

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

Joan Lara Formoso

5 de febrer de 2026

Resum— La Visió per Computador ofereix noves eines que poden aportar millores i més precisió d'anàlisi i arbitratge als esports professionals, amb mètodes com l'estimació de pose i el reconeixement automàtic d'accions. L'objectiu en aquest projecte és el d'a partir d'un vídeo d'exercici pujat a la web per l'usuari, l'eina desenvolupada sigui capaç de identificar l'exercici i comptar les repeticions que es realitzen. Per a reconèixer l'acció de l'usuari, s'utilitza una xarxa LSTM, entrenada amb el model Sapiens d'estimació de pose de Meta. Per al comptatge de repeticions s'utilitza Estimació de Pose amb YOLOv11 i una transformació homogràfica de la estimació a un pla de referència per corregir la perspectiva de la càmera. L'eina és accessible a través de un navegador, i aquesta ha estat desenvolupada en un entorn frontend React amb una API backend FastAPI. La característica principal en aquesta eina és la seva robustesa a la variació d'angle de gravació, que permet un comptatge precís amb una tolerància gran a les variacions introduïdes per l'usuari al vídeo d'entrada.

Paraules clau— Visió per Computador, Estimació de Pose, LSTM, Sapiens, YOLOv11, FastAPI, React, Comptador de Repeticions

Abstract— Computer Vision offers new tools that can add improvements and more precise analysis and refereeing to professional sports, with methods such as Pose Estimation and automatic action recognition. This project's objective is that from an exercise video uploaded to the web by the user, the developed tool is capable of identifying the exercise and counting the repetitions made. To recognize the user's action, a LSTM Network is utilized, trained by Meta's Sapiens Pose Estimation model. For repetition counting, YOLOv11 Pose Estimation is used, along with an homography transformation that converts the estimation to a reference plane, adapting the camera's angle. The tool is accessible through a browser, and has been developed using a React frontend environment and a backend API FastAPI. The project's main feature is its robustness to the camera angle variation, that allows for an accurate counting with high tolerance to variations introduced by the user in the input video.

Keywords— Computer Vision, Pose Estimation, LSTM, Sapiens, YOLOv11, FastAPI, React, Repetition Counter

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

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 ànalisi posterior. És per això que existeix un potencial molt gran a la implementació d'aquestes tecnologies a tot tipus d'esports.

Un dels camps on lentament s'està estandarditzant un ús

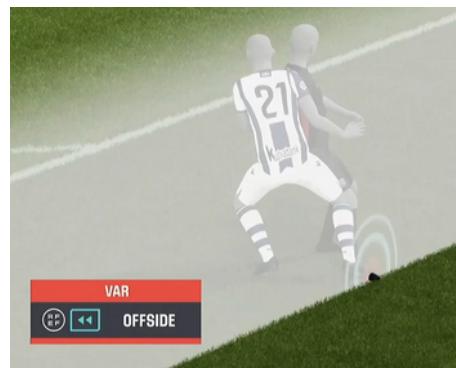


Fig. 1: Visió per Computador al esport: VAR

- E-mail de contacte: larafarmosojoan@gmail.com
- Menció: Tecnologies de la Informació
- Treball tutoritzat per: Coen Antens
- Curs 2025/26

professional de la tecnologia és el fitness, que engloba exer-

cicis aeròbics, l'aixecament de peses, CrossFit i molts altres tipus de competicions similars. Degut als alts costos d'instal·lació d'equips d'anàlisi professional, Per exemple, en molts d'arbitratges, el comptatge de repeticions es realitza encara amb mesures subjectives i no regulades (Fig. 2), cosa que provoca problemes amb la parcialitat de l'arbitratge. És aquest el problema que es vol solucionar en aquest projecte: creant un comptador de repeticions estandarditzat per a exercicis de fitness amb tecnologia de Visió per Computador.



Fig. 2: Ambigüetat al moment exacte de comptabilitzar l'exercici

En aquest treball es busca desenvolupar una eina automàtica que sigui capaç de realitzar 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 focalitza en la precisió dels càlculs i en l'ús senzill per a l'usuari d'aquesta eina.

A través d'un portal web. l'usuari pot pujar un vídeo realitzant un exercici, i a través de la web pot visualitzar els moments en els què 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 es requereix per a cada tasca, i a les seccions posteriors s'expliquen 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 d'una xarxa per a la predicción del tipus d'exercici.
- Construir un algoritme per a comptar les repeticions.
- Creació i configuració d'una API web utilitzant frameworks existents amb mesures bàsiques de seguretat, com l'autenticació d'usuaris.

3 METODOLOGIA

Aquest projecte segueix la metodologia SCRUM, adaptada a un format unipersonal. 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, durant el desenvolupament d'aquest projecte s'ha realitzat una reunió setmanal per fer un millor seguiment i poder fer canvis menors de direcció abans de les revisions de Sprint.

