

## Informe de progrés 1

# Web de comptador automàtic de repeticions d'exercicis amb Visió per Computador

Joan Lara Formoso

16 de novembre de 2025

## 1 INTRODUCCIÓ - CONTEXT DEL TREBALL

És ben conegut que la tecnologia es troba avançant a una velocitat vertiginosa a gairebé tots els camps de la nostra vida diària, incloent els esports. Des del 'photo finish' fins al 'VAR' al futbol, la tecnologia avui dia és integral als esports professionals, especialment per a un correcte arbitratge i anàlisi posterior. És per això que existeix un potencial molt gran per implementació d'aquestes tecnologies a tot tipus d'esports menys populars als que encara no han arribat aquests mètodes.

Un dels camps on és més sorprenent que encara no s'ha arribat a estandarditzar un ús professional de la tecnologia és el fitness, que engloba aixecament de peses, l'halterofília, CrossFit i molts altres tipus de competicions similars. Per exemple, en l'arbitratge, el comptatge de repeticions es realitza encara amb mesures subjectives i no regulades, cosa que provoca problemes amb la parcialitat de l'arbitratge. És aquest el problema que es vol solucionar en aquest projecte: es tractarà de crear un comptador de repeticions estandarditzat per a exercicis de fitness amb tecnologia actual de visió per computador (Fig. 1).

En base a aquesta problemàtica, en aquest treball es buscarà desenvolupar una eina automàtica que durà a terme dues tasques: determinar automàticament l'exercici realitzat i trobar els moments exactes on es comença i on s'acaba una repetició de l'exercici. Es focalitzarà en la precisió dels càlculs i en l'ús senzill per a l'usuari d'aquesta eina. Per això, es desenvoluparà una aplicació web des d'on es podrà fer servir.

## 2 OBJECTIUS

En aquesta secció es mostren els objectius als quals es vol arribar per a la finalització del projecte. S'indica, a nivell tècnic, que se requereix per a cada tasca, i a les seccions posteriors s'explicarà la planificació i metodologia que es seguiran per dur-les a terme.

- E-mail de contacte: larafirmosjoan@gmail.com
- Menció: Tecnologies de la Informació
- Treball tutoritzat per: Coen Antens (Centre de Visió per Computador)
- Curs 2025/26



Fig. 1: Exemple de estimació de pose humana

- Revisar l'estat de l'art sobre la detecció de pose i l'aplicació a anàlisi d'exercicis. Recopilació d'informació, mètodes i selecció de models i dades a utilitzar.
- Entrenament i extracció de resultats de segmentació (YOLO11seg) .
- Entrenament i extracció de resultats de la RNN (Recurrent Neural Network) per a la predicción del tipus d'exercici.
- Construir algoritme per a comptador de repeticions.
- Creació i configuració d'API web utilitzant frameworks existents (Vue, React, Angular, etc.) amb mesures bàsiques de seguretat, com l'autenticació d'usuaris.

## 3 METODOLOGIA

Aquest projecte seguirà la metodologia SCRUM. Aquesta és una estratègia de planificació per a projectes utilitzada extensament al camp de l'enginyeria informàtica. Aquesta metodologia divideix els objectius del projecte en "sprints". Per a cada sprint, que comença i acaba en dates preestablertes, s'ha de complir amb unes tasques determinades. Al final d'aquests es fa una "sprint review", és a dir, unaavaluació de les tasques acabades així com una reavaluació de com procedir. Addicionalment, a aquest projecte es realitzarà una reunió setmanal per fer un millor seguiment i poder fer canvis menors de direcció abans de les revisions de sprint.

Per seleccionar el programa amb el qual es farà aquest seguiment, s'ha provat d'utilitzar diverses opcions: Trello, Monday i Asana. Després de realitzar la planificació amb les tres, s'ha escollit Asana com a eina. Aquesta ofereix més opcions i detalls a afegir a les tasques, com la prioritat o la data d'inici, a més de tenir una versió gratuïta del programa que s'adqua a les necessitats d'aquest projecte. Monday no ofereix una versió gratuïta, i Trello no permet els camps a les tasques que sí permet Asana, com ara la data prevista de finalització de la tasca.

