

Informe de progrés 2

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

Joan Lara Formoso

13 de desembre de 2025

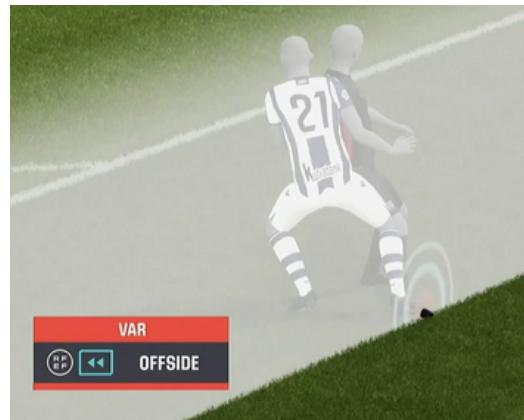


Fig. 1: Visió per Computador al esport: VAR

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

És ben conegut que la tecnologia es troba avançant a una gran velocitat a gairebé tots els camps de la nostra vida diària, incloent els esports. Des del 'photo finish' fins al 'VAR' al futbol (Fig. 1), 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 la implementació d'aquestes tecnologies a tot tipus d'esports.

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 l'aixecament de peses, exercicis d'alta intensitat, 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.

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 on-line des d'on es podrà fer servir. L'usuari podrà pujar un vídeo realitzant un exercici, i a través de la web podrà visualitzar els moments en els que es completen les repeticions, a més de poder observar un anàlisi dels càlculs i un historial d'ús.

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

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

postiors s'explicarà la planificació i metodologia que es seguiran per dur-les a terme.

- 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.
- 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, adaptada a un format unipersonal, amb un equip format per només un membre. 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, una avaluació 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 transcurr 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 plaforma Asana, on s'indiquen les diferents dates d'entrega.

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.

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 aplicen filtres convolucionals que permeten aprendre característiques d'una imatges per a després poder classificar-les. La imatge, a través de filters, atravesa diferents capes per a arribar a una capa "fully connected" la qual segons tots els resultats obtinguts, dona una classificació final (Fig. 4). Més tard apareixerien els Bounding Box Detector, els quals delimiten a una imatge on es troben els diferents objectes a detectar, com per exemple la família YOLO (You Only Look Once) [8].

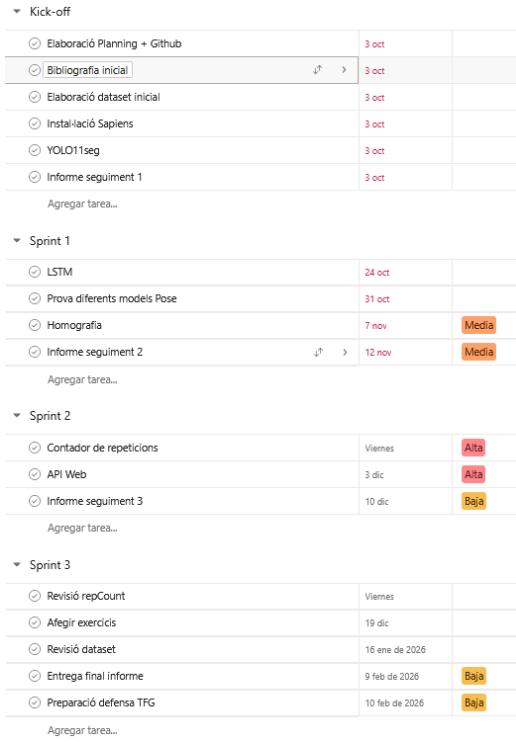


Fig. 2: Planificació Projecte a Asana



Fig. 3: Diagrama de Gantt TFG - ProjectLibre

5.2 Segmentació de màscara

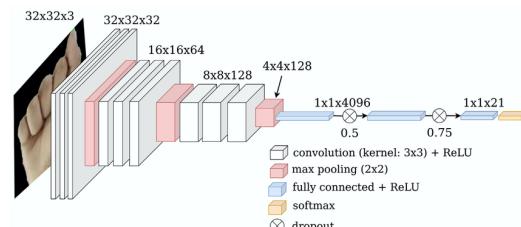


Fig. 4: Estructura Convolutional Neural Network

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.

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 introduiran estructures de codifica-

ció 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 l'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).

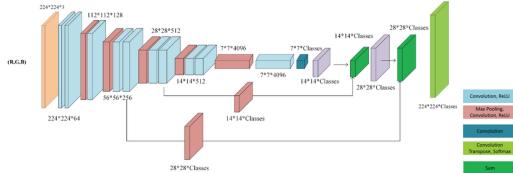


Fig. 5: Estructura Fully Convolutional Network

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", i afegeixen una predicción de màscara 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 millores 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, espatlles, 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 aquestes 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" i després concreten les articulacions, han esdevingut més populars per la seva major robustesa. Els models actuals es basen en aquesta metodologia. Entre aquests es troba el principal model a emprar en aquest projecte: YOLOv11pose.

