KU LEUVEN kulak

Wetenschap & Technologie Kulak

Human Pose Estimation: Toepassing

Isaac venus
Stan Vanhecke
Mathieu Vanooteghem

KU Leuven Kulak, Wetenschap & Technologie

Titularis: Koen Van Den Abeele
Begleider: Jens Goemare
Academiejaar 2020 – 2021

Inhoudsopgave

Inleiding

Aan materiaal in de medische wereld hangt steeds een stevig prijskaartje, dus vroegen wij ons af hoe we dit wat lichter kunnen maken voor de portemonnee. Wat natuurlijk niet mag is het gebruik van hoogtechnologische medische apparatuur maar dat er in de plaats de technologie gebruikt wordt die iedereen al heeft zoals een laptop en een smartphone. We moeten dus software schrijven die met de elektrica die we al hebben kan werken. In dit verslag richten wij ons op *human pose estimation* (HPE), of lichaamspositiebepaling. We gebruiken hiervoor de open-source HPE software openpose die via neurale netwerken de positie van een persoon schat op een foto. Met dit programma kunnen we dan software schrijven voor medische doeleinden.

Hoofdstuk 1

HPE, revolutionair?

HPE is gemakkelijk, snel, goedkoop en steeds nauwkeuriger heatmaps en kansen uitleg en over openpose specifiek vermelden dat het werkt met neurale netwerken (cfr. infra)

Hoofdstuk 2

Theoretische achtergrond

2.1 Openpose

Openpose is een open-source programma dat *human pose estimations* kan doen. Openpose is grotendeels geprogrammeerd in C++ maar heeft een uitstekende Python API. Dit programma onderscheid zich van ander HPE programma's in de manier waarop het beelden met meerdere personen kan verwerken. De meeste voorgaande programma's waren maar in staat om met één persoon te werken of localiseerden eerst alle personen op de foto om nadien individueel hun positie te bepalen. Bij Openpose is het anders, het programma kan synchroon de positie van alle personen bepalen in de foto [Cao et al.(2019)Cao, Hidalgo Martinez, Simon, Wei, and Sheikh].

2.1.1 Werking

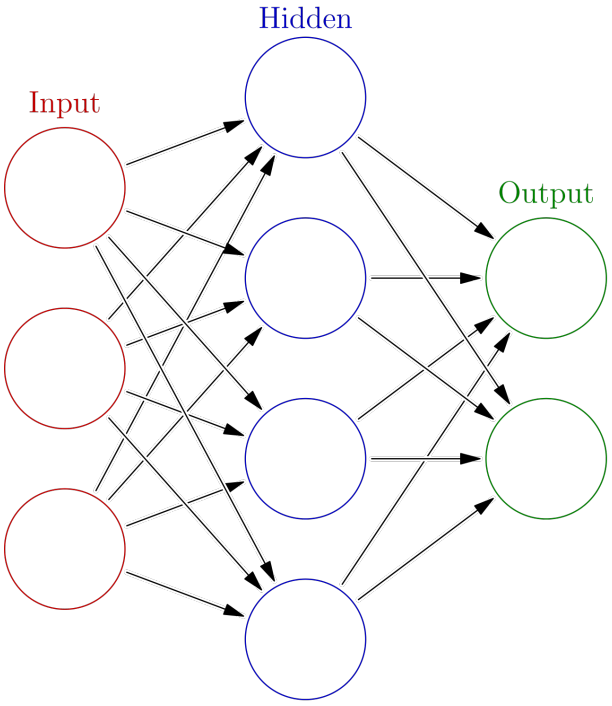
Openpose berekent eerst voor alle lichaamsdelen een heatmap. Dat zijn de locaties met het meeste kans dat het lichaamsdeel zich daar bevindt. Op basis daarvan kan hij dan de coördinaten bepalen [Bulat and Tzimiropoulos(2016)].

2.1.2 Training datasets

Dit programma is getrained met twee datasets [Cao et al.(2019)Cao, Hidalgo Martinez, Simon, Wei, and Sheikh]: de COCO dataset, die bestaat uit meer dan 100.000 beelden en meer dan 1.000.000 *keypoints*, en de MPII dataset.