## 4 PLANIFICACIÓ

Les dates dels sprints review coincideixen amb les dels informes de seguiment. Així doncs, l'informe de seguiment inclourà un seguiment de les tasques desenvolupades en el corresponent sprint, així com la direcció a seguir per al següent sprint. La data final de cada sprint és marcada per l'entrega de l'informe de seguiment corresponent; però, per a una millor gestió del temps dins de cada sprint, les principals tasques tenen dates d'entrega que poden ser més flexibles segons el transcurs del projecte.

- 15/9 - 3/10 **Kick-off del projecte.**
- 4/10 - 12/11 **Sprint 1**
- 13/11 - 10/12 **Sprint 2**
- 11/12 - 12/11 **Sprint 3**

A la figura 2 s'inclou la planificació del projecte a la plataforma Asana, on s'indiquen les diferents dates d'entrega.

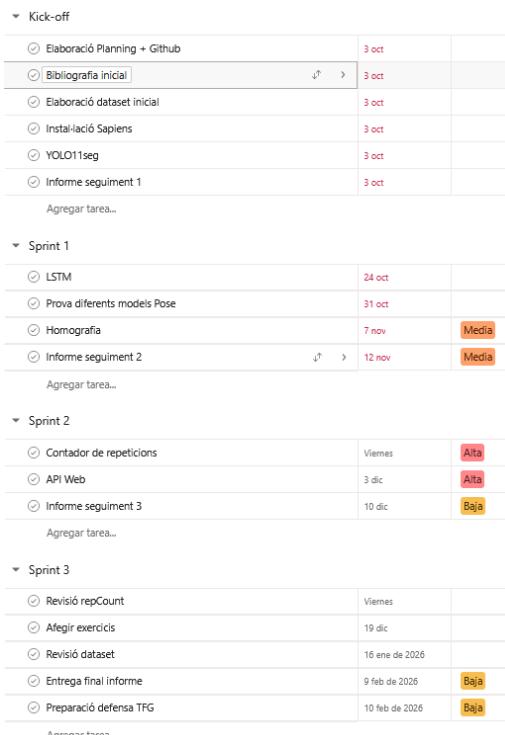


Fig. 2: Planificació Projecte a Asana

A continuació es mostra el diagrama de Gantt conseqüent de l'anterior planificació (Fig. 3). Aquest ha estat realitzat a la plataforma ProjectLibre. Aquest mostra les dependències que es troben entre les diferents tasques dins de cada sprint.

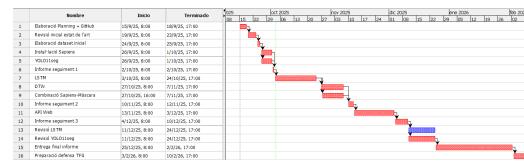


Fig. 3: Diagrama de Gantt TFG - ProjectLibre

## 5 ESTAT DE L'ART

### 5.1 Fonaments de la visió per computador

Tot i que el concepte d'intel·ligència artificial podem considerar que va néixer als anys 50 (tasques de classificació lineal) [1], el camp de la visió per computador tal com el coneixem avui dia es remunta a la invenció de les CNN (Convolutional Neural Networks) amb el model LeNet-5 [2].

Aquestes funcionen amb els coneguts com a "kernels" o filtres convolucionals. Aquests són matrius de pesos entrenables que es desplacen a través de la imatge per donar com a resultat una nova imatge a una capa inferior. Després d'atravessar les diferents capes, es connecten tots els resultats a una "fully connected layer" per obtenir un resultat de classificació, on la imatge queda classificada entre una de les opcions possibles sobre les quals la xarxa s'ha entrenat. Els pesos dels diferents filtres s'entrenen per poder extreure característiques (o "features") diferenciatives de la imatge. (Fig. 4)

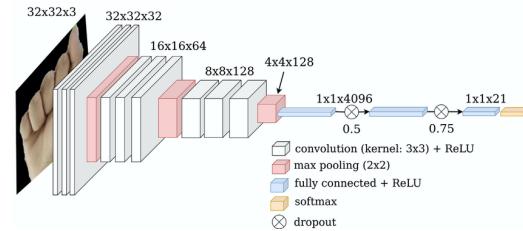


Fig. 4: Estructura Convolutional Neural Network

Aquesta és l'estructura que segueixen les CNN de classificació clàssiques com AlexNet (2012) [3], la qual va popularitzar l'ús de les CNN i de la visió per computador. Però, d'aquesta estructura bàsica de les CNN han aparegut múltiples ramificacions, com els models de Human Pose Estimation o els de segmentació de màscara, els quals s'empraran per el desenvolupament d'aquest projecte.