Per a seleccionar el programa amb el qual es fa 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 gratuita del programa que s'adqua a les necessitats d'aquest projecte. Monday no ofereix una versió gratuita, 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 inclou 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 l'apèndix es pot trobar la planificació visualitzada a la plataforma Asana, així com el diagrama de Gantt resultant de les dependències entre les tasques.

5 ESTAT DE L'ART

En aquesta secció es detalla l'evolució de la Visió per Computador i com ha arribat a donar peu a les tecnologies utilitzades per al desenvolupament d'aquest projecte.

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, atravessa diferents capes per a arribar a una capa "fully connected",

la qual, segons tots els resultats obtinguts, dona una classificació final (Fig. 3). Més tard apareixerien els detectors a nivell de caixa, 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].

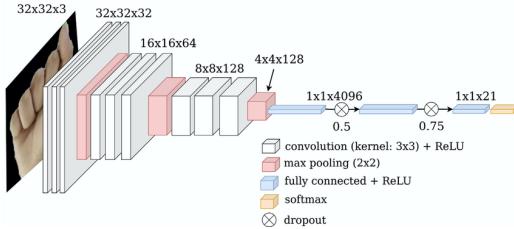


Fig. 3: Estructura Convolutional Neural Network

L'estructura mostrada a la imatge de la Figura 3 és la 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 durant el desenvolupament d'aquest projecte.

5.2 Segmentació de màscara

La segmentació de màscara es pot definir com la detecció de objectes en 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. 4) 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 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).

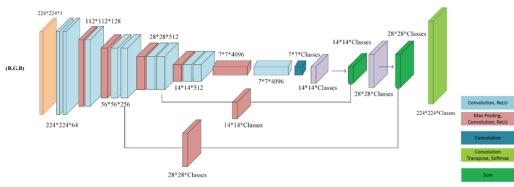


Fig. 4: Estructura Fully Convolutional Network

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. 5)

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" (NMS), per fer el model més ràpid. L'algoritme de NMS assigna la predicció amb més confiança al objecte a detectar, descartant les properes i considerant-les la mateixa detecció. Però, des de la versió 10 de YOLO, només es realitza una predicció per objecte, fent el NMS innecessari. 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], que realitzen càlculs amb milions de paràmetres.



Fig. 5: 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. 6)

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 formar l'esquelet amb aquests, entre aquests el més destacable és l'OpenPose (2017) [12].

Tot i ser aquests models efectius, els models "Top-down", que detecten primer la persona en una caixa delimitadora 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 d'estimació de pose però basat en l'estructura de transformers [9]. Aquest model ofereix uns resultats molt més precisos al ser un model entrenat amb milions d'imatges i contenir una quantitat de paràmetres molt elevada, i per aquest motiu la inferència és molt lenta.

5.4 Recurrent Neural Networks

Una de les funcions principals a desenvolupar en aquest projecte és la determinació automàtica de l'exercici realit-



Fig. 6: Exemple de pose estimation

zat. Per a aquesta tasca ("Action Recognition"), s'utilitzen models RNN.

Les RNN (Recurrent Neural Networks) són un tipus diferent de model de Deep Learning que funcionen sobre una seqüència d'objectes (de caràcters, paraules, imatges, etc.), calculant valors a cada pas de la seqüència dependent dels elements anteriors, donant com a output una predicción final que té en compte tots els elements i la relació temporal entre ells.

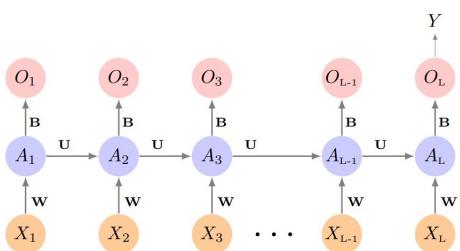


Fig. 7: Estructura RNN

En el cas aplicat al predictor d'exercici, un vídeo d'entrada es pot considerar una seqüència d'imatges. Cada predicción feta a cada frame del vídeo d'entrada es té en compte per a la predicción al següent frame, dependent el resultat de tota la seqüència i no només d'un instant.

LSTM [14] és una de les primeres RNN capaç de ser entrenada per classificar diferents accions. Aquesta funciona amb un estat de cel·la, que actua com una memòria sobre la qual es decideix conservar la informació més rellevant de l'input (Fig. 8). Posteriorment al LSTM, s'han introduït multiples variacions: com el Bi-LSTM [25], que extrau informació de la seqüència en ambdues direccions temporals, o el CNN-LSTM [26], que aplica filtres convolucionals a cada punt de la seqüència.

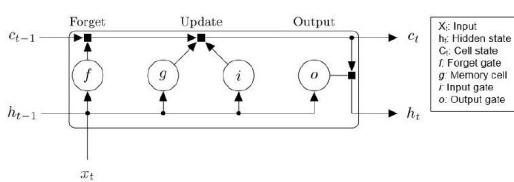


Fig. 8: Estructura LSTM

En aquest projecte, la RNN a utilitzar és una LSTM, degut a que el input a utilitzar serà directament la predicción de pose a cada imatge d'un vídeo, no la imatge de forma direc-ta. Per tant, una xarxa simple i ràpida s'adapta millor a la

predicció d'exercici, que consistirà en analitzar seqüencialment el moviment de la pose estimada en patrons repetitius per a cada exercici.