2.2 Neurale netwerken

Een (artificieel) neuraal netwerk of ANN is een netwerk geïnspireerd op een biologisch neuraal netwerk zoals die voorkomt in hersenen, met als doel deze iets te doen leren. Het ANN is opgebouwd uit artificiële neuronen die een neuron in een brein moeten voorstellen. Al de neuronen zijn verbonden met andere neuronen en kunnen signalen uitsturen en ontvangen. In een ANN zijn de signalen getallen en kan elk neuron een bepaalde bewerking uitvoeren op een binnenkomend signaal om deze verder te sturen. Elke connectie heeft ook een gewicht dat bepaalt hoe sterk het signaal is dat hij uitstuurt. Dit gewicht past zich aan als het neuraal netwerk leert.



Figuur 2.1: Voorbeeld van een neuraal netwerk. (afbeelding van wikipedia.org)

2.2.1 Opbouw van een neuraal netwerk

Neuronen Zoals er hierboven staat is een artificieel neuron geïnspireerd op een neuron uit een brein. Elk neuron heeft een of meerdere inputs en één output die kan verzonden worden naar meerdere andere neuronen. De input van een neuron kan ofwel komen van de data die in het neuraal netwerk wordt gestoken of van andere neuronen. De output van de output neuronen is wat uit het programma komt. Om de output van een neuron te berekenen neem je de gewogen som van de inputs, met als gewicht de waarden van de connecties. Daarbij wordt ook nog eens een bias of vertekening opgeteld.

Connecties Om te kunnen leren moet de data natuurlijk kunnen worden doorgegeven van het ene neuron naar de andere. Dit verloopt via een "connectie" die een bepaald gewicht heeft dat de sterkte en daarmee ook het belang van een bepaalde connectie weergeeft.

Lagen Een neuraal netwerk bestaat uit meerdere lagen van neuronen, meer bepaald de *input layer*, *hidden layers* en *output layer* 2.1. Zoals de namen wel duidelijk maken geef je de data aan de *input layer* en komen de berekende waarden uit de *ouput layer*. Daartussen zijn er eventueel *hidden layers*. Tussen twee lagen zijn er meerdere manieren van connecties tussen de neuronen mogelijk.

2.2.2 Trainen van een neuraal netwerk

Bij het programmeren van een neuraal netwerk weet je natuurlijk niet wat de gewichten zijn van alle connecties. Je moet het model dus doen leren. Dit doe je door de gewichten (over meerdere leercycli) aan te passen zodat de output zo goed mogelijk past bij het gewenste resultaat. Er zal altijd een bepaalde fout op de output zitten dus het model gaat nooit perfect zijn. Het trainen van een model is gedaan als de fout op de output niet meer verkleint en dus een minimum heeft bereikt. Het aanpassen van de gewichten gebeurt met een zekere *learning rate*. Dit bepaalt de grootte van de correcties die gebeuren bij de gewichten. Bij een grote *learning rate* kan je model snel gedaan zijn met leren maar meer kans op een grotere fout op de output. Een kleine *learning rate* zal het trainen vertragen maar je uiteindelijke resultaat zal beter zijn.

Hoofdstuk 3

Toepassingen

Zoals eerder al gezegd zullen we ons focussen op toepassingen in de medische wereld dat gebruik kan maken van HPE waarvoor wij een goedkope oplossing kunnen ontwikkelen. Hierbij testen we mogelijke problemen en analyseren we of dit wel een goede optie is. We proberen dit dan allemaal op een methodische manier uit te werken, waarbij we een specifiek geval steeds breder gaan bekijken.

Een eerste toepassing is het opvolgen van de revalidatie na een schouderoperatie. Met een foto gemaakt met een gsm kunnen wij een analyse uitvoeren van de beweging van de schouder na de operatie.

Een tweede en uitgebreidere toepassing is het analyseren van de positie op de fiets. Veel amateurwielrenners hebben een slechte houding op de fiets en een professionele bikefitting kost al snel een paar honderd euro. Via lichaamspositiebepaling kunnen wij weer met een simpele foto de positie bepalen op de fiets en dan correcties voorstellen. Dit is een goedkope oplossing die voor een zeer breed publiek inzetbaar is. Voor professionele wielrenners zal dit wel waarschijnlijk niet voldoende zijn, maar voor een amateur kan dit een goedkoop alternatief zijn.

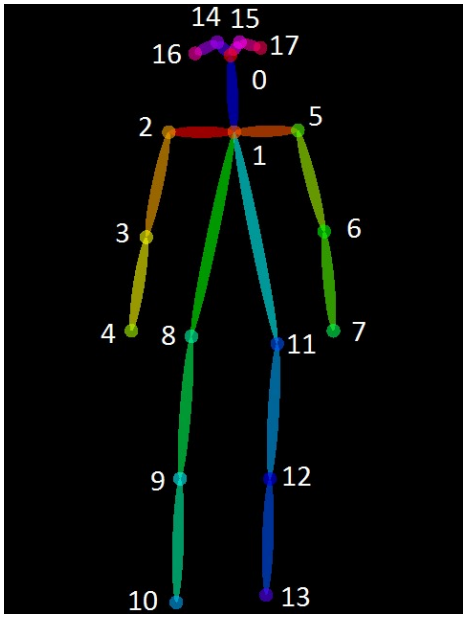
3.1 Toepassing 1: opvolgen van revalidatie na schouderoperatie

3.1.1 Bepalen van de hoek tussen arm en lichaam

We willen meten hoe ver een persoon zijn arm kan roteren voor de opvolging van de revalidatie na een schouderoperatie. Hiervoor kunnen we de hoek tussen het opperarmbeen en een ander lichaamsdeel bepalen. Op figuur 3.1 komt dat dus neer op de hoek bepalen van de lijnstukken [32] en [21] of [65] met een ander lichaamsdeel. Als we Openpose gebruiken om de positie te schatten van een persoon op een foto krijgen we als output de coördinaten van de verschillende knooppunten. Met deze coördinaten hebben we voldoende informatie om de hoek te berkenen en zo objectieve informatie te krijgen over het verloop van de revalidatie.

Bij een foto vanuit vooraanzicht kunnen we berekenen hoe ver de patiënt zijn arm zijwaarts omhoog kan brengen. Bij een foto genomen vanuit zijaanzicht kunnen we berekenen hoe ver hij de arm vooruit kan omhoog steken. Het is we

belangrijk dat de foto altijd vanuit dezelfde positie wordt getrokken omdat er anders variatie kan zijn op de hoek.

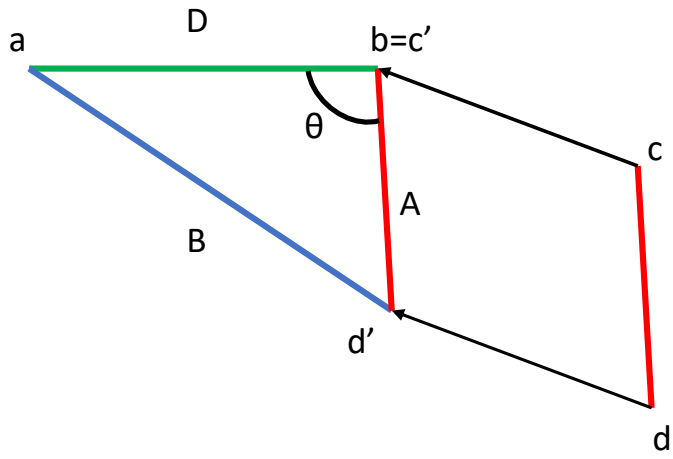


Figuur 3.1: Voorstelling van de positie bepaald via Openpose (afbeelding van openpose)

3.1.2 Voorlopige resultaten

Via Openpose krijgen we de coördinaten van alle punten die openpose kan herkennen. Hiermee kunnen we dan verder onderzoek doen. Op dit moment zijn we in staat om hoeken tussen bepaalde lichaamsdelen te berekenen met als doel objectieve gegevens te bekomen tijdens bijvoorbeeld een revalidatie na een ongeval.

Wiskundige achtergrond Om de hoek tussen twee lichaamsdelen te berekenen gebruiken we de cosinusregel. Hier volgt een klein beetje duiding over hoe we die precies gebruiken (zie figuur 3.2).



Figuur 3.2: Hoek tussen 2 lichaamsdelen

We stellen a, b de coördinaten die het eerste lichaamsdeel afbakenen en c, d de coördinaten van het tweede lichaamsdeel. De hoek tussen de lichaamsdelen is dan de θ van op de figuur. We kunnen gewoon de cosinusregel gebruiken om de hoek te bepalen als we c op b leggen. We krijgen dan

$$B^2 = A^2 + D^2 - 2 \cdot A \cdot D \cos\theta$$

en dus

$$\cos\theta = \frac{A^2 + B^2 - B^2}{2 \cdot A \cdot D}$$

met

$$A = \sqrt{(d_x - c_x)^2 + (d_y - c_y)^2}$$

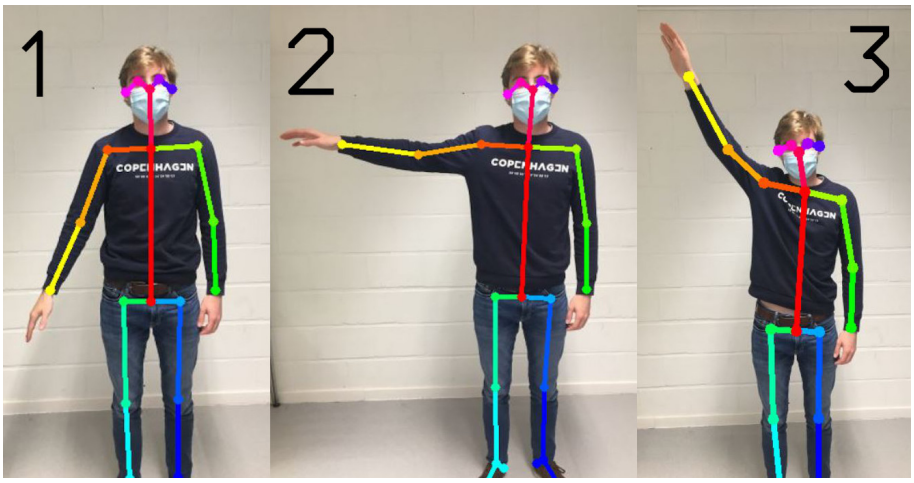
$$B = \sqrt{(b_x - a_x + d_x - c_x)^2 + (b_y - a_y + d_y - c_y)^2}$$

$$D = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2}$$

Als we dus de 4 coördinaten weten kunnen we de hoek bepalen.

Voorbeeld Een voorbeeld bij het bepalen van een hoek tussen is afbeelding 3.3. Als we de hoek bereken tussen de ruggengraat (rood) en het opperarmbeen(licht oranje) krijgen we als waarde:

- 1: 21.63 graden
- 2: 78.06 graden
- 3 130.00 graden



Figuur 3.3: Voorbeeld bij het bepalen van een hoek.

3.1.3 Voorlopige conclusies

Door deze toepassing weten we dat het relatief eenvoudig is om met openpose zinvolle berekeningen te kunnen doen. Door de coördinaten die gegeven worden als output kunnen we gemakkelijk programma's schrijven die hiermee werken.

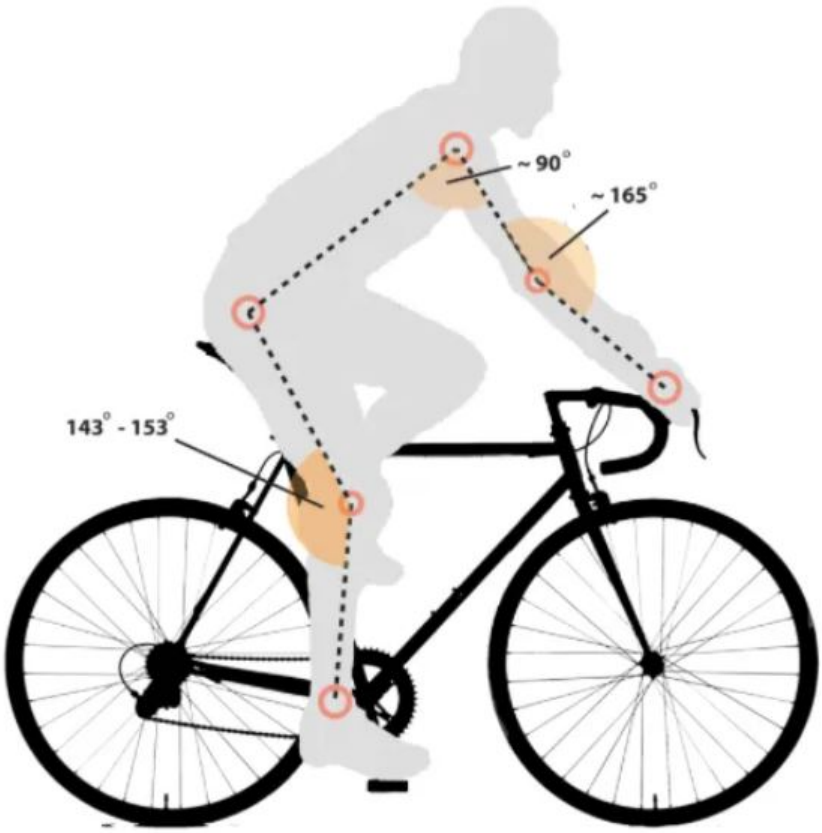
3.2 Toepassing 2: fietspositie bepalen

Voor de tweede toepassing willen we een goedkoper alternatief bieden voor een bikefitting. Hierbij worden er foto's getrokken van de persoon op de fiets en kunnen we dan op basis van de positie bepaald met openpose eventuele correcties uitvoeren op de fiets.

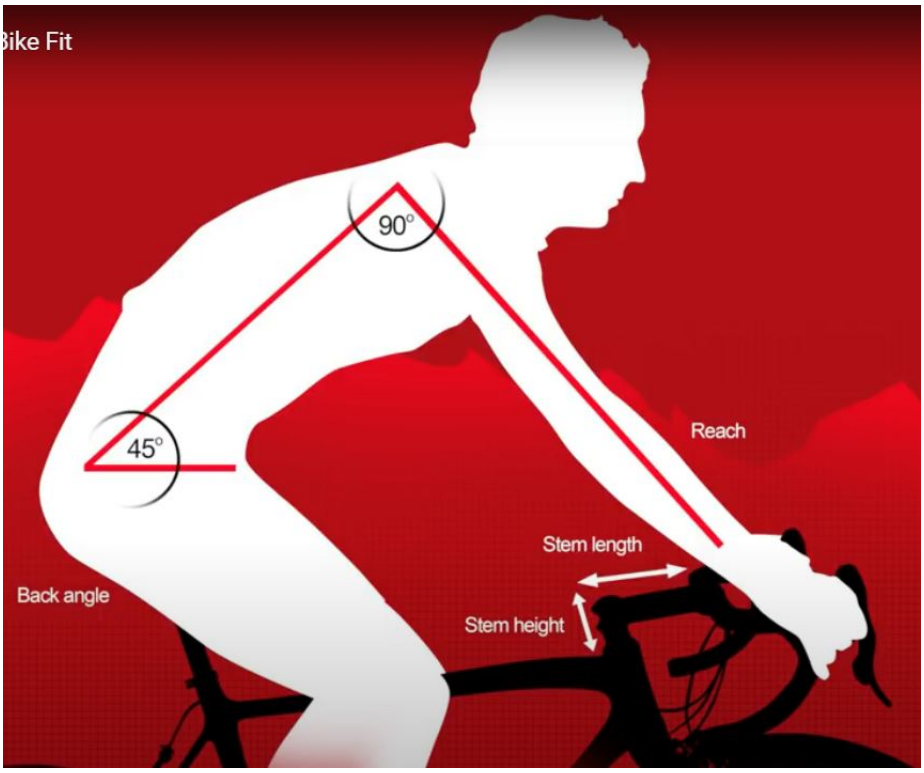
Een bikefit is eigenlijk het analyseren en eventueel aanpassen van de positie op de fiets. Het belangrijkste doel daarvan is het voorkomen van blessures. Bij wielrenners die aan competitie doen heeft een bikefit ook als doel om een zo aerodynamisch mogelijke positie op de fiets aan te nemen. Ook kan een betere positie ervoor zorgen dat men een groter wattage kan leveren op de pedalen. Alleen maar voordelen dus.