A més, s'utilitzarà Sapiens [13] de Meta, que també és un model de estimació de pose però basat en l'estructura de transformers [9]. Aquest model ofereix uns resultats molt més precisos a canvi de ser un model molt més gran i amb una inferència molt lenta, ja que ha estat entrenat amb milions d'imatges i per tant conté una quantitat de paràmetres molt elevada.



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 diferent d'estructura de model de Deep Learning. Aquest pren sobre una seqüència d'imatges, no només una sola, la predicció 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.

6 DESENVOLUPAMENTO

Aquest projecte es pot dividir en tres parts: la predicció automàtica 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.

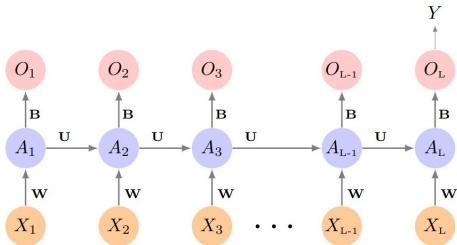


Fig. 8: Estructura RNN

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, aquesta té com a entrada una seqüència, i és capaç de classificar quin és el tipus d'aquesta seqüència. Un vídeo és una seqüència de imatges, així que la RNN classificarà el tipus d'exercici desde els vídeo sencer, observant com varia el moviment per a decidir l'exercici predit.

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.

Després d'entrenar el model, s'ha obtingut la següent matríu de confusió (Fig. 9) amb el conjunt de vídeos de test, els quals no han format part de l'entrenament. El conjunt de test està format per 6 vídeos de cada exercici, mentre que el de entrenament és de 14.

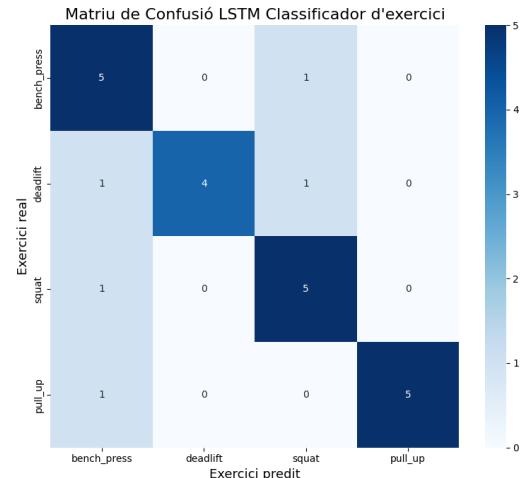


Fig. 9: Matriu de confusió LSTM

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. 10)



Fig. 10: Comptador de repeticions

Això s'aconsegueix a través de la Pose Estimation a cada frame, obtinguda amb YOLOv11 (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 YOLOv11). 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 keypoints 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).

Un dels principals problemes que afronten aquest tipus d'algoritmes que analitzen la postura humana és la varia-

bilitat de pose, dimensions i angle de càmera que hi poden haver entre diferents vídeos. La posició dels píxels detectats per la Pose Estimation no ha de variar molt entre diferents vídeos per a poder realitzar uns càlculs estandarditzats. Per a poder afegir un marge de variabilitat i transformar els keypoints de les articulacions detectades a un domini comú, s'ha optat per a utilitzar el mètode de la homografia, que transforma keypoints de qualsevol imatge a un pla vist de referència.

Per a realitzar una homografia es multiplica per una matriu d'homografia (3×3) per a transformar coordenades de una imatge ($x, y, 1$) a coordenades tridimensionals (x, y, z) referents a un pla. Aquesta matriu es calcula enllaçant certs punts d'una imatge a un altres punts dins d'un pla de referència. A la Figura 11 es pot observar com definint uns punts a transformar (vermells) i un pla de referència (punts vermells), es calcula la matriu d'homografia, i multiplicant cada píxel de la imatge per la matriu queda el resultat a la Figura 12. Es veu com la imatge queda transformada a un pla observat des de el cel, tot i que la xarxa queda mal reconstruïda ja que només es transformen els píxels de la imatge, la transformació no pot deduir com són els píxels darrere de la xarxa al no tenir-los a la vista.