5.5 Tecnologies web

Gràcies a les tecnologies web, des de qualsevol dispositiu podem accedir a tot tipus de recursos, aplicacions i informació des d'un navegador.

Originalment, les tecnologies web requerien una recàrrega completa de la pàgina amb cada interacció de l'usuari, ja que l'aplicació web s'executava en un únic servidor i, a cada interacció, es redireccionava a un nou arxiu HTML. Amb l'arribada de la distribució backend/frontend, el servidor frontend carrega la pàgina, mentre que el backend s'encarrega de proporcionar funcions necessàries per a la interacció de l'usuari a través de APIs. El frontend utilitzava la informació rebuda per les APIs del backend amb Javascript, que funciona de forma asíncrona (sense recarregar la pàgina). És així com a través de l'API, el frontend pot interaccionar asíncronament amb les bases de dades (Fig. 9). Molts serveis web actuals utilitzen una estructura SPA (Single Page Application), que no recarreguen en cap moment, sinó que només mostren un arxiu HTML que interactua amb el backend. Exemples de webs que empren aquesta estructura són Google Maps o Netflix.

Per a seleccionar els entorns backend i frontend han sigut considerats alguns dels entorns d'ús més extens. Al backend, Django [27], Flask [28] i FastAPI [23] permeten integració directa del codi principal al treballar amb Python. Django destaca per la seva adaptabilitat per a projectes grans, tot i que és més pesat i complex. Flask destaca en la seva simpleza i facilitat d'implementació, tot i que permet menys llibertat al realitzar crides asíncrones. FastAPI ofereix tant lleugeresa com respostes asíncrones integrades, amb documentació automàtica per ajudar a la implementació i validació automàtica de dades amb Pydantic. Degut a les seves característiques, l'entorn d'API utilitzat en aquest projecte és FastAPI.

Per a l'entorn backend s'han considerat els entorns React [31], Vue [29] i Angular [30]. Els requisits a prioritzar són una integració simple, adaptada a una petita escala i amb una bona integració per a mostrar elements de forma asíncrona amb TypeScript (variació de JavaScript). Angular ofereix un framework complex i amb molta flexibilitat d'implementació. Pel contrari, Vue és un framework d'implementació més simple, tot i que menys documentat i amb una integració amb menys opcions amb Typescript. Finalment, la opció escollida és React, el qual no és un framework, sinó una llibreria JavaScript que permet més llibertat amb l'ús de TypeScript, ja que no té cap estructura definida pel framework.

6 DESENVOLUPAMENT

El projecte es pot dividir en tres parts: la predicció automàtica d'exercici, el comptador de repetitions 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.

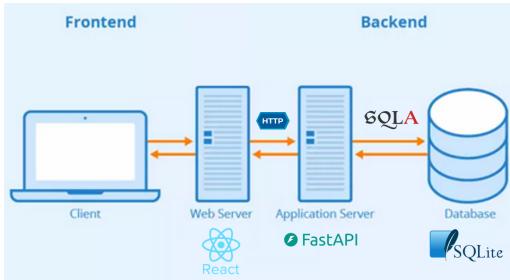


Fig. 9: Estructura frontend-backend web

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 són: press de banca, squat, pes mort i dominades (Fig. 10).

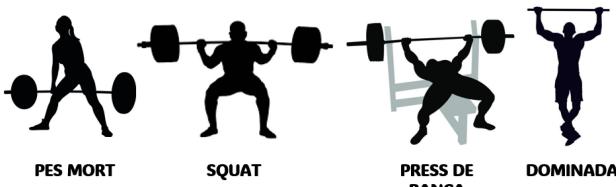


Fig. 10: Exercicis sobre els quals la xarxa LSTM ha estat entrenada

Per això s'utilitza una LSTM, que funciona com un classificador de seqüències. Aquesta classificarà els vídeos pujats a la web per l'usuari, utilitzant els vídeos com seqüències d'imatges. Llavors, aquesta xarxa LSTM podrà classificar quin dels exercicis es realitza en el vídeo, observant no només una imatge, sinó com evoluciona el moviment a les diferents imatges que componen el vídeo.

Les seqüències d'input del model no seran directament les imatges, sinó que seran els keypoints en format COCO [32]. Els keypoints són les ubicacions a cada imatge de les diferents articulacions del cos (canells, genolls, colzes...) que calcula el model d'estimació de pose. Utilitzar directament aquests keypoints és més computacionalment eficient per predir el tipus d'exercici que utilitzar la imatge sencera. La xarxa ja tindrà indicada la pose de la persona, d'aquesta forma s'elimina el soroