits bewiesen hat, das ResNet50. [30, 16]

Um das ResNet50 an die Aufgabenstellung anzupassen, wurde es mit einem Datensatz [18] nachtrainiert, der jeweils 1200 Bilder der oben genannten Posen, als Silhouette dargestellt, enthält, vgl. Abbildung 4.2. Um die Bilder der Kamera den Trainingsdaten anzupassen, wurde eine weitere Funktion von MediaPipe genutzt: Die Körper Segmentierung. [11]

Hierfür wird im Konstruktor des MediaPipe Pose Objekts die Flag `enable_segmentation = True` gesetzt. Anschließend berechnet MediaPipe neben den Keypoints auch eine Maske, die als Array, mit denselben Maßen des Kamerabilds, abgebildet wird. Die

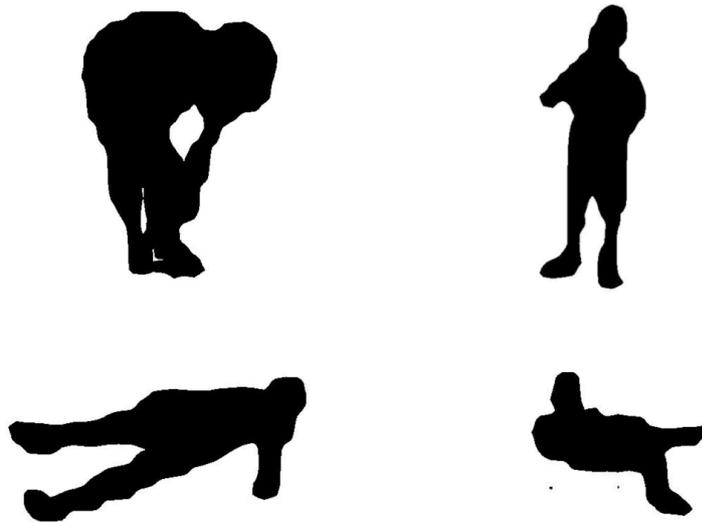


Abbildung 4.2: Vier Bilder aus dem Datensatz, der für das nachträgliche Training des ResNet50 benutzt wurde. Links oben: Gebeugt, rechts oben: Stehend, links unten: Liegend, rechts unten: Sitzend. [18]

Maske enthält Werte zwischen 0,0 und 1,0, wobei der Wert 0,0 dafür steht, dass an diesem Pixel des Bildes nicht der Körper der Person vermutet wird.

Mit Hilfe dieser Maske können Bilder erzeugt werden, die, genauso wie die Trainingsdaten, nur die Silhouette der Person enthalten, vgl. Abbildung 4.3.

Bewertung des Ansatzes

Für die Bewertung des Ansatzes werden 84 Bilder erzeugt, die eine Person im Sitzen zeigen, und 47 Bilder, die eine Person im Stehen zeigen. Alle Bilder sind mittels Segmentierung zu einem Binärbild aufbereitet, wie in Abbildung 4.3 dargestellt.

Diese Bilder werden an das ResNet50 übergeben. Dabei wird neben der Klassifizierung auch die Zeit geloggt, die für die Klassifizierung benötigt wird.

Von den 47 Testbildern mit stehender Person werden 31 korrekt klassifiziert. Neun werden als gebeugt, sieben als liegend und null als sitzend klassifiziert. Das ergibt eine Trefferquote von 65,96 %.

Von den 84 Testbildern mit sitzender Person werden 18 korrekt klassifiziert. 15 werden als gebeugt, vier als liegend und 47 als stehend klassifiziert. Das ergibt eine Trefferquote von 21,43 %.



Abbildung 4.3: Zwei Bilder die mittels Segmentierung aufbereitet wurden und so an das ResNet50 übergeben werden. Links: Sitzend, rechts: Stehend.

Neben der unzureichenden Trefferquote der Klassifizierung, welche gegebenenfalls durch mehr Trainingsdaten verbessert werden könnte, stellt sich die Zeit, die für eine Klassifizierung benötigt wird, als schwerwiegenderes Problem dieses Ansatzes heraus. Die durchschnittliche Zeit, die für eine Klassifizierung eines Bildes benötigt wird, liegt bei 115 ms. Das bedeutet, dass mit diesem Ansatz eine maximale Bildfrequenz von 8,7 fps erzielt werden kann. Somit wäre Requirement R04 nicht erfüllt.

4.1.4 Sturzerkennung über Pose aus Keypoints

Idee und Umsetzung

Das dritte Lösungskonzept vereint Ideen aus den Ansätzen, die in Kapitel 4.1.2 und 4.1.3 beschrieben sind.

Ein Sturz wird hier über eine Kombination aus der momentanen Pose und den vergangenen Posen der Person und einem Faktor, welcher für den Höhenunterschied zwischen den Füßen und dem Kopf der Person steht, ermittelt.

Für die Sturzerkennung soll zwischen einer stehenden, sitzenden und liegenden Haltung unterschieden werden. Außerdem soll das System Posen, die auf Grund von fehlenden Keypoints, nicht identifiziert werden können, als undefiniert ausweisen. Die momentane Pose wird dabei über Winkel zwischen bestimmten Keypoints, das Seitenverhältnis einer Bounding-Box, die zwischen dem niedrigsten Hüft-Keypoint und dem höchsten Kopf-Keypoint aufgespannt wird, und einem Faktor und der von nun an Head-To-Toe genannt wird, bestimmt.

Die genaue Funktionsweise des Pose-Klassifikators ist in Kapitel 4.2.1 beschrieben. Die genaue Funktionsweise der Sturzerkennung ist in Kapitel 4.2.2 beschrieben.

Bewertung des Ansatzes

Da sich dieser Ansatz als die vielversprechendste erwiesen hat, wird er ausführlich in den Kapiteln 5 und 6 diskutiert.

4.2 Systementwurf

Für die Implementierung des in Kapitel 4.1.4 beschriebenen Ansatzes wird ein System entworfen, das den Anforderungen genügt.

Zunächst wird eine Komponente benötigt, die die Pose der Person anhand der, von MediaPipe ausgegebenen, Pose Estimation berechnen kann. Diesem Zweck dient das Package `pose_classifier`. Die Klasse `PoseClassifier` berechnet die erforderlichen Winkel, den Höhenunterschied zwischen Kopf und Knöchel und die Bounding Box, die den Oberkörper und den Kopf umschließt. Pro Frame wird so, wenn die Pose Estimation erfolgreich durchgeführt werden konnte, entschieden, ob die Pose der Person stehend, sitzend, liegend oder undefiniert ist.

Die zweite entscheidende Komponente ist die Sturzerkennung. Sie ist in dem Package `fall_detection` enthalten. Die Klasse `FallDetector` entscheidet anhand der Werte, die im Pose-Klassifikator berechnet werden, ob ein Sturz vorgefallen ist oder nicht. Alle Packages und Klassen sind in Abbildung 4.4 dargestellt. Zusätzlich werden in dem Package `util` Klassen für die Visualisierung und das Logging bereitgestellt. Das Package `common` enthält Enumerationen, die sowohl von der Sturzerkennung, als auch von dem Pose-Klassifikator benötigt werden.

4.2.1 Funktionsweise Pose-Klassifikator

Der Pose-Klassifikator bestimmt anhand von Winkeln zwischen Keypoints, dem Head-To-Toe und dem Seitenverhältnis einer Bounding-Box die Haltung der Person. Der Ablauf ist in einem Aktivitätsdiagramm in Abbildung 4.5 dargestellt.

Zunächst werden in der Funktion `calculate_joint_angles()` die relevanten Winkel zwischen den Keypoints berechnet. Zum Zeitpunkt des Schreibens wird nur der Winkel

zwischen Schulter und Knie ausgewertet. Hierfür wird der Winkel zwischen linkem Knie und linker Schulter, sowie zwischen rechtem Knie und rechter Schulter berechnet. Das arithmetische Mittel der beiden ergibt den gesuchten Winkel.

Anschließend wird Head-To-Toe berechnet. Der Faktor spiegelt die Höhendifferenz zwischen Kopf und Knöchel wider. Für die Berechnung wird eine weitere Liste von Keypoints genutzt, die von MediaPipe ermittelt wird – die `pose_world_landmarks` [10], ab jetzt World-Keypoints genannt. Diese World-Keypoints geben nicht die relative Position eines Keypoints innerhalb des Bildes an, sondern haben ihren Koordinatenursprung in der Mitte der Hüfte. Die x-, y- und z-Koordinate dieser World-Keypoints beschreiben dann den Abstand⁵ des jeweiligen Keypoints zu der Mitte der Hüfte.

Um mithilfe dieser World-Keypoints den Höhenunterschied zwischen Kopf und Knöchel zu identifizieren, wird zunächst die y-Komponenten der Nase und des linken sowie rechten Knöchels ausgelesen. Anschließend wird jeweils die Höhendifferenz zwischen Nase und linkem Knöchel und Nase und rechtem Knöchel berechnet. Die Differenz mit dem kleineren Zahlenwert wird als Head-To-Toe zurückgegeben.

Mit allen benötigten Werten wird nun, wie in Abbildung 4.5 dargestellt, entschieden, ob die Haltung der Person definiert werden kann und falls ja, in welcher Haltung sich die Person befindet. Die Entscheidung folgt dabei folgender Logik:

- Die Person liegt, wenn ein Wert für Head-To-Toe berechnet werden konnte und dieser Wert kleiner als der parametrisierbare Schwellwert `head_to_toe_threshold` ist.
- Die Person steht, wenn der Winkel zwischen Oberkörper und Oberschenkel größer als 120° ⁶ und das Seitenverhältnis der Bounding-Box $> 1,5$ ist.
- Die Person sitzt, wenn der Winkel zwischen Oberkörper und Oberschenkel zwischen 120° und 60° liegt.
- Die Person hat eine undefinierte Haltung, wenn keine der oben aufgeführten Haltungen zugeordnet werden konnte.

Neben dem Bestimmen der Pose zählt der Pose-Klassifikator die aufeinanderfolgenden Frames, in denen eine stehende Person erkannt wurde. Dieser Zähler wird nur zurückgesetzt, wenn eine sitzende oder liegende Haltung erkannt wird, eine undefinierte Haltung verändert den Zähler nicht.

⁵MediaPipe gibt an, dass dieser Abstand die Einheit Meter hat.

⁶Eine aufrecht stehende Person hätte einen Winkel zwischen Oberkörper und Oberschenkel von $\sim 180^\circ$.

Der Zähler wird im weiteren Verlauf dafür genutzt, ein erneutes Aufstehen der gestürzten Person zu erkennen und eine latente Sturzmeldung gegebenenfalls abzurechnen.

4.2.2 Funktionsweise Sturzerkennung

Die Sturzerkennung entscheidet anhand der durch den Pose-Klassifikator berechneten Werte, ob ein Sturz vorgefallen ist oder nicht. Der Ablauf der Auswertung ist im Aktivitätsdiagramm in Abbildung 4.6 dargestellt.

Zunächst wird geprüft, ob zuvor ein latenter Sturz ausgewiesen wurde. Ist dies der Fall, wird in `check_if_latent_fall_active()` geprüft, ob der latente Sturz länger als 10s zurückliegt und in diesem Zeitraum die Haltung der Person nicht erneut als stehend eingeordnet wurde. Treffen diese Bedingungen zu, so gibt die Sturzerkennung eine Sturzmeldung aus.

Wie in Requirement R07 beschrieben, sollen nur Stürze erkannt werden, die in einer liegenden Haltung enden. Wurde die Pose als liegend identifiziert, so ist der nächste Schritt den Start des Sturzes zu bestimmen. Hierfür werden in der Funktion `get_last_head_to_toe_before_fall()` alle gebufferten Pose-Tuples durchlaufen. Dabei wird die Differenz des Head-To-Toe eines Tuples mit dem seines Vorgängers berechnet. Ist die Differenz größer als der Schwellenwert, der sich prozentual aus dem Head-To-Toe des aktuelleren Tuples berechnet, so gilt diese Pose als Startpunkt für den Sturz.

Nachfolgend wird geprüft, ob die Pose, zu dem ermittelten Startpunkt des Sturzes, sitzend war. Wenn dies der Fall ist, wird die minimale Sturztiefe um ein Drittel verringert. Das hat den Hintergrund, dass bei einer sitzenden Person davon ausgegangen wird, dass sie durch ihre bereits tiefere Position nicht so tief stürzen kann wie eine stehende Person. Die Sturztiefe berechnet sich aus der Differenz des aktuellen Head-To-Toe und des Head-To-Toe der Startposition. Erreicht diese Sturztiefe nicht einen festgelegten Schwellenwert, so wird die Bewegung der Person nicht als Sturz identifiziert.

Im letzten Schritt wird geprüft, ob der Sturz innerhalb eines festgelegten Zeitfensters von statten ging. So soll vermieden werden, dass beispielsweise ein langsames Hinlegen der Person als Sturz erkannt wird. Lief der Sturz schnell genug ab, so wird die Situation als latenter Sturz klassifiziert und der aktuelle Zeitstempel zwischengespeichert. Ein latenter Sturz wird nach oben beschriebener Logik zu einer Sturzmeldung.

Alle Schwellenwerte werden in einer Konfigurationsdatei eingestellt. Über eine optimale Konfiguration wird in Kapitel 5.3 diskutiert.

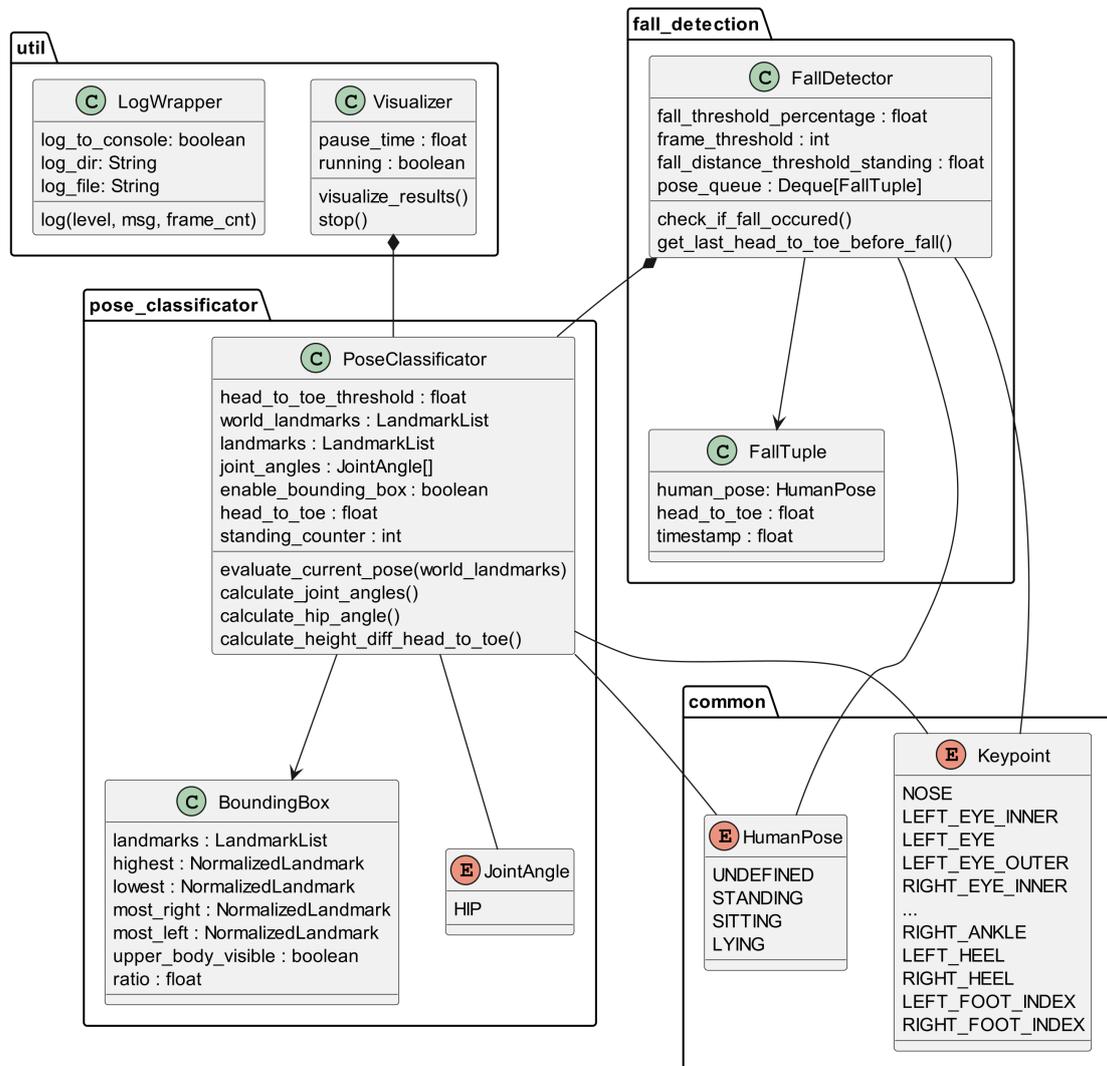


Abbildung 4.4: Klassendiagramm der Sturzerkennung.

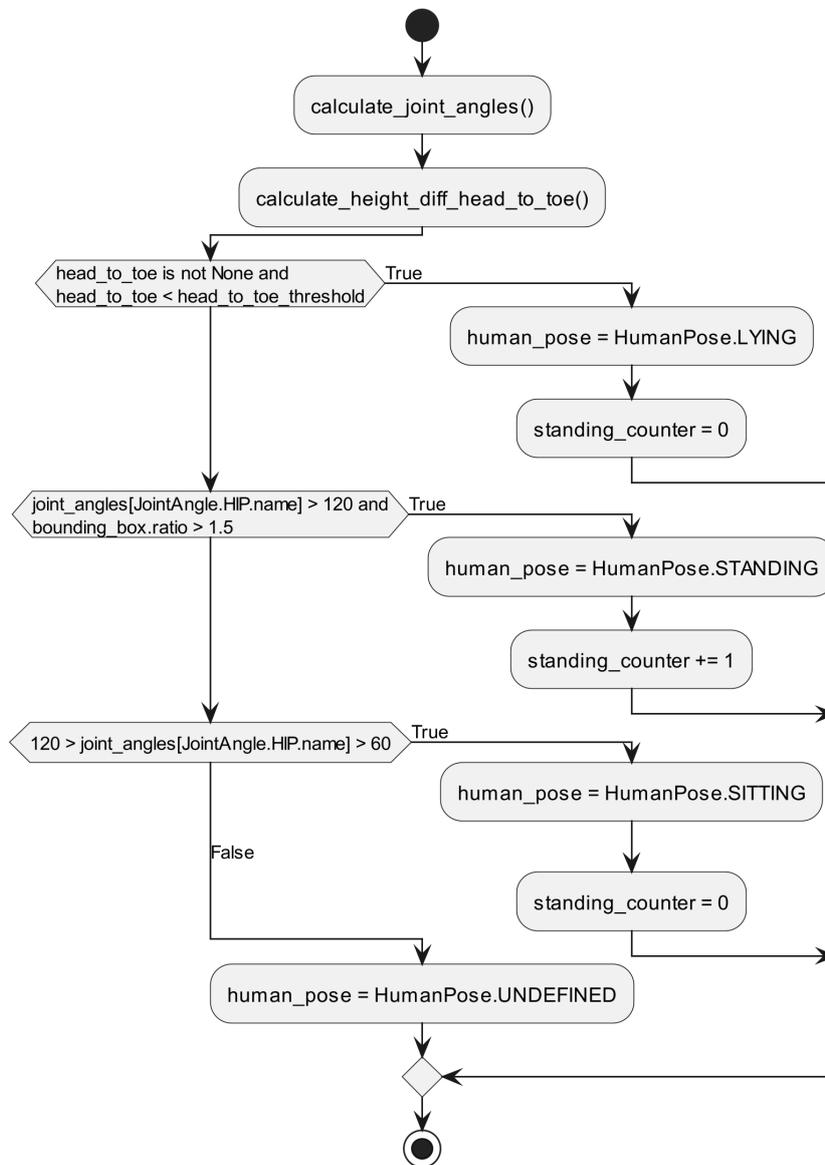


Abbildung 4.5: Aktivitätsdiagramm des Pose-Klassifikators.

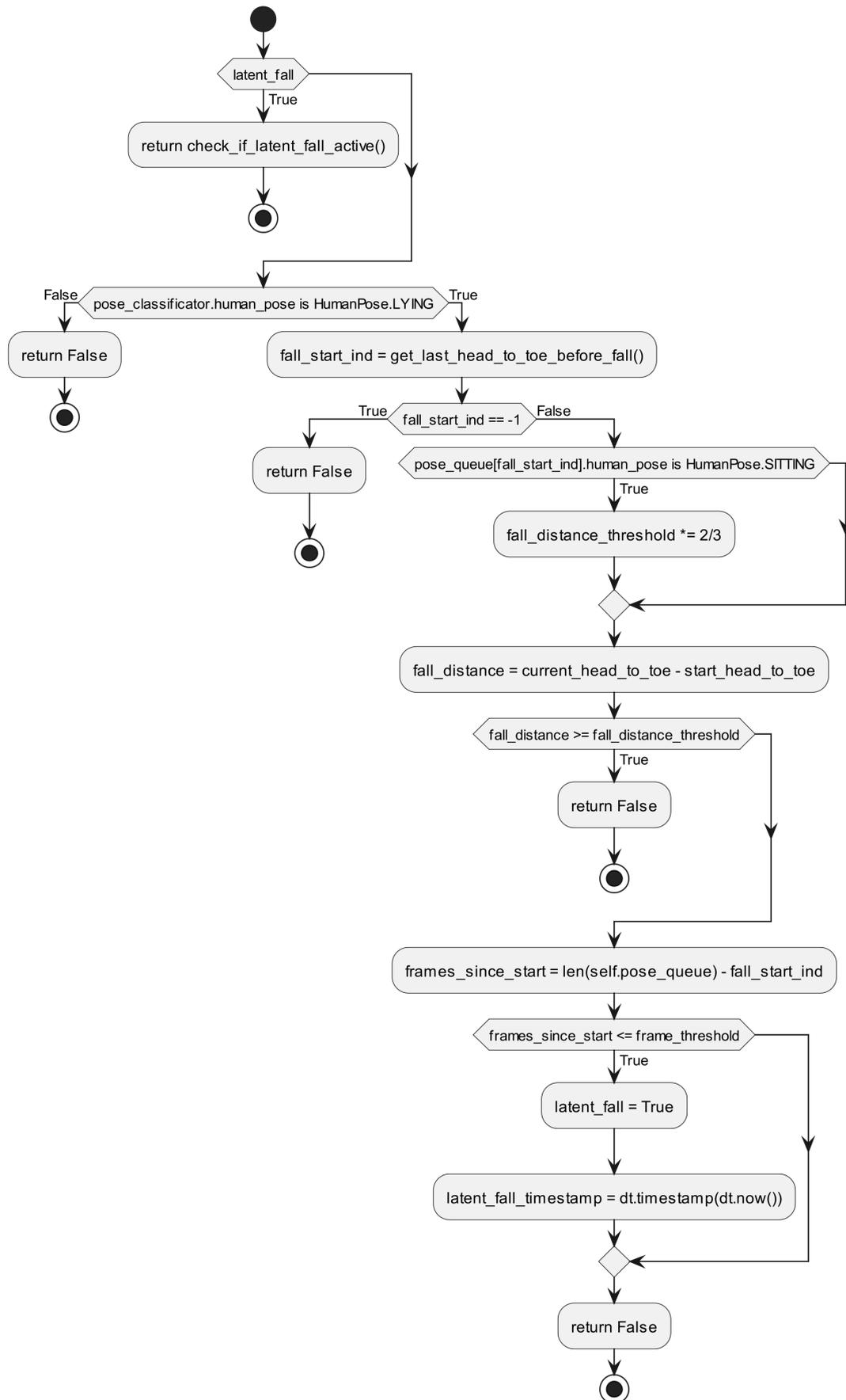


Abbildung 4.6: Aktivitätsdiagramm der Sturzerkennung.

5 Ergebnisse

Nachfolgend werden Aufnahmen mit unterschiedlichen Rahmenbedingungen untersucht. Die Aufnahme gliedern sich einerseits in Aufnahmen von alltäglichen Bewegungen einer Person, ab jetzt Activity of Daily Living (ADL) genannt, und in Stürze einer Person. In Kapitel 5.2 werden diese Situationen von einer einzelnen stationären Webcam aufgezeichnet. In Kapitel 5.4 werden zusätzlich Aufnahmen mehrerer Domkameras in einer Smart Home Umgebung genutzt. Abschließend werden in Kapitel 5.5 Aufnahmen Dritter genutzt, die stark schwankende Umgebungsbedingungen aufweisen. Vorangestellt ist in Kapitel 5.1 eine Übersicht über die verwendete Konfiguration von MediaPipe.

5.1 Konfiguration MediaPipe

Aufgrund der verwendeten Logik für die Sturzerkennung, ist die Qualität der Sturzerkennung stark von den ermittelten Keypoints abhängig. MediaPipe bietet insgesamt vier Konfigurationsparameter, die die Qualität der ermittelten Keypoints beeinflussen können.

`static_image_model`

Wird dieser Parameter auf `False` gesetzt, so behandelt MediaPipe die übergebenen Bilder als ein zusammenhängendes Video. MediaPipe ermittelt dabei innerhalb der ersten Frames die am deutlichsten erkennbare Person. Wurde hierbei eine Person detektiert, so werden in den nachfolgenden Bildern die Keypoints mit Hilfe der zuvor ermittelten Position der Person weitergeführt. Kann MediaPipe die Keypoints nicht länger weiterführen, wird erneut das gesamte Bild nach einer Person abgesucht.

Ist dieser Parameter auf `False` gesetzt, verringert sich die Rechenzeit, die für die Ermittlung der Keypoints benötigt wird, nachdem eine Person erkannt wurde. Der Parameter

ist standardmäßig auf `False` gesetzt und sollte nur auf `True` gesetzt werden, wenn Bilder verarbeitet werden, die keinen gemeinsamen Kontext haben.

`min_detection_confidence`

Über diesen Parameter kann konfiguriert werden wie sicher sich MediaPipe bei der Erkennung einer Person sein muss. Dabei können Werte zwischen 0,0 und 1,0 eingestellt werden, wobei höhere Werte bedeuten, dass MediaPipe nur für Personen eine Pose Estimation durchführt, bei denen es die Wahrscheinlichkeit dafür, dass es sich tatsächlich um eine Person handelt höher ist. Standardmäßig wird dieser Parameter mit dem Wert 0,5 versehen.

`min_tracking_confidence`

Dieser Parameter steht im Zusammenhang mit dem bildübergreifendem Tracking der Keypoints. Auch hier können Werte zwischen 0,0 und 1,0 eingestellt werden. Niedrigere Werte haben dabei den Effekt, dass MediaPipe seltener das gesamte Bild nach einer Person absucht und länger an den zuvor ermittelten Keypoints festhält. Dieser Parameter ist nur relevant, wenn `static_image_mode = False` ist und ist standardmäßig auf den Wert 0,5 gesetzt.

`model_complexity`

Hierüber kann definiert werden, welches Keypoint Model als Grundlage für die Pose Estimation genutzt werden soll. Gültige Eingaben für diesen Parameter sind 0, 1 und 2, wobei höhere Zahlenwerte für ein komplexeres Model stehen. Standardmäßig ist dieser Parameter auf 1 gestellt. Wird ein Wert ungleich 1 festgelegt, lädt sich MediaPipe das passende Model herunter.¹

¹Zum Zeitpunkt dieser Arbeit war ein Umstellen der Model Komplexität nicht möglich, da die, in den Sourcen hinterlegten, Download-Links der Models ungültig waren.

Verwendete Konfiguration

Für alle nachfolgenden Auswertungen wird die in Tabelle 5.1 beschriebene Konfiguration verwendet.

Der Parameter `static_image_model` wird für die folgenden Untersuchungen auf `False` gesetzt, da es sich bei den verwendeten Bildern um zusammenhängende Videodateien handelt. Tests, in denen `static_image_model = True` konfiguriert wurde, zeigen, dass die Keypoints deutlich weniger zuverlässig berechnet werden konnten und, dass die Zeit, die MediaPipe für das Extrahieren der Keypoints benötigt im Durchschnitt zweimal² so groß ist, wie, wenn der Parameter auf `False` gesetzt wird.

Weiterhin haben Tests ergeben, dass das Erhöhen der `min_detection_confidence` das Auftreten von Artefakten³ verhindern kann. Ein solches Artefakt ist in Abbildung 5.1 dargestellt. Hier missinterpretiert MediaPipe einen Bereich des Bildes als Person. Gleichzeitig kann ein erhöhter Wert dafür sorgen, dass tatsächliche Personen spät oder gar nicht von MediaPipe erkannt werden. Der eingestellte Wert hat sich als sinnvoller Kompromiss erwiesen.

Durch das Verringern der `min_tracking_confidence` kann erreicht werden, dass MediaPipe, wenn es einmal eine Person identifiziert hat, länger Keypoints für diese Person extrahiert, auch wenn die Person eine Haltung einnimmt, die sonst nicht von MediaPipe erkannt werden würde. In Abbildung 5.2 wird dieses Verhalten deutlich: Die Person entfernte sich von der Kamera und stürzte. Da MediaPipe in den vorangegangenen Bildern Keypoints zu dieser Person berechnen konnte, hält es bei einer geringen `min_tracking_confidence` an der Person fest.

Problematisch wird eine zu geringe `min_tracking_confidence`, wenn ein Artefakt auftritt, da MediaPipe dann sehr lange für diesen, fälschlicherweise als Person ausgewiesenen, Bereich Keypoints berechnet und eine erneute Personenerkennung hinausgezögert wird. Nachfolgend wird der Standardwert für diesen Parameter genutzt.

5.2 Auswertung von Webcam-Aufnahmen

Dieser Datensatz wird in erster Linie dafür genutzt, die grundlegende Funktionalität der Sturzerkennung zu entwickeln.

²Mit `static_image_model = False` benötigt MediaPipe im Durchschnitt 25 ms für das Berechnen der Keypoints.

³Keypoints, die Stellen des Bildes zugeschrieben werden an denen sich keine Person befindet.

Tabelle 5.1: Verwendete Konfiguration für MediaPipe.

Parameter	Wert
static_image_model	false
min_detection_confidence	0.8
min_tracking_confidence	0.5
model_complexity	1

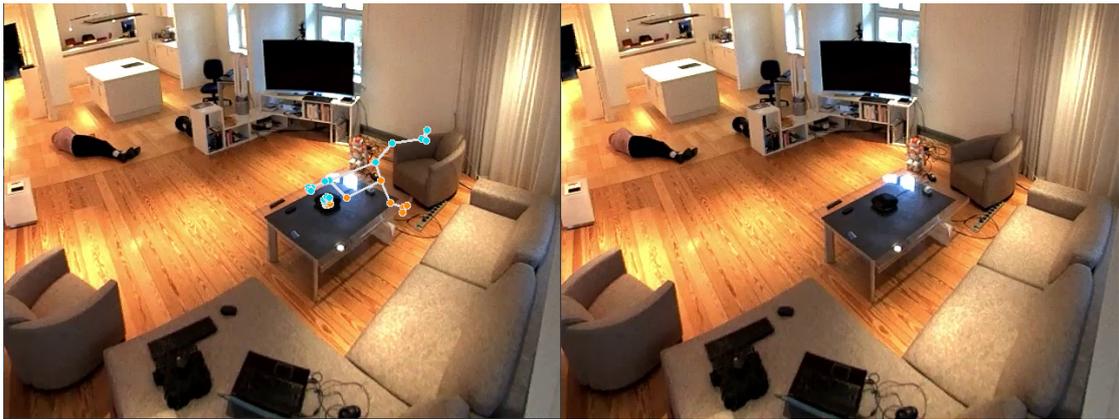


Abbildung 5.1: Einfluss des Parameters `min_detection_confidence`.
 Links: `min_detection_confidence = 0.5`,
 Rechts: `min_detection_confidence = 0.8`.

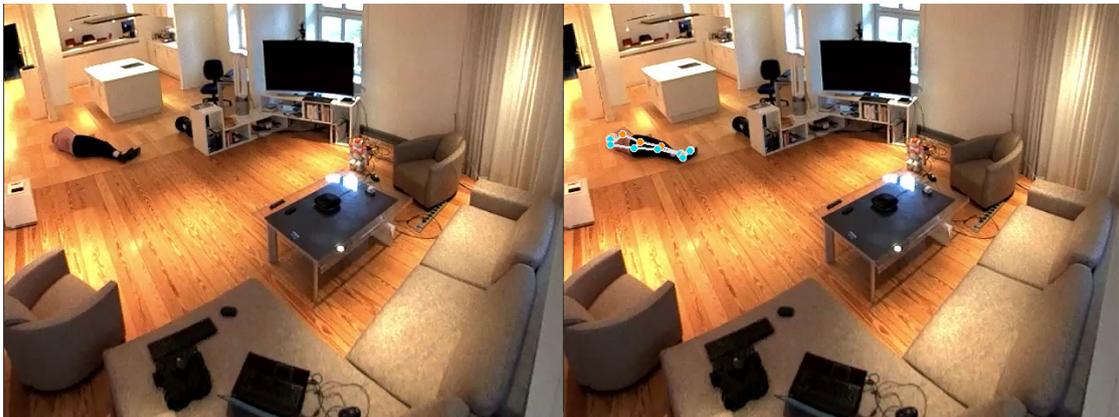


Abbildung 5.2: Einfluss des Parameters `min_tracking_confidence`.
 Links: `min_tracking_confidence = 0.5`,
 Rechts: `min_tracking_confidence = 0.0`.

Tabelle 5.2: Übersicht über die Webcam-Aufnahmen.

ADL		Sturz
Person steht im Raum.		Person stürzt nach hinten/von der Kamera weg.
Person setzt sich an den Tisch auf einen Stuhl.		Person stürzt nach vorne/zur Kamera hin (Zwei Aufnahmen).
Person setzt sich auf den Boden.		Person stürzt seitlich.
Person beugt sich nach vorne.		Person stürzt und stützt den Sturz an einem Stuhl ab.
Person hebt einen Gegenstand vom Boden auf.		
Person bindet sich die Schuhe.		
Person legt sich langsam flach auf den Boden.		
Person macht einen Crunch.		
Person macht Liegestütze.		

5.2.1 Beschreibung der Aufnahmen

Diese Aufnahmen wurden mit einer einzelnen Webcam aufgenommen. Die Kamera war stets am selben Ort stationiert und alle Aufnahmen wurden innerhalb eines kurzen Zeitfenster aufgenommen, sodass die Lichtverhältnisse über den Aufnahmezeitraum nahezu konstant waren. Die Aufnahmen haben eine Auflösung von $640 * 480$ Pixeln bei einer Bildfrequenz von 25 fps.

Dieser Datensatz enthält insgesamt fünf Stürze, die auf unterschiedliche Weise durchgeführt wurden, und neun ADL. Eine Übersicht über die abgebildeten Situationen bietet Tabelle 5.2.

5.2.2 Auswertung

Die Schwellwerte werden folgendermaßen festgelegt: Die minimale Fallhöhe muss 0,4 m und der prozentuale Höhenunterschied zwischen zwei Frames, der den Start eines Sturzes markiert, muss -15% betragen. Diese Werte haben sich nach kurzen Testläufen als sinnvoll erwiesen. Eine Optimierung der Schwellwerte erfolgt mit den Aufnahmen aus Kapitel 5.4 in Kapitel 5.3.

Activity of Daily Living-Aufnahmen

Das Tracking der Keypoints ist in fast allen Aufnahmen akkurat. Die Ausnahme bildet die ADL-Aufnahme in der sich die Person nach vorne beugt. Hier missinterpretiert MediaPipe für einige Frames die Position des linken Fußes erheblich, vgl. Abbildung 5.3. Der Fuß wird so nah am Kopf der Person vermutet, dass der Schwellwert für eine liegende Position unterschritten wird. Da die Veränderung der Fußposition sprunghaft erfolgt, wird die Bewegung fälschlicherweise als Sturz bewertet.

Alle weiteren ADL-Aufnahmen werden nicht als Sturz klassifiziert.

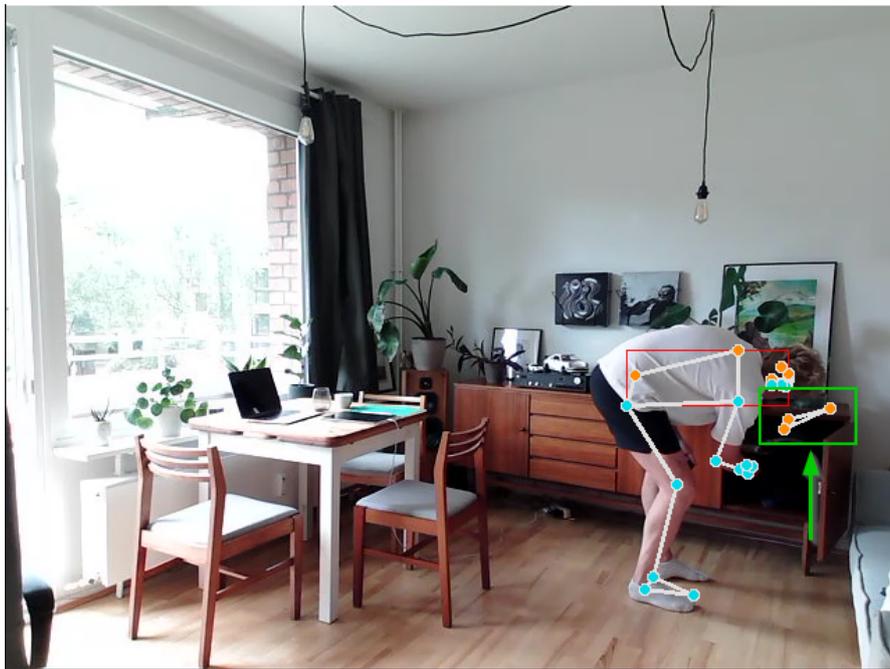


Abbildung 5.3: Die Position des linken Fußes (grün markiert) wird missinterpretiert. Die Situation wird als Sturz gewertet.

Sturz-Aufnahmen

Bei den Stürzen bildet die Aufnahme des gestützten Sturzes die Ausnahme. Dieser Sturz läuft durch das Abstützen so langsam ab, dass er nicht als Sturz erkannt wird. Da die Zeit, die, vom ersten Absenken des Kopfes bis zum Erliegen kommen auf dem Boden, vergeht, mit 3,2s länger ist als die in Requirement R08 definierte Zeit von 3s, befindet sich das

System noch innerhalb der Spezifikation. Die restlichen Aufnahmen wurden korrekt als Sturz klassifiziert.

5.3 Ermitteln einer Optimalkonfiguration für die Schwellwerte

In den folgenden Kapiteln werden größere Datensätze ausgewertet, um die Grenzen des entwickelten Systems abzustecken. Für diese Auswertung muss das System zunächst konfiguriert werden. Dieses Kapitel beschreibt wie die verwendete Konfiguration ermittelt wird. Die vollständigen Tabellen und alle verwendeten Videoaufnahmen, die für die Auswertung genutzt wurden, sind im digitalen Anhang A zu finden.

Die Sturzerkennung entscheidet über drei Schwellwerte, ob eine Situation als Sturz bewertet wird oder nicht. Die Konfiguration dieser Schwellwerte hat damit direkten Einfluss auf die Qualität der Sturzerkennung. Bei der Konfiguration geht es darum, einen Kompromiss zwischen einer zuverlässigen Sturzerkennung und einem System zu finden, das möglichst wenige Alltagssituationen missinterpretiert.

Folgende Schwellwerte gilt es einzustellen:

- `frame_threshold`: Die Anzahl an Frames, die ein Sturz maximal einnehmen darf.
- `fall_threshold_percentage`: Die minimale prozentuale Abnahme des Höhenunterschieds zwischen dem Kopf und den Füßen, die zwischen zwei aufeinanderfolgenden Frames bestehen muss, sodass der Startpunkt eines Sturzes markiert wird.
- `fall_distance_threshold_standing`: Die minimale Fallhöhe, die während eines Sturzes zurückgelegt werden muss.

Nachfolgend wird der Einfluss der Schwellwerte `fall_threshold_percentage` und `fall_distance_threshold_standing` untersucht. Der Schwellwert `frame_threshold` bleibt über alle Tests hinweg konstant bei 60 Frames, da das mit der maximalen Sturzdauer in Requirement R03, bei einer Bildfrequenz von 20 fps, übereinstimmt.

Eine Optimalkonfiguration wird in drei Tests ermittelt.

Tabelle 5.3: Verwendete Konfigurationen für Test 1.

	Konfig. 1.1	Konfig. 1.2	Konfig. 1.3
fall_distance_ threshold_standing	-0,4	-0,65	-0,15
fall_threshold_per- centage	-0,15	-0,1	-0,3
Korrekte Stürze	23	21	21
Nicht erkannte Stürze	19	21	21
Korrekte ADL	25	20	27
ADL als Sturz	10	15	8
Insgesamt korrekt er- kannt	48	41	48

- **Test 1:** Es werden zunächst drei Konfigurationen getestet, die sich untereinander in beiden Schwellwerten unterscheiden. Aus diesen Konfigurationen wird die Konfiguration bestimmt, die sowohl die meisten ADL als auch die meisten Stürze korrekt klassifiziert.
- **Test 2:** Es werden drei neue Konfigurationen getestet, die sich nur in dem Schwellwert `fall_threshold_percentage` unterscheiden, der Wert für `fall_distance_threshold_standing` wird mit dem Wert der besten Konfiguration aus Test 1 versehen und konstant gehalten. Aus den Ergebnisse dieser Konfigurationen wird erneut die ausgewählt, die die meisten Situationen korrekt interpretiert.
- **Test 3:** Es wird nun der Wert für `fall_threshold_percentage` aus Test 2 übernommen und der Wert für `fall_distance_threshold_standing` variiert. Die Konfiguration mit den meisten korrekt bewerteten Situationen ist die Optimalkonfiguration.

5.3.1 Test 1

Für diesen Test wurden drei Konfigurationen ausgewählt, die unterschiedliche Charakteristiken abbilden sollen:

- **Konfiguration 1.2:** Eine Konfiguration, die schon kleine Verringerungen in der Höhendifferenz (10%) zwischen dem Kopf und den Füßen innerhalb von zwei Fra-

mes als möglichen Sturz klassifiziert. Die Sturztiefe ist hingegen mit 0,65 m im Vergleich zu den anderen Konfigurationen am höchsten angesetzt.

- **Konfiguration 1.3:** Eine Konfiguration, die den Start eines Sturzes erst markiert, wenn der Höhenunterschied zwischen dem Kopf und den Füßen innerhalb von zwei Frames um 30 % abnimmt. Die Sturztiefe ist mit 0,15 m im Vergleich zu den anderen Konfigurationen am geringsten.
- **Konfiguration 1.1:** Ein Konfiguration, die auf eine prozentuale Verringerung von 15 % und auf eine Sturztiefe von 0,4m prüft und damit einen Kompromiss aus Konfiguration 1.1 und Konfiguration 1.2 bildet.

Die Ergebnisse für diese Konfigurationen sind in Tabelle 5.3 festgehalten. Alle Tests werden nach den korrekt erkannten Stürzen, den nicht erkannten Stürzen, den korrekt erkannten ADL, den ADL, die fälschlicherweise als Sturz eingeordnet wurden, und den insgesamt korrekt erkannten Situationen ausgewertet.

Ein Blick in Tabelle 5.3 zeigt, dass Konfiguration 1.1 und Konfiguration 1.3 48 der insgesamt 77 Situationen korrekt bewertet haben. Dabei hat Konfiguration 1.3 weniger ADL als Sturz klassifiziert, allerdings auch mehr tatsächliche Stürze nicht erkannt.

Konfiguration 1.1 missinterpretiert zwei ADL mehr als Konfiguration 1.3 als Sturz, erkennt dafür zwei Stürze mehr als Konfiguration 1.3.

Konfiguration 1.2 schneidet mit 41 korrekt erkannten Situationen am schlechtesten ab. Auffällig ist, dass Konfiguration 1.2 dabei vor allem erheblich mehr ADL als Stürze interpretiert als die anderen Konfigurationen.

5.3.2 Test 2 - Konstante Fallhöhe

Für Test 2 wird die Fallhöhe der Konfiguration 1.1 übernommen und konstant gehalten. Der Schwellwert `fall_threshold_percentage` wird variiert.

Die Parameter und die Ergebnisse der Konfigurationen können Tabelle 5.4 entnommen werden.

Die Konfigurationen zeigen, dass ein Erhöhen des Schwellwerts `fall_threshold_percentage` die Anzahl von missinterpretierten ADL deutlich verringern kann. Ferner zeigt Konfiguration 2.3, dass ein zu großer Wert für diesen Schwellwert auch die Anzahl an korrekt erkannten Stürzen verringert.

Konfiguration 2.2 wird als beste Konfiguration dieses Tests ausgewählt. Konfiguration

Tabelle 5.4: Verwendete Konfigurationen für Test 2. Konstante Fallhöhe.

	Konfig. 2.1	Konfig. 2.2	Konfig. 2.3
fall_distance_- threshold_standing	-0,4	-0,4	-0,4
fall_threshold_per- centage	-0,1	-0,2	-0,3
Korrekte Stürze	24	22	15
Nicht erkannte Stürze	18	20	27
Korrekte ADL	20	30	33
ADL als Sturz	14	5	2
Insgesamt korrekt er- kannt	44	52	48

Tabelle 5.5: Verwendete Konfigurationen für Test 3. Konstanter prozentualer Höhenunterschied.

	Konfig. 3.1	Konfig. 3.2	Konfig. 3.3
fall_distance_- threshold_standing	-0,3	-0,5	-0,6
fall_threshold_per- centage	-0,2	-0,2	-0,2
Korrekte Stürze	26	22	20
Nicht erkannte Stürze	16	20	22
Korrekte ADL	30	32	32
ADL als Sturz	5	3	3
Insgesamt korrekt er- kannt	56	54	52

2.2 erkennt zwei Stürze weniger als Konfiguration 2.1, verringert allerdings erheblich die Anzahl der ADL, die als Sturz angesehen werden.

5.3.3 Test 3 - Konstanter prozentualer Höhenunterschied

Für Test 3 wird der Wert für `fall_threshold_percentage` von Konfiguration 2.2 übernommen und konstant gehalten. Der Schwellwert `fall_distance_threshold_standing` wird variiert.

Die Parameter und die Ergebnisse der Konfigurationen können Tabelle 5.5 entnommen werden.

Tabelle 5.6: Vergleich einer weiteren Konfiguration mit der Konfiguration 3.1.

	Konfig. 3.0	Konfig. 3.1
fall_distance_- threshold_standing	-0,25	-0,3
fall_threshold_per- centage	-0,2	-0,2
Korrekte Stürze	26	26
Nicht erkannte Stürze	15	15
Korrekte ADL	28	30
ADL als Sturz	7	5
Insgesamt korrekt er- kannt	54	56

Konfiguration 3.2 und Konfiguration 3.3 zeigen, dass eine Erhöhung der minimalen Fallhöhe zur Folge hat, dass weniger Stürze korrekt erkannt werden, aber auch, dass weniger ADL als Sturz eingeordnet werden. Niedrigere Werte für die minimale Fallhöhe haben den gegenläufigen Effekt. Konfiguration 3.1 erkennt mehr Stürze, missinterpretiert aber auch mehr ADL als Sturz.

Konfiguration 3.1 hat von allen getesteten Konfigurationen die meisten Situationen korrekt bewertet. Um zu überprüfen, ob eine Konfiguration mit einem noch kleineren Wert für `fall_distance_threshold_standing` ein noch besseres Ergebnis erzeugt, wurde nachträglich eine weitere Konfiguration getestet. Der Vergleich dieser Konfiguration (Konfiguration 3.0) mit der Konfiguration 3.1 ist in Tabelle 5.6 festgehalten. Es zeigt sich, dass ein noch kleinerer Wert für die minimale Fallhöhe zur Folge hat, dass mehr ADL als Sturz eingeordnet werden. Daher wird Konfiguration 3.1 für die restliche Arbeit als Optimalkonfiguration angesehen.

5.4 Auswertung von Aufnahmen in einer Smart Home Umgebung

5.4.1 Beschreibung der Aufnahmen

Diese Aufnahmen wurden mit vier Domkameras und zusätzlich mit derselben Webcam, die in Kapitel 5.2 genutzt wurde, erzeugt. Die Domkameras überblicken das Schlafzimmer, den Flur, das Wohnzimmer und die Küche des LP. Teilweise überschneiden sich die

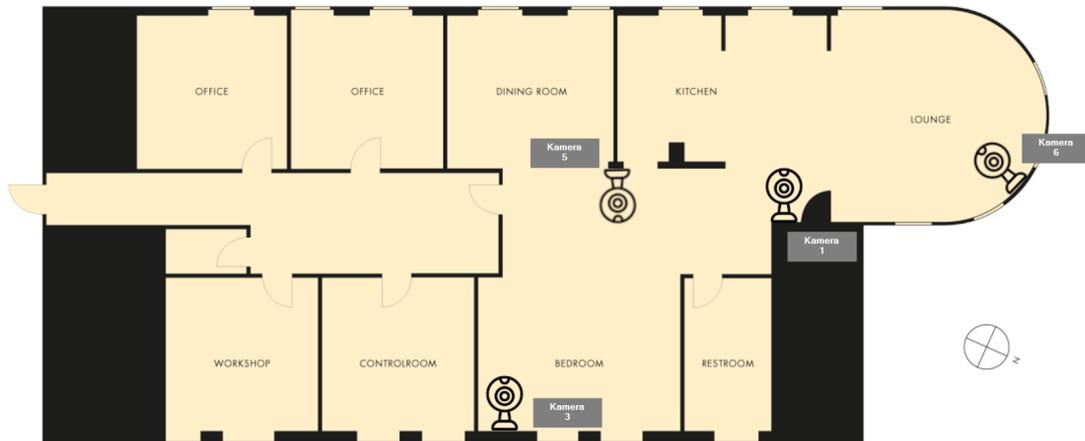


Abbildung 5.4: Übersicht über die genutzten Domkamas des LP. Eigene Darstellung nach [1].