### 5.2 Segmentació de màscara

La segmentació de màscara és pot definir com la detecció de objectes a imatges amb precisió de píxels. No es dibuixa una caixa limitadora sobre l'objecte detectat, sinó que es determina el seu contorn píxel a píxel. Els primers models de segmentació semàntica, com FCN (2015) [4], a la seva capa "fully connected" donen un resultat a cada píxel de la imatge, que és classificat segons el tipus d'objecte. (Fig. 5) Els següents models introduïren estructures de codificació i descodificació, com U-Net [5] o DeconvNet [6]. Models més moderns com Deeplab [7] utilitzen "Atrous Convolutions", que canvien la mida dels filtres a les diferents convolucions per a tenir una millor representació de la escala dels objectes. Aquests només classifiquen cada píxel en

una classe d'objecte, però no diferencien entre els mateixos objectes de la mateixa classe (semantic segmentation).

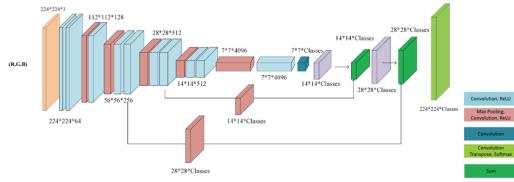


Fig. 5: Estructura Fully Convolutional Network

En canvi, els següents models, com Mask R-CNN (2017), distingueixen entre diferents instàncies del mateix objecte (instance segmentation). Aquest últim amplia la funcionalitat de Faster R-CNN, que és un model de detecció per caixa delimitadora o "bounding box" afegixen una predicción de máscara per objecte. (Fig. 6)

YOLOv11seg [8] és el model que s'utilitzarà per a aquest projecte, i és un tipus de model de "instance segmentation", igual que Mask R-CNN. Tots els models de la família YOLO (You Only Look Once), prediuen alhora caixes delimitadores per als objectes de la imatge, així com un mapa de probabilitats de les classes a les quals poden pertànyer cada una de les caixes. Des de YOLOv8 que s'afegeix una capa final de segmentació de màscara a cada caixa, sent YOLOv11 una versió millorada i més ràpida. Entre altres millors estructurals, elimina el procés de "Non-Maximum-Suppression", per fer el model més ràpid. És per la seva velocitat que YOLOv11 es selecciona com a model de segmentació de màscara a aquest projecte, per davant de models més lents que utilitzen mètodes de transformers [9] com SAM [10].



Fig. 6: Resultats Mask R-CNN

### 5.3 Human Pose Estimation

Els models de Human Pose Estimation s'especialitzen en trobar els punts d'articulació del cos humà: els colzes, estpatlls, peus, etc. per a formar un esquelet amb la seva postura. Per això necessiten detectar les coordenades de píxels a la imatge on es troben aquests articulacions, i després unir-les. (Fig. 7)

El model pioner en aquesta problemàtica es considera DeepPose[11], el qual és capaç de detectar les coordenades de les articulacions a través d'una CNN que té com a output final aquestes coordenades. Els següents models més efectius eren "Bottom-Up", és a dir, detectaven les articulacions per separat per després format l'esquelet amb aquests.

Entre aquests el més destacable és OpenPose (2017) [12].

Tot i ser aquests models efectius, els models "Top-down", que detecten primer la persona en una "bounding box" després concreten les articulacions, han esdevingut més populars per la seva major robustesa. Els models actuals es basen en aquesta metodologia. Entre aquests es troben els dos models a emprar en aquest projecte: Sapiens [13] i YOLOv11pose. YOLOv11pose és un model més ràpid basat en la estructura YOLO explicada anteriorment, mentre que Sapiens aprofita la estructura de transformers per fer un model gran entrenat amb milions de imatges d'exemple. Aquest requereix d'una computació molt més gran, però produeix resultats amb gran precisió.