Fig. 11: Selecció de punts des d'on calcular la matriu d'homografia

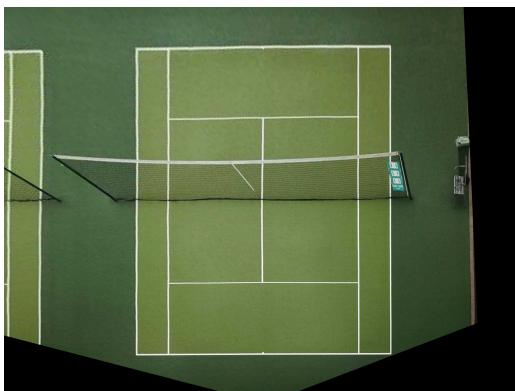


Fig. 12: Transformació de imatge per homografia

Per aplicar la homografia a aquest projecte, cal definir un pla de referència per exercici, a partir del qual es realitzaran els càlculs. S'agafa un pla de referència per a cada exercici marcat manualment (pla entre mans i espalles a press de banca, mans i peus a pes mort, etc.) i tots els keypoints equivalents als de referència que es veuran a qualsevol vídeo es

transformaran a aquest pla. La limitació d'aquest mètode és que la matriu és la mateixa per a totes les imatges d'un vídeo, per tant no hi pot haver moviment de càmera.

A la imatge (Fig. 13) 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.

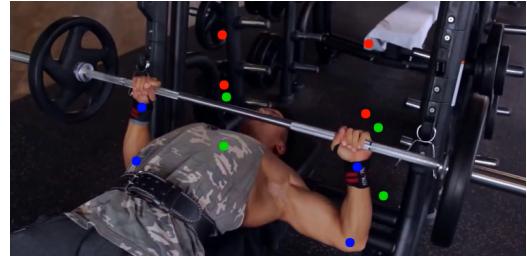


Fig. 13: Transformació de keypoints per homografia

Respecte a la planificació original, abans d'emprar 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 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. 14), 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. 14: Inestabilitat a la predicció YOLO11seg (colors vermell i rosa)

Després d'haver obtingut els keypoints transformats per l'homografia, es calculen les distàncies entre els keypoints a aquest pla (distància entre mans i espalles a press de banca, mans i peus a pes mort, espalles i mans a dominades i espalles i genolls a squat). Segons l'evolució d'aquesta distància es detecten els moments on inicia la repetició, acaba, i es comptabilitza el moment on es retorna a la posició inicial. Per a això també es tenen en compte uns valors mínims i màxims proporcionals a la posició estàtica (inicial) als quals s'ha d'arribar per a acabar i tornar a començar una repetició.

A les següents figures (Fig. 15, 16, 17, 18) es poden observar exemples de l'anàlisi de les distàncies entre keypoints a diferents vídeos de prova. Les línies puntejades grogues indiquen quan es troba la posició final i les verdes quan es retorna a la inicial. La gràfica superior indica el resultat directe del càlcul als keypoints transformats per homografia, mentres que a la d'abaix es mostra després d'una

suavització dels màxims i mínims pensada per filtrar outliers provenint d'errors de l'estimació de pose.

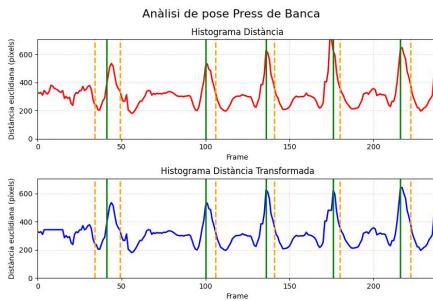


Fig. 15: Anàlisi a un exemple de press de banca

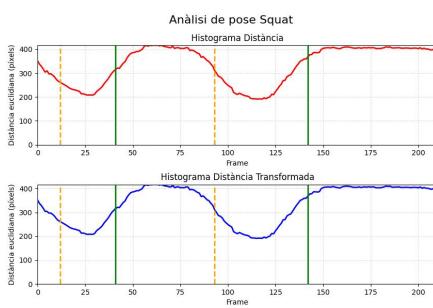


Fig. 16: Anàlisi a un exemple de squat

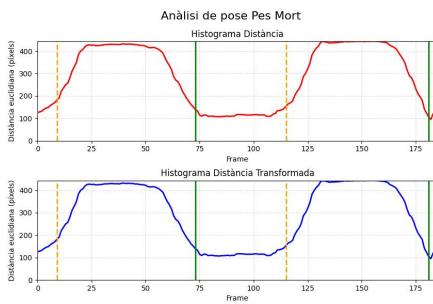


Fig. 17: Anàlisi a un exemple de pes mort

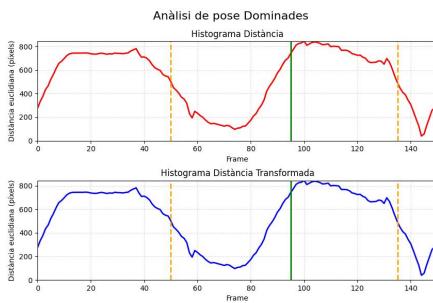


Fig. 18: Anàlisi a un exemple de dominades

6.3 API Web

Per a poder accedir a l'eina desenvolupada, s'utilitza una API web, la qual té assignada dos ports al servidor remot (a dos docker containers diferents): un per al frontend i un per al backend. El frontend mostra al navegador de l'usuari l'eina web, mentre que el backend conté les funcions a les quals el frontend accedeix. Entre aquestes funcions es troben el comptador de repeticions, el sistema de gestió d'usuaris i la gestió d'historial i descàrregues.

