

Pose estimation for Time up and go (TUG)

Chafik Chahraestan, Moa Eklund, Oscar Regnat, Tony Rusteberg, Anna Yazdi

Sammanfattning

Fallbenägenhet är ett allvarligt problem bland äldre och en ledande orsak till skador och sjukhusinläggningar. Traditionella metoder för att bedöma rörlighet och fallrisk, såsom TUG-testet, kräver vårdgivare och noggrann övervakning, vilket gör processen tidskrävande och subjektiv. Vi har utvecklat ett automatiserat system som använder en standardkamera och 2D-baserad pose estimation för att analysera TUG-testet. Resultaten visar effektiv spårning av nyckelpunkter och bedömning av rörlighet, vilket presenteras i form av grafer för vidare analys.

Introduktion

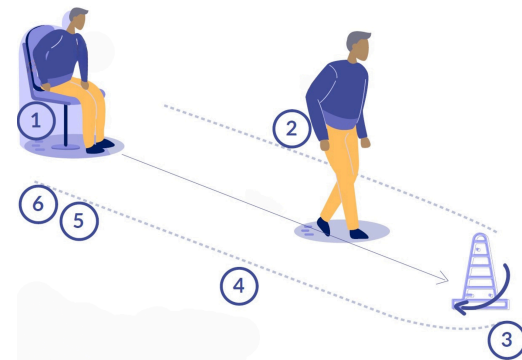
Enligt en rapport från Folkhälsomyndigheten faller tusentals äldre varje år i Sverige, vilket leder till betydande fysiska och ekonomiska konsekvenser för både individen och samhället. Fallrelaterade skador resulterar i cirka 70 000 sjukhusinläggningar årligen, och fallolyckor är en av de tio vanligaste dödsorsakerna för personer över 70 år [1]. I en studie av Shumway-Cook et al. (2000) beskrivs Time Up-and-Go-testet (TUG) som ett standardiserat verktyg för att bedöma funktionell rörlighet och fallrisk hos äldre [2].

Traditionellt utförs TUG-testet manuellt av vårdgivare, där en sjuksköterska eller terapeut övervakar en persons rörelser och mäter tiden med ett stoppur. Denna metod är dock beroende av subjektiv bedömning och manuell mätning, vilket kan leda till variationer i noggrannhet och resultat. Alternativa tekniska lösningar har introducerats för att förbättra denna process. MDPI rapporterar att RGB-D-kameror och sensorer som Kinect används för att automatisera rörelsespårning under TUG-testet [3]. Dessa lösningar har förbättrat noggrannheten i analysen av TUG-testet, men kräver ofta specialiserad hårdvara eller djupgående teknisk kunskap för implementering.

Ce Zheng presenterar ett system som använder en standardkamera och 2D-baserad pose estimation för automatisk, realtidsanalys av bilddata. Mjukvaran bygger på algoritmer som

använder maskininlärning för att analysera kroppens rörelser

Bidragen från denna uppsats inkluderar utvecklingen av ett system som använder en standardkamera och 2D-baserad pose estimation för att automatiskt analysera TUG-testet. Systemet använder en webbplattform och datorns webbkamera för att spåra kroppens nyckelpunkter och segmentera testfaser. Genom visualisering i MATLAB erbjuds mätvärden av rörlighet och fallrisk, vilket ger en kostnadseffektiv lösning för klinisk utvärdering.



Figur 1. Time Up-and-Go-test (TUG)

Design

Hårdvara

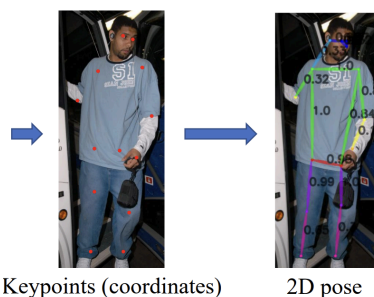
Systemet för pose estimation och skeleton tracking bygger på en kombination av hårdvara och mjukvara. Hårdvaran utgörs av en enhet med inbyggd kamera, vilket kan vara en dator, mobiltelefon eller annan kameraförsedd enhet. Denna kamera fångar realtidsbilder av individens rörelser, som sedan bearbetas för vidare analys. Hårdvaran är en central komponent då den ansvarar för att förse systemet med visuella data av tillräcklig kvalitet för att mjukvaran ska kunna utföra sin uppgift korrekt [4].

Mjukvara

TensorFlow.js är en JavaScript-plattform för maskininlärning som möjliggör körning av modeller direkt i webbläsaren som användes för detta projekt. PoseNet-modellen spårar nyckelpunkter på kroppen, såsom axlar och knän, och möjliggör realtidsanalys av rörelser [4]. Designen är byggd i ett 2D-system, vilket

innebär att rörelsedata och kroppens nyckelpunkter visualiseras i två dimensioner. Detta tillvägagångssätt är enklare att implementera jämfört med ett 3D-system och är tillräckligt för att analysera de grundläggande vinklarna och positionerna som krävs för att genomföra Time Up-and-Go-testet.

I ett 2D-system används bilddata från en enskild kameravinkel, och beräkningarna fokuserar på kroppens rörelser i ett plan, vilket reducerar komplexiteten men ändå möjliggör en effektiv analys av rörelsemönster. Genom att exakt lokalisera dessa nyckelpunkter kan systemet segmentera och identifiera olika rörelser som ingår i Time Up-and-Go-testet.