Fig. 7: Exemple de pose estimation

### 5.4 Recurrent Neural Networks

Un cop s'han determinat les coordenades de les articulacions a cada fotograma, cal determinar l'acció que la persona està realitzant. Per a aquesta tasca ("Action Recognition"), s'utilitzen models RNN.

Les RNN (Recurrent Neural Networks) són un tipus different d'estructura de model de Deep Learning. Aquest prediu sobre una seqüència d'imatges, no només una sola, la predicción feta a cada frame s'enviarà cap al següent frame, depenent el resultat de tota la seqüència i no només d'un frame. La RNN a emprar és LSTM, [14] una de les primeres RNN capaç de ser entrenada per classificar diferents accions.

Existeixen altres estructures més complexes, com BiLSTM [15], però com que l'input que introduirem al model seran els keypoints de les articulacions i no el vídeo sencer, la simplicitat de LSTM s'ajusta més a aquest cas.

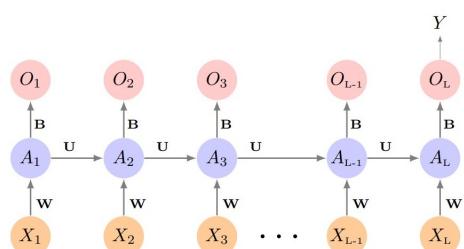


Fig. 8: Estructura RNN

## 6 DESENVOLUPAMENT

Aquest projecte es pot dividir en tres parts: la predicción automática d'exercici, el comptador de repeticions i l'eina web des d'on s'accedirà a les dues funcions anterioris.

Els diferents models s'han entrenat amb el dataset públic de Kaggle [16]. Aquest dataset recopila vídeos públics de diferents exercicis de força i atlètics.

Tota la programació és realitzada en Python a través de Visual Studio Code. Python permet utilitzar múltiples biblioteques públiques com Open CV, Ultralytics, Sapiens o PyTorch que permetran treballar amb el dataset i amb els models a entrenar.

Tot el codi, vídeos, imatges, models i prediccions s'han pujat paral·lelament a un repositori remot de GitHub, el qual s'ha anat amb cada objectiu complert.

### 6.1 Predictor d'exercici

El predictor d'exercici es pot considerar una tasca de classificació, on a partir d'un input de vídeo es prediu l'exercici realitzat, dins d'unes opcions disponibles sobre les quals el model ha estat entrenat. Els exercicis sobre els quals es pot predir, provisionalment, són: bench press, squat, deadlift i pull-up.

Per això s'utilitza una RNN (Recurrent Neural Network), la estructura de la qual s'adapta a la tasca, ja que té en compte els outputs previs per a calcular els següents. Això vol dir que per a cada imatge del vídeo, té en compte el que el model ha vist a les anteriors, així és com pot analitzar el moviment del vídeo, i no només la pose a cada imatge de forma independent.

Dins dels diversos tipus de xarxes RNN, com les LSTM (Long-Short Term Memory), BiLSTM i la fusió amb les CNN convencionals amb CNN-LSTM es treballarà amb una LSTM degut a les seves propietats de retenció de memòria superior [17].

Les seqüències d'input del model no seran directament les imatges, sinó que seran els key-points (articulacions del cos: canells, espalles, colzes...) calculats pel model de Human-Pose-Estimation, de forma més computacionalment eficient i amb menys marge d'error. [18]

Per a entrenar el model LSTM s'utilitza el model d'estimació Sapiens de Meta [13]. Aquest ofereix una precisió al nivell de l'estat de l'art, a canvi de uns requeriments de computació molt alts i una velocitat reduïda. Aquest model s'ha utilitzat per obtenir uns keypoints de pose estimation del dataset, així que la baixa velocitat no és un factor al tenir en compte, ja que no afectarà al temps d'ús de l'aplicació final. Per al correcte entrenament és preferible obtenir una màxima precisió de càlcul dels keypoints.

Respecte a la planificació original, cal afegir que el Dynamic Time Warping no ha sigut utilitzat. La predicción de keypoints és molt diferenciable entre exercicis, el model ja és capaç de classificar correctament a partir de les seqüències utilitzades sense necessitat d'estandarditzar la seva duració. A més, s'utilitzen 13 dels 17 keypoints que el model predui: s'eliminen les cames ja que a moltes seqüències, com a les de bench press, aquestes no apareixen al vídeo, ja que només s'enfoca el tronc de la persona.

### 6.2 Comptador de repeticions

Una vegada determinat el tipus d'exercici, segons aquest, es procedeix amb el comptador de repeticions, que tracta de comptabilitzar l'instant precís on es fa una repetició a l'exercici. (Fig. 9)

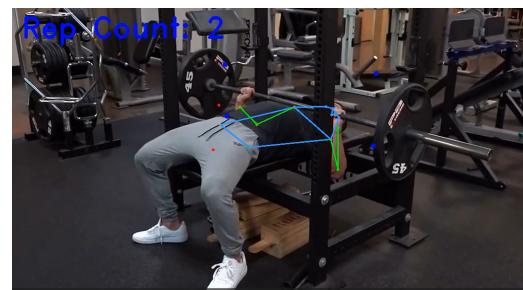


Fig. 9: Comptador de repeticions

Això s'aconsegueix a través de la Pose Estimation a cada frame, obtinguda amb YOLO11 (You Only Look Once) [8]. S'empra aquest model en comptes de Sapiens per la seva major velocitat, que permet un ús més dinàmic de l'aplicació que no és necessari per a la construcció del dataset. (10 s per frame amb Sapiens i 0.5 s amb YOLO11). Amb els resultats d'aquesta estimació, es busca un estat inicial i un estat final de l'exercici. Per exemple, en el bench press, l'estat inicial es considera quan els braços estan estirats i el final quan estan flexionats. Tenint en compte les mesures relatives de la distància dels canells als colzes, es calculen aquests estats, i es busca quan es retorna a l'estat inicial després del final.

Múltiples factors s'han de tenir en compte per a aquest procés, com els possibles errors de localització dels key-points de la Pose Estimation (especialment quan les espalles no són visibles) i la variància de posició entre les diferents seqüències. Per aquesta raó, s'ha realitzat un filtratge del dataset per tenir vídeos amb perspectives similars i sense moviment, i un filtratge dels resultats de Pose Estimation per obtenir els cossos més grans detectats i amb la major puntuació de confiança (per evitar detectar persones al fons del vídeo).

Tot i així, la millora més significativa s'ha obtingut en realitzar homografies cap a una pose de referència. Una homografia tracta de convertir els píxels on es troben els keypoints a un pla de referència. D'aquesta manera es pot estandarditzar el càlcul tot i la diferència en proporció i posició. S'agafa un pla de referència per a cada exercici marcat manualment (per exemple, pla entre canells i espalles) i tots els canells i espalles dels fotogrames dels diferents vídeos es transformen a aquest pla. Això es fa mitjançant una matriu d'homografia calculada per vídeo, amb les correspondències cap a les coordenades de referència, es troba aquest pla al nou vídeo i es transformen conseqüentment a aquest pla per als següents vídeos.

A la imatge (Fig. 10) es poden observar en blau els keypoints detectats, en vermell els que s'utilitzen com a referència de pla per calcular l'homografia, i en verd els keypoints transformats al pla de referència.

Respecte a la planificació original, abans d'empar el mètode de la homografia, s'havia plantejat utilitzar una xarxa YOLO11 de segmentació de màscara. Aquesta trobaria una referència respecte al cos, com per exemple, la barra

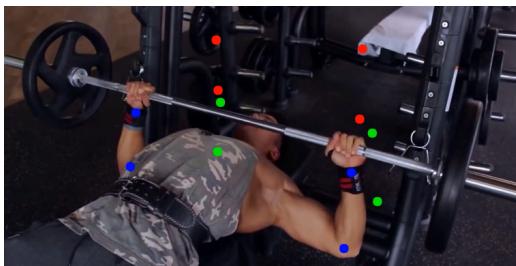


Fig. 10: Transformació de keypoints per homografia

de dominades al pull-up o la barra de peses al bench-press. Aquest mètode va ser descartat aviat, degut a la inestabilitat a l'hora de trobar aquests objectes en espais amb reflexos i oclusió (Fig. 11), a la variància entre l'aspecte d'aquests objectes en diferents vídeos, i a la poca flexibilitat que té aquest mètode quan l'angle de gravació varia. L'homografia precisament busca estandarditzar un pla de coordenades des d'on es pugui treballar amb mesures relatives des de qualsevol vídeo d'entrada.