Bereiche, die von den Kamas eingesehen werden. Die Positionen der genutzten Kamas sind in Abbildung 5.4 skizziert. Die Domkamas liefern Aufnahmen mit einer Auflösung von $1280 * 480$ Pixeln bei einer Bildfrequenz von 20 fps. Die Aufnahmen der Webcam haben eine Auflösung von $640 * 480$ Pixeln bei einer Bildfrequenz von 25 fps.

Die Aufnahmen aller Kamas kombiniert ergeben 35 ADL-Aufnahmen und 42 Sturz-Aufnahmen. Wie bereits in Kapitel 5.2 wird bei den Aufnahmen darauf geachtet, dass sowohl die Alltagssituationen als auch die Stürze möglichst divers sind. Einige Situationen wurden dabei von zwei Kamas gleichzeitig aufgenommen.

5.4.2 Auswertung

Die Auswertung dieses Datensatzes ist aufgrund der unterschiedlichen Dimensionen und Umgebungen nachfolgend in die Webcam-Aufnahmen und die Aufnahmen der Domkamas unterteilt.

Webcam-Aufnahmen

Insgesamt wurden 15 ADL-Aufnahmen erstellt, von denen elf Aufnahmen eine Person bei einer Sportübung zeigen. Die Sportübungen werden in einer, dem Liegen ähnelnden, Position ausgeführt, weshalb eine Betrachtung dieser Situationen von besonderem Interesse

ist. Zu den ausgeführten Sportübungen zählen Liegestütze, Crunches und Bergsteiger. Die restlichen ADL-Aufnahmen zeigen eine Person, die sich zu einem Stuhl bewegt und hinsetzt, eine Person, die etwas von dem Boden aufhebt, und zwei Mal eine Person, die sich flach auf den Boden legt.

Das Tracking der Keypoints war in allen Situationen zuverlässig. In der Optimalkonfiguration wurden keine ADL-Aufnahmen, die keine Sportübung zeigen, als Sturz missinterpretiert. Von den elf Aufnahmen mit Sportübungen wurden zwei als Sturz klassifiziert: Eine Aufnahme von Liegestützen und eine Aufnahme des Bergsteigers.

Domkamera-Aufnahmen

Die Extraktion von Keypoints aus den Aufnahmen der Domkameras stellt sich als größere Herausforderung heraus. Die Aufnahmen weisen ein Seitenverhältnis von 8 : 3 auf. Tests mit Aufnahmen in diesem Seitenverhältnis zeigen, dass es nicht zuverlässig gelingt Keypoints zu berechnen.

Um diesem Problem entgegen zu wirken, werden die Aufnahmen, wenn sie von einer Domkamera stammen, auf eine Auflösung von 640 * 480 Pixel und damit auf ein Seitenverhältnis von 4 : 3 zugeschnitten. Das Zuschneiden erfolgt im Programmcode, somit müssen die Aufnahmen zuvor nicht bearbeitet werden. Welcher Bereich der Aufnahme ausgeschnitten werden soll, ist abhängig von der verwendeten Domkamera und kann in einer Konfigurationsdatei eingestellt werden. Durch ihr breites Bild weisen alle Domkameras Bereiche auf, die für die Sturzerkennung uninteressant sind und weitgehend ohne Informationsverlust abgeschnitten werden können. Abbildung 5.5 zeigt den Vergleich zwischen einem nicht zugeschnittenen Bild und einem zugeschnittenen Bild. Die Aufnahmen wurden mit den selben Konfigurationen untersucht. Eine Pose Estimation kann nur bei dem zugeschnittenen Bild durchgeführt werden.

Nach dem Zuschneiden des Bildes arbeitet die Pose Estimation zuverlässiger, dennoch gibt es weiterhin einige Aufnahmen, für die keine Keypoints berechnet werden können. Unter den 47 Aufnahmen, die mit den Domkameras erstellt wurden, sind insgesamt 21 Aufnahmen (sechs ADL-Aufnahmen und 15 Sturz-Aufnahmen) für die teilweise oder vollständig keine Pose Estimation durchgeführt werden kann. In dem Großteil der Aufnahmen befindet sich die Person zu weit von der Kamera entfernt, sodass es MediaPipe nicht gelingt Keypoints auszugeben. In zwei der Sturz-Aufnahmen fällt die Person so aus dem Bild, dass nach dem Sturz keine Haltung ermittelt werden kann. Anhand dieser Aufnahmen lässt sich keine Aussage über die Qualität der hier entwickelten Sturzerken-



Abbildung 5.5: Oben: Nicht zugeschnittenes Bild, keine Keypoints können extrahiert werden. Unten: Zugeschnittenes Bild, Keypoints können extrahiert werden.

nung tätigen, da die Sturzerkennung ohne Keypoints keine Annahmen über die Situation trifft.

Nimmt man diese Aufnahmen aus der Wertung ergibt sich für Konfiguration 3.3 das in Tabelle 5.7 dargestellte Ergebnis. In Tabelle 5.8 ist die zugehörige Konfusionsmatrix abgebildet. Insgesamt erreicht die Lösung eine *True-Positive-Rate* von 100% und eine *False-Positive-Rate* von 17,2%.

5.5 Auswertung von Videos Dritter

Abschließend wird die Flexibilität der entwickelten Sturzerkennung getestet. Hierfür werden Videoaufnahmen stürzender Personen von Video-Plattformen ausgewertet.

5.5.1 Beschreibung der Aufnahmen

Die Qualität, die Kameraführung, die Kameraposition und die Umgebung der Videos variieren dabei stark. Insgesamt wurden 15 Videos getestet. Enthalten sind Videos

Tabelle 5.7: Auswertung der Konfiguration 3.3 nach Abzug der Aufnahmen, für die keine Pose Estimation erfolgen konnte.

	Konfiguration 3.3
fall_distance_threshold_standing	-0,3
fall_threshold_percentage	-0,2
Korrekte Stürze	26
Nicht erkannte Stürze	0
Korrekte ADL	24
ADL als Sturz	5
Insgesamt korrekt erkannt	50

Tabelle 5.8: Konfusionsmatrix der Auswertung für Konfiguration 3.3.

		Klassifiziert als	
		Sturz	kein Sturz
Tatsächlich Sturz	26	0	$TPR = \frac{26}{26} = 100\%$
Tatsächlich kein Sturz	5	24	$FPR = \frac{5}{29} = 17,2\%$

- mit frei geführter Kamera,
- mit schwarzweiß Aufnahmen einer Nachtkamera,
- von Überwachungskameras im Industrieumfeld,
- von Kameras im privaten Innenbereich,
- von Überwachungskameras im Außenbereich und
- von stürzenden Personen bei Sportarten.

5.5.2 Auswertung

Die Auswertung erfolgt mit der in Kapitel 5.3 ermittelten Optimalkonfiguration. Für etwa die Hälfte der Aufnahmen (sieben von 15 Aufnahmen) ist keine Pose Estimation möglich. Von den verbleibenden acht Aufnahmen kann für fünf Aufnahmen zumindest zeitweise eine Pose Estimation durchgeführt werden, oft verliert MediaPipe kurz vor dem Sturz oder während des Sturzes die Person, da sie sich zu weit von der Kamera entfernt hat oder

durch zu viele Objekte verdeckt wird. Für diese Aufnahmen wurde kein Sturz erkannt. Die übrigen drei Aufnahmen werden als Sturz klassifiziert, wobei eine der Aufnahmen nur aufgrund eines Artefakts als Sturz eingeordnet wird.

6 Diskussion

Zunächst werden die in Kapitel 5 beschriebenen Ergebnisse evaluiert und die Grenzen des Systems aufgezeigt. Abschließend werden Empfehlungen für weiterführende Arbeiten gegeben.

6.1 Evaluierung der Ergebnisse

Die Ergebnisse der Untersuchungen zeigen, dass das Ziel, eine Sturzerkennung mittels Pose Estimation zu entwickeln, erreicht wurde. Deutlich werden dabei zwei Voraussetzungen, die für eine Einführung eines solchen Systems in ein reales Zuhause nötig sind.

Voraussetzung 1 - Eine erfolgreiche Pose Estimation

Die Grundvoraussetzung für das System ist, dass es MediaPipe möglich ist eine Pose Estimation für die übergebenen Dateien durchzuführen. Die gesamte Sturzerkennung beruht auf berechneten Werten, die sich aus den Keypoints der Pose Estimation ergeben. Fehlen diese Keypoints oder werden diese nicht akkurat ermittelt, ist es dem System nicht möglich eine Aussage über die Situation zu treffen. Die Wahrscheinlichkeit für eine erfolgreiche Pose Estimation kann dabei erhöht werden, wenn der Blick der Kamera auf die Person möglichst nicht durch Gegenstände verdeckt wird. Ein weiterer Faktor ist der Abstand der Person zu der Kamera. Hier kann beobachtet werden, dass es MediaPipe gelingt eine Person auch in größerer Entfernung zu tracken, wenn für diese Person im Vorhinein erfolgreich eine Pose Estimation durchgeführt werden konnte. Tritt eine Person in gleicher Entfernung in das Bild, werden keine Keypoints ermittelt, bis sich die Person der Kamera weit genug genähert hat. Dieses Verhalten wird in Abbildung 6.1 deutlich. Hier sind 4 Frames einer Videoaufnahme abgebildet. Die Aufnahme beginnt mit dem Bild links oben, in dem sich die Person nah genug an der Kamera befindet, sodass MediaPipe eine Pose Estimation durchführen kann. Die Person bewegt sich quer durch den



Abbildung 6.1: Zeitliche Abfolge der Bilder: links oben, rechts oben, links unten, rechts unten. Einfluss der Entfernung zwischen Kamera und Person für eine erfolgreiche Pose Estimation.

Raum und entfernt sich damit von der Kamera. MediaPipe liefert weiterhin Keypoints der Person, bis sie vollständig hinter einer Wand verschwindet. Die Person tritt nach kurzer Zeit wieder hinter der Wand hervor. Obwohl es MediaPipe zuvor in derselben Entfernung möglich war eine Pose Estimation durchzuführen, muss sich die Person der Kamera weiter nähern bevor eine erneute Pose Estimation möglich ist.

Dieses Verhalten kann durch ein Verringern des Parameters `min_detection_confidence` minimal verbessert werden. Allerdings hat, wie in Kapitel 5.1 beschrieben, ein Verringern des Parameters das vermehrte Auftreten von Artefakten zur Folge. Artefakte verändern ihre Position von Frame zu Frame unvorhersehbar und können damit als Sturz klassifiziert werden. Aus diesem Grund wird das Vermeiden von Artefakten der geringen Verbesserung in der Personenerkennung vorgezogen.



Abbildung 6.2: Aufnahme derselben Situation mit den Domkameras 1 und 6, vgl. Abbildung 5.4. Links: Kamera 6, keine Sturzmeldung aufgrund fehlender Pose Estimation. Rechts: Kamera 1, Pose Estimation erfolgreich, Sturz wird erkannt.

Eine andere Möglichkeit diesem Problem entgegen zu wirken ist die Installation von weiteren Kameras. Dabei gilt es dieselben Bereiche aus mehreren Blickwinkeln einzufangen. In Abbildung 6.2 ist ein Beispiel hierfür aus dem LP dargestellt. Während es im linken Bild MediaPipe nicht möglich war eine Pose Estimation für die Person durchzuführen, gelingt es aus einem anderen Blickwinkel. Der Sturz wäre, wenn nur das Bild von Kamera 6 zur Verfügung gestanden hätte, unbemerkt vergangen. Durch den zweiten Blickwinkel wurde der Sturz erkannt.

Voraussetzung 2 - Umgang mit False-Positives

Es wird deutlich, dass das hier entwickelte System nicht so konfiguriert werden kann, dass es ausreichend sensibel auf Stürze reagiert und gleichzeitig keine Situationen des Alltags als Sturz klassifiziert. Einige Bewegungen, vor allem Sportübungen, liefern Daten, die mit den hier verwendeten Methoden kaum von einem Sturz zu unterscheiden sind.

Um gegen dieses Problem anzugehen, wurde in einem ersten Schritt definiert, dass eine Bewegung die als Sturz klassifiziert wurde, erst final als Sturz gewertet wird, wenn die Person nicht innerhalb von 10s erneut eine stehende Haltung eingenommen hat und diese vom System erkannt wurde. Weitere Umgangsweisen mit diesem Problem werden im folgenden Kapitel diskutiert.

6.2 Ausblick

Die Empfehlungen für weiterführende Arbeiten lassen sich in drei Bereiche unterteilen: Die Implementation eines Feedbacksystems, das Absetzen eines Notrufs nach Erkennung eines Sturzes und das Gewähren eines Zugangs zu der Wohnung der gestürzten Person.

Feedbacksystem

Für den Umgang mit False-Positives wäre es denkbar, dass das System, wenn es einen Sturz erkannt hat, versucht mit der Person in einen Dialog zu treten. So hätte die Person die Möglichkeit selbst zu bewerten, ob sie sich in einer Situation befindet in der sie Hilfe benötigt. Denkbar wäre, dass das System, wenn es einen Sturz erkannt hat, über akustische Signale (Warnton oder menschliche Sprache) dem Bewohnenden deutlich macht, dass es einen Sturz erkannt hat und ohne Intervention einen Notruf absetzen würde.

Ist das Zuhause mit einer Spracherkennung ausgestattet, kann die Person sprechend antworten. In einem Projekt von 2022 wurde im LP ein Dialogsystem entwickelt, das für diesen Zweck genutzt werden könnte [26].

Gibt es keine funktionierende Spracherkennung, wäre es denkbar das Feedbacksystem über die, für die Sturzerkennung erforderlichen, Kameras zu steuern. So könnte es festgelegte Gesten oder Abfolgen von Gesten geben, die eine latente Sturzmeldung verwerfen und das Absetzen eines Notrufes verhindern.

Eine weitere Möglichkeit wäre, die Sturzerkennung präventiv auszusetzen. Weiß die Person, dass sie sich in eine Position begeben wird, die eventuell als Sturz missinterpretiert werden könnte, beispielsweise während eines Heimtrainings, so könnte sie das dem System mitteilen und die Sturzerkennung für eine festgelegte Zeit aussetzen. Hierfür müsste die Person das System nicht zwingend über einen Rechner steuern. Vorstellbar wären, wie oben aufgeführt, ein Dialog über Sprache oder Gesten. Das System sollte sich dabei nach einer festgelegten Zeit selbst reaktivieren, sodass ein Vergessen ausgeschlossen wird.

Absetzen eines Notrufs

Wurde die Sturzmeldung abgesetzt und nicht von der gestürzten Person abgebrochen, wird bislang die Meldung eines Sturzes durch das System nicht weiter verarbeitet. Wünschenswert wäre, dass auf eine Sturzmeldung mit einer definierten Notrufrkette reagiert wird.

In einem ersten Schritt könnte eine oder mehrere hinterlegte Personen kontaktiert werden, dass ein Sturz erkannt wurde. Reagieren diese Personen nicht oder melden, dass sie nicht in der Lage sind sich der Situation anzunehmen, kann der Notruf an eine zentrale Notrufstelle umgeleitet werden. Dabei könnten Standortdaten des Zuhauses, Daten über die Person (Alter, Gewicht, Geschlecht, Größe), Patienten-/Patientinnendaten und Aufnahmen des Sturzes übermittelt werden. Die Übermittlung solcher Daten müsste datenschutzrechtlich geprüft werden.

Zugang zur gestürzten Person

Die Sturzmeldung wurde an eine*n Ersthelfer*in übertragen. Die gestürzte Person befindet sich möglicherweise noch immer auf dem Boden und kann sich selbst nicht aufrichten. Um eine Versorgung der gestürzten Person zu gewährleisten muss den Helfenden Zugang zu dem Zuhause der gestürzten Person gewährt werden. Hierfür wäre der Einsatz von elektronischen Türschlössern denkbar. Diese Türschlösser können, je nach Bauart, über einen Zahlencode, einen NFC-Chip oder einen QR-Code geöffnet werden.

Im Falle eines Zahlencodes, könnte entweder der aktuell eingestellte Zahlencode an die kontaktierte Person übertragen werden, dieser müsste aus Sicherheitsgründen nach dem Lösen der Notfallsituation geändert werden, oder ein temporärer Zahlencode übermittelt werden, der ein einmaliges Entsperren der Tür ermöglicht.

Kann das Türschloss über einen NFC-Chip entsperrt werden, müsste ein solcher NFC-Chip vor dem Sturz bei der kontaktierten Person hinterlegt worden sein. Diese Lösung ist denkbar, wenn die helfende Person bereits vor dem Sturz feststeht.

Wird die Tür über das Scannen eines QR-Codes entriegelt, könnte ein passender QR-Code bei der Notfallmeldung übermittelt werden. Dieser kann für ein einmaliges Entsperren der Tür genutzt werden.

6.3 Fazit

In dieser Arbeit konnte eine Sturzerkennung mittels Pose Estimation entworfen werden. Hierfür wurden drei Lösungskonzepte erarbeitet und getestet. Das Lösungskonzept mit den vielversprechendsten Erfolgchancen wurde in mehreren Schritten konfiguriert und anschließend mit einem Datensatz aus Webcam-Aufnahmen und Aufnahmen aus einer Smart-Home Umgebung getestet und evaluiert.

Die Sturzerkennung erkennt mithilfe der, von MediaPipe ausgegebenen, Keypoints, die aktuelle Haltung einer Person. Wird die Haltung der Person als liegend klassifiziert, wird über zwischengespeicherte Werte aus der Vergangenheit entschieden, ob die Person diese Haltung aufgrund eines Sturzes eingenommen hat. Gelingt es MediaPipe eine Pose Estimation durchzuführen, so konnte in Kapitel 5.4 gezeigt werden, dass die Sturzerkennung für den verwendeten Datensatz eine *True-Positive-Rate* von 100 % erreicht.

Einige Alltagssituationen, vor allem Sportübungen, erzeugen dabei Daten, die denen eines Sturzes ähnlich sind. Diese Bewegungsabläufe werden von der Sturzerkennung ebenso als Sturz klassifiziert. In Kapitel 6.2 werden Lösungsvorschläge für den Umgang mit diesen False-Positives aufgezeigt.

Die Grundvoraussetzung für die Sturzerkennung ist, dass es MediaPipe möglich ist, eine Pose Estimation durchzuführen. Ist dies nicht der Fall, liegen der Sturzerkennung keine Daten vor, die ausgewertet werden können. Für diese Fälle wird keine Annahme über einen möglichen Sturz getroffen. Wie in Kapitel 6.1 dargelegt, kann die Wahrscheinlichkeit für eine erfolgreiche Pose Estimation durch eine höhere Kamera-Abdeckung des Zuhauses vergrößert werden.