Op Figuur 3.4 staan enkele hoeken die voor de standaard persoon als optimaal gezien worden voor de positie op een koersfiets. Met behulp van Openpose zouden we deze hoeken dus kunnen berekenen aan de hand van een foto of video van de wielrenner in kwestie. Als deze hoeken te veel afwijken van wat voorop gesteld wordt, dan kunnen we een verbetering voorstellen. Dit kan een verhoging van het zadel zijn, een verandering van de lengte van de stuurpen, een verandering van de stuurhoogte... Met behulp van de coördinaten van de keypoints zouden we dan bijvoorbeeld tamelijk exact kunnen berekenen met hoeveel centimeter het zadel omhoog moet. Op Figuur 3.5 staat de optimale hoek aangeduid die de romp maakt met de horizontale. Deze hoek is voornamelijk afhankelijk van de zadelhoogte, de lengte van de stuurpen en de stuurhoogte. Ook hier kunnen we verschillende veranderingen voorstellen indien de hoek in

de praktijk te veel afwijkt van 45 °.



Figuur 3.4: Verschillende hoeken voor optimale positie op de fiets



Figuur 3.5: Optimale hoek die het bovenlichaam maakt met de horizontale

3.2.1 Voorlopige resultaten

Komt op finaal verslag.

3.2.2 Voorlopige conclusies

Komt op finaal verslag.

Hoofdstuk 4

Vakintegratie

Het is natuurlijk belangrijk om dit project een plaats te geven binnen onze opleiding ingenieurswetenschappen. We bekijken welke vakken uit de eerste 3 semesters van onze bachelor het meest gebruikt worden tijdens dit project. De belangrijkste link is met het vak begingselen van programmeren. In dit vak leerden we werken met Python en verworven we inzicht in het programmeren. Wat heel handig is, want tijdens ons project maken we veelvuldig gebruik van Python. Elk zelf gemaakt programma is geschreven in Python want Openpose heeft een uitstekende Python API. Verder maakt wiskunde ook een groot deel uit van ons project, want het berekenen van hoeken of afstanden kan niet zonder de wiskunde. We gebruiken hierbij kennis uit verschillende wiskundige vakken zoals, Analyse & calclus en Lineaire algebra. Als laatste kunnen we Statistiek ook nog linken aan dit project aangezien Openpose enkel een schatting geeft van de lichaamspositie. Het werkt met *heatmaps* en kiest per knooppunt dan het coördinaat met de hoogste kans. We gebruiken zelf geen statistische technieken, het helpt ons wel bij het begrijpen van Openpose.

Hoofdstuk 5

Planning

(gantt chart enzo)

Bibliografie

[Bulat and Tzimiropoulos(2016)] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. *CoRR*, abs/1609.01743, 2016. URL `http://arxiv.org/abs/1609.01743`.

[Cao et al.(2019)Cao, Hidalgo Martinez, Simon, Wei, and Sheikh] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

KU Leuven Kulak
Wetenschap & Technologie
Etienne Sabbelaan 53, 8500 Kortrijk
Tel. +32 56 24 60 20
isaac.venus@student.kuleuven.be