Fig. 11: Inestabilitat a la predicción YOLO11seg (colors vermell i rosa)

## REFERÈNCIES

- [1] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. doi: <https://doi.org/10.1037/h0042519>
- [2] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, “Object recognition with gradient-based learning,” in *Shape, Contour and Grouping in Computer Vision*, pp. 319–345, Springer, Berlin, Heidelberg, 1999.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, pp. 1–9, 2012.
- [4] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, vol. 9351, Springer, Cham, 2015. doi: <https://doi.org/10.1007/978-3-319-24574-4-28>
- [6] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, vol. 8689, Springer, Cham, 2014. doi: <https://doi.org/10.1007/978-3-319-10590-1-53>
- [7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. doi: [10.1109/TPAMI.2017.2699184](https://doi.org/10.1109/TPAMI.2017.2699184)
- [8] Ultralytics, YOLO11 Models. [Online]. Available: <https://docs.ultralytics.com/es/models/yolo11/>
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... and I. Polosukhin, “Attention is All You Need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, ... and R. Girshick, “Segment Anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.
- [11] A. Toshev and C. Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1653–1660, 2014.
- [12] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2019.
- [13] Meta, Sapiens Pose Estimation. [Online]. Available: <https://github.com/facebookresearch/sapiens>
- [14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)
- [15] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *arXiv preprint arXiv:1508.01991*, 2015.
- [16] Kaggle, Workout/Exercises Video Dataset. [Online]. Available: <https://www.kaggle.com/datasets/hasyimabdillah/workoutfitness-video?resource=download>
- [17] M. Zaher, A. S. Ghoneim, L. Abdelhamid, and A. Atia, “Unlocking the potential of RNN and CNN models for accurate rehabilitation exercise classification on multi-datasets,” *Multimedia Tools and Applications*, vol. 84, pp. 1261–1301, 2025. doi: <https://doi.org/10.1007/s11042-024-19092-0>
- [18] T. Rangari, S. Kumar, P. P. Roy, D. P. Dogra, and B. G. Kim, “Video based exercise recognition and correct pose detection,” *Multimedia Tools and Applications*, vol. 81, pp. 30267–30282, 2022. doi: <https://doi.org/10.1007/s11042-022-12299-z>

- [19] M. Slupczynski, A. Nekhviadovich, N. Duong-Trung, and S. Decker, “Analyzing Exercise Repetitions: YOLOv8-Enhanced Dynamic Time Warping Approach on InfiniteRep Dataset,” in International Workshop on Sensor-Based Activity Recognition and Artificial Intelligence, *Springer Nature Switzerland*, pp. 94–110, Sept. 2024. doi: <https://doi.org/10.1007/978-3-031-80856-2>
- [20] A. Patil, D. Rao, K. Utturwar, T. Shelke, and E. Sarda, “Body posture detection and motion tracking using AI for medical exercises and recommendation system,” in *ITM Web of Conferences*, vol. 44, p. 03043, 2022. doi: <https://doi.org/10.1051/itmconf/20224403043>
- [21] Q. Yu, H. Wang, F. Laamarti, and A. El Saddik, “Deep learning-enabled multitask system for exercise recognition and counting,” *Multimodal Technologies and Interaction*, vol. 5, no. 9, p. 55, 2021. doi: <https://doi.org/10.3390/mti5090055>

## DECLARACIÓ D’ús de intel·ligència artifical generativa

Aquest document ha estat redactat **completament** per l'autor. La IA generativa, específicament Gemini 2.5 s'ha utilitzat només per indicar correccions i suggeriments de millora que s'han implementat de forma manual.

El codi elaborat i documentat al GitHub del treball ha sigut redactat **completament** per l'autor, amb ajudes de codi i llibreries públiques, com Sapiens o Ultralytics. La IA generativa, com GitHub Copilot 4.0 i Gemini 2.5 s'han utilitzat com a font de revisió, ajuda a la comprensió i a la implementació, amb redacció manual del codi final.