El frontend ha estat desenvolupat en l'entorn React Vite [22], degut a la seva flexibilitat a l'hora d'integrar eines asíncrones amb TypeScript.

El backend funciona en l'entorn FastAPI [23], el qual permet obtenir respostes del servidor de forma més ràpida respecte als seus equivalents, a més d'ofrir una programació, cosa que simplifica la integració del comptador de repeticions com a funció. A més FastAPI integra "documentació automàtica interactiva", una eina que crea automàticament documentació d'ús de les funcions i permet utilitzar-les des d'un frontend simulat amb Swagger UI (Fig. 19).. Amb /docs a la url de entrada al backend, permet provar les funcions i introduir entrades per comprovar en fase de desenvolupament les respostes que proporcionaran al frontend.

Fig. 19: Documentació automàtica interactiva de la funció de registrar usuari

La pàgina principal demana la selecció d'un vídeo per processar, i al pujar-lo es desplega de forma asíncrona dins de la mateixa pàgina un indicador de progrés. Quan la funció retorna el vídeo processat, es mostra junt a la imatge d'anàlisi de l'exercici i botons per descarregar-los (Fig. 20).

Per a poder accedir a aquesta eina, la web requereix un inici de sessió d'usuari. Les contrasenyes es guarden a la base de dades de forma segura, encriptades amb l'algorisme SHA256. Al iniciar sessió es validen usuari i contrasenya al backend, i aquest retorna al frontend un token encriptat amb caducitat de 30 minuts, necessari per a poder accedir a qualsevol funció i als arxius als qual l'usuari té accés, com les dades de perfil. La autenticació de l'usuari es fa amb l'esquema OAuth2, el qual s'encarrega de filtrar els inputs d'usuari maliciosos i de generar el token d'accés. La web també permet el canvi de dades de perfil, la de contrasenya i esborrar el compte. Totes les interaccions d'usuari amb el backend es realitzen de forma asíncrona sense recarregar la pàgina, amb les crides als endpoints del backend realitzades

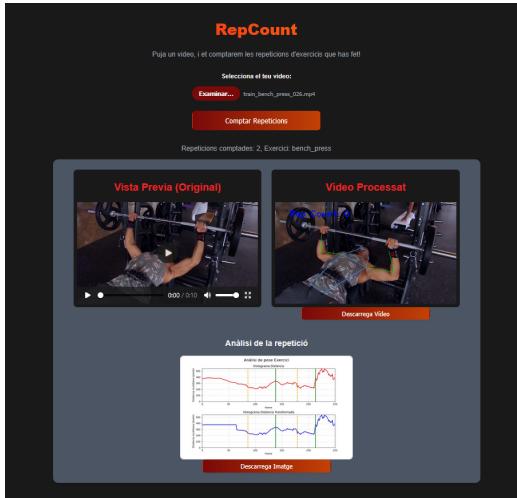


Fig. 20: Pàgina principal web

amb TypeScript.

La web emmagatzema un historial d'ús per a cada usuari. Al emprar l'eina es guarda un historial indicant la data i hora d'ús, una imatge de previsualització, el tipus d'exercici predit per la xarxa LSTM i les repeticions comptades. L'usuari pot accedir-hi i comprovar el seu historial (Fig. 21).

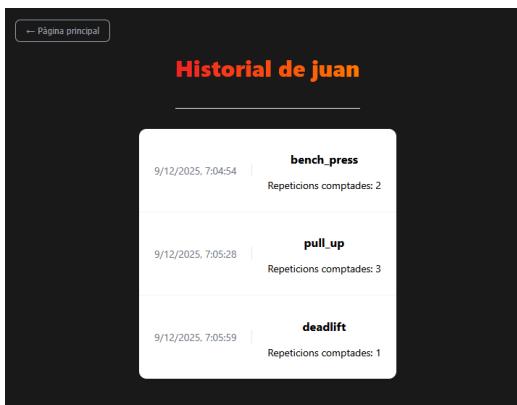


Fig. 21: Historial d'ús de l'aplicació web

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>
- [22] Vite Team, Vite: Next Generation Frontend Tooling. [Online]. Available: <https://vitejs.dev/>
- [23] F. Tiangolo, FastAPI. [Online]. Available: <https://fastapi.tiangolo.com/>

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.