Figur 2. visar nyckelpunkter i 2D position

2.3 Integration och visualisering

Systemet är byggt för att fungera i ett webbläsarbaserat gränssnitt, vilket gör det plattformsoberoende och tillgängligt från vilken enhet som helst med en kompatibel webbläsare.

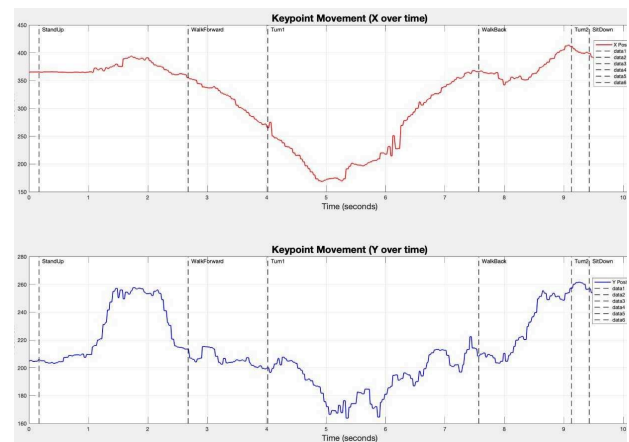
Resultat och diskussion

Efter att rörelsedata har samlats in från hårdvaran och bearbetats av mjukvaran, exporteras denna information till MATLAB för vidare analys. MATLAB används för att visualisera de insamlade data genom grafer som representerar vinklar och positioner av kroppen under hela testets gång. Graferna visar hur en person ställer sig upp, går framåt, vänder sig om, går tillbaka, vänder sig om igen och sätter sig, vilket ger en detaljerad bedömning av funktionell rörlighet och eventuella avvikelser i rörelsemönstret. Resultatet av fem körningar och genomsnittet kan betraktas i figur 3.

En betydande utmaning har varit att med hög precision simulera individer med olika rörelsenedsättningar. Funktionell videoanalys

av personer med många olika rörelsemönster kräver sofistikerad kod, vilket i sin tur är mycket tidskrävande. För att resultatet ska bli så noggrant och bra som möjligt är det viktigt att kameran placeras på samma exakta sätt vid varje körning för att säkerställa tillförlitliga resultat. Under försöken har det fastställts att den bästa vinkeln på kameran är snett åt sidan. Om kameran är placerad för långt bort minskar precisionen i att identifiera de valda nyckelpunkterna, vilket kan leda till felaktiga analyser av rörelser och vinklar. Därför valdes ett avstånd på 2 meter istället för 3 meter.

Under försöken markerades den exakta slutpunkten för sträckan på golvet för att möjliggöra jämförbara resultat mellan varje körning. En utmaning låg i att bibehålla en konstant hastighet samt att säkerställa att samtliga nyckelpunkter registrerades i varje mätning. Slutsatsen blev att en egenutvecklad PoseNet-modell skulle krävas för att uppnå den höga precision som eftersträvades i nyckelpunkt detektion, men på grund av tidsbrist kunde detta inte genomföras inom projektets ramar.



Figur 3. Grafen visar olika rörelser som i en TUG

Slutsats

Ett system för automatisk analys TUG har utvecklats med hjälp av en standardkamera. För framtida arbete rekommenderas att utvidga systemets funktionalitet genom att utveckla egna algoritmer för maskininläring som är ändamålsenliga för TUG-scenariot.

Källhänvisning:

[1] Fallolyckor bland äldre – en samhällsekonomisk analys och effektiva

preventionsåtgärder [Internet].
Folkhalsomyndigheten.se. [citerad 2024 Sep
24]. Hämtad från:
<https://www.folkhalsomyndigheten.se/publikationer-och-material/publikationsarkiv/f/fallolyckor-bland-aldre-en-samhallsekonomisk-analys-och-effektiva-preventionsatgarder/>

[2] Lamb SE, Jørstad-Stein EC, Hauer K, Becker C, Prevention of Falls Network Europe and Outcomes Consensus Group. Development of a common outcome data set for fall injury prevention trials: the Prevention of Falls Network Europe consensus. J Am Geriatr Soc [Internet]. 2005;53(9):1618–22. Hämtad från:
<https://pubmed.ncbi.nlm.nih.gov/16137297/>

[3] Choi Y, Bae Y, Cha B, Ryu J. Deep learning-based subtask segmentation of time up-and-go test using RGB-D cameras. Sensors (Basel) [Internet]. 2022 [citerad 2024 Sep 23];. Hämtad från:
<https://www.mdpi.com/1424-8220/22/17/6323>

[4] Ce Z, Wenhan W, Chen C, Taojiannan Y, Sijie Z, Ju S, et al. Deep learning-based human pose estimation: A survey [Internet]. arXiv [cs.CV]. 2020. Hämtad från:
<http://arxiv.org/abs/2012.13392>

Källhänvisning figurer:

Figur 1: Ortega-Bastidas P, Gómez B, Aqueveque P, Luarte-Martínez S, Cano-de-la-Cuerda R. Instrumented Timed Up and Go test (iTUG)—more than assessing time to predict falls: A systematic review. Sensors (Basel) [Internet]. 2023 [cited 2024 Oct 8];23(7):3426. Hämtad från:
<https://www.mdpi.com/1424-8220/23/7/3426>

Figur 2: Ce Z, Wenhan W, Chen C, Taojiannan Y, Sijie Z, Ju S, et al. Deep learning-based human pose estimation: A survey [Internet]. arXiv [cs.CV]. 2020. Available from:
<http://arxiv.org/abs/2012.13392>

Figur 3: Graf skapat i matlab av författarna.