Literaturverzeichnis

- [1] HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN: *Welcome to the LIVING PLACE lab!*. – URL <https://livingplace.haw-hamburg.de/>. – Zugriffsdatum: 2022-09-14
- [2] ANDREADIS, Stelios ; STAVROPOULOS, Thanos ; MEDITSKOS, Georgios ; KOMPATSIARIS, Ioannis: *Dem@Home: Ambient Intelligence for Clinical Support of People Living with Dementia*, 05 2016, S. 357–368. – ISBN 978-3-319-47601-8
- [3] AYIMDI TEKEMETIEU, Armel ; PIGOT, H el ene ; BOTTARI, Carolina ; GAGNON-ROY, Mireille ; GIROUX, Sylvain: *Modeling an Adaptive Resident-System Interaction for Cognitive Assistance in Ambient Assisted Living*. In: *Proceedings of the 9th International Conference on Human-Agent Interaction*. New York, NY, USA : Association for Computing Machinery, 2021 (HAI '21), S. 183–192. – URL <https://doi.org/10.1145/3472307.3484166>. – ISBN 9781450386203
- [4] BALAKRISHNAN, Sumathi ; VASUDAVAN, Hemalata ; MURUGESAN, Raja K.: *Smart Home Technologies: A Preliminary Review*. In: *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*. New York, NY, USA : Association for Computing Machinery, 2018 (ICIT 2018), S. 120–127. – URL <https://doi.org/10.1145/3301551.3301575>. – ISBN 9781450366298
- [5] BENDEL, Prof. Dr. O.: *Definition: Was ist Smart Home?* 2021. – URL <https://wirtschaftslexikon.gabler.de/definition/smart-home-54137/version-384532>. – Zugriffsdatum: 2022-09-14
- [6] BUNDESMINISTERIUM F UR FAMILIE, SENIOREN, FRAUEN UND JUGEND: *L anger zuhause leben - Ein Wegweiser f ur das Wohnen im Alter*. Bundesministerium f ur Familie, Senioren, Frauen und Jugend, 2019
- [7] DEUTSCHES ROTES KREUZ: *Hausnotruf*. – URL <https://www.drk.de/hilfe-in-deutschland/senioren/altersgerechtes-wohnen/hausnotruf/>. – Zugriffsdatum: 2022-10-06

- [8] EUROPÄISCHE KOMMISSION REFERAT SOZIALE UND DEMOGRAFISCHE ANALYSE: *Die demografische Zukunft Europas: von der Herausforderung zur Chance*. Amt für Amtliche Veröff. der Europ. Gemeinschaften, 2007 (Kom: Europäische Kommission). – URL <https://books.google.de/books?id=4oWmQAACAAJ>. – ISBN 9789279020919
- [9] GINÉS, Hidalgo ; ZHE, Cao ; TOMAS, Simon ; SHIH-EN, Wei ; YAADHAV, Raaj ; HANBYUL, Joo ; SHEIKH, Yaser: *OpenPose*. – URL <https://github.com/CMU-Perceptual-Computing-Lab/openpose>. – Zugriffsdatum: 2022-09-21
- [10] GOOGLE LLC: *MediaPipe Pose*. – URL <https://google.github.io/mediapipe/solutions/pose.html>. – Zugriffsdatum: 2022-09-20
- [11] GRISHCHENKO, Ivan ; BAZAREVSKY, Valentin ; SABIE, Ahmed ; MAYES, Jason: *Body Segmentation with MediaPipe and TensorFlow.js*. – URL <https://blog.tensorflow.org/2022/01/body-segmentation.html>. – Zugriffsdatum: 2022-09-23
- [12] HOFFMANN, Sarah: *Ambient Assisted Living Deutschland*. 2016. – URL <http://www.aal-deutschland.de/>. – Zugriffsdatum: 2022-09-19
- [13] HOLTE, Michael B. ; TRAN, Cuong ; TRIVEDI, Mohan M. ; MOESLUND, Thomas B.: Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments. In: *IEEE Journal of Selected Topics in Signal Processing* 6 (2012), Nr. 5, S. 538–552
- [14] JENSEN, Magnus T. ; MAROTT, Jacob L. ; HOLTERMANN, Andreas ; GYNTELBERG, Finn: Living alone is associated with all-cause and cardiovascular mortality: 32 years of follow-up in the Copenhagen Male Study. In: *European Heart Journal - Quality of Care and Clinical Outcomes* 5 (2019), 01, Nr. 3, S. 208–217. – URL <https://doi.org/10.1093/ehjqcco/qcz004>. – ISSN 2058-5225
- [15] JIUQIANG, Tang ; CHUOLING ; GOWROJI, Sunil: *MediaPipe*. – URL <https://github.com/google/mediapipe>. – Zugriffsdatum: 2022-09-21
- [16] KERAS: *ResNet and ResNetV2*. – URL <https://keras.io/api/applications/resnet/>. – Zugriffsdatum: 2022-09-22
- [17] KREISS, Sven ; BERTONI, Lorenzo ; ALAHI, Alexandre: *PifPaf: Composite Fields for Human Pose Estimation*. 2019. – URL <https://arxiv.org/abs/1903.06593>

- [18] KUMAR, Abhishek ; DENI RAJ, Ebin: *Silhouettes for Human Posture Recognition*. 2020. – URL <https://dx.doi.org/10.21227/9c9b-3j44>
- [19] LEE, Tracy ; MIHAILIDIS, Alex: An intelligent emergency response system: preliminary development and testing of automated fall detection. In: *Journal of Telemedicine and Telecare* 11 (2005), S. 194 – 198
- [20] LI, Miaopeng ; ZHOU, Zimeng ; LI, Jie ; LIU, Xinguo: Bottom-up Pose Estimation of Multiple Person with Bounding Box Constraint, 08 2018, S. 115–120
- [21] MARTINEZ, Julieta ; HOSSAIN, Rayat ; ROMERO, Javier ; LITTLE, James J.: A simple yet effective baseline for 3d human pose estimation. In: *Proceedings IEEE International Conference on Computer Vision (ICCV)*. Piscataway, NJ, USA : IEEE, Oktober 2017
- [22] MAUREENTKT: *Selecting Your Real-Time Pose Estimation Models*. – URL <https://maureentkt.medium.com/selecting-your-2d-real-time-pose-estimation-models-7d0777bf935f>. – Zugriffsdatum: 2022-09-21
- [23] MUBASHIR, Muhammad ; SHAO, Ling ; SEED, Luke: A survey on fall detection: Principles and approaches. In: *Neurocomputing* 100 (2013), S. 144–152. – URL <https://www.sciencedirect.com/science/article/pii/S0925231212003153>. – Special issue: Behaviours in video. – ISSN 0925-2312
- [24] MUNEA, Tewodros L. ; JEMBRE, Yalew Z. ; WELDEGEBRIEL, Halefom T. ; CHEN, Longbiao ; HUANG, Chenxi ; YANG, Chenhui: The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation. In: *IEEE Access* 8 (2020), S. 133330–133348
- [25] NAIT-CHARIF, H. ; MCKENNA, S.J.: Activity summarisation and fall detection in a supportive home environment. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*. Bd. 4, 2004, S. 323–326 Vol.4
- [26] NEITZEL, Kathleen: *Kontextbasiertes Dialogsystem für Companions*. 2022
- [27] OMEGA ELECTRONIC GMBH: *Wie hell darf es denn bitte sein? Ein Beleuchtungsratgeber für jeden Raum*. 2021. – URL <https://www.omega.ms/blogs/ratgeber/wie-hell-darf-es-denn-bitte-sein-ein-beleuchtungsratgeber-fur-jeden-raum>. – Zugriffsdatum: 2022-09-21

- [28] PIRES, Samara ; RODRIGUES, Sonia ; AROKIADASS, Lourd B. ; CHOPRA, Sejal: A Real-Time Position Monitoring System for Fall Detection and Analysis Using Human Pose Estimation. In: *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, 2021, S. 1–7
- [29] SINGH, Komal ; RAJPUT, Akshay ; SHARMA, Sachin: Human Fall Detection Using Machine Learning Methods: A Survey. In: *International Journal of Mathematical, Engineering and Management Sciences* 5 (2019), 11, S. 161–180
- [30] TSANG, Sik-Ho: *Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection)*. – URL <https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8>. – Zugriffsdatum: 2022-09-22
- [31] WILLIAMS, G. ; DOUGHTY, K. ; CAMERON, K. ; BRADLEY, D.A.: A smart fall and activity monitor for telecare applications. In: *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286)* Bd. 3, 1998, S. 1151–1154 vol.3
- [32] WORLD HEALTH ORGANIZATION: *WHO global report on falls prevention in older age*. 2008
- [33] YU, Xinguo: Approaches and principles of fall detection for elderly and patient, 08 2008, S. 42 – 47. – ISBN 978-1-4244-2280-7
- [34] ZION MARKET RESEARCH: *Smart Home Market - Global Industry Analysis*. 2021. – URL <https://www.zionmarketresearch.com/report/smart-home-market>. – Zugriffsdatum: 2022-09-14

A Digitaler Anhang

Auf dem Datenträger ist folgende Ordnerstruktur vorzufinden:

- **Dokumentation:** Enthält dieses Dokument als \LaTeX und PDF.
- **Datenanalyse:** Enthält alle verwendeten Excel-Tabellen, die für die Analyse erhobener Daten genutzt wurden.
- **Diagramme:** Enthält alle selbst erstellten Diagramme dieser Arbeit.
- **Code:** Enthält den vollständigen Sourcecode der entwickelten Software inklusive der Bilder und Videos, die zum testen genutzt wurden.

Glossar

False-Positive-Rate Wahrscheinlichkeit, dass eine Situation fälschlicherweise als Sturz klassifiziert wird.

True-Positive-Rate Wahrscheinlichkeit, dass eine Situation richtig klassifiziert wird.

Artefakt Keypoints, die Stellen des Bildes zugeschrieben werden an denen sich keine Person befinde.

Bounding Box Das kleinste mögliche Rechteck, welches alle Punkte eines Interessengebiets umschließt.

HAW Hamburg Die HAW Hamburg ist die vormalige Fachhochschule am Berliner Tor.

Keypoint Datenpunkt der durch zwei bis drei Koordinaten spezifiziert wir.

LIVING PLACE lab Labor der HAW Hamburg zur Untersuchung von Forschungsfragen im Kontext von Smart Home Umgebungen.

Pose Estimation Verfahren in der Computer Vision, um die Position und Orientierung einer Person in einem Bild festzustellen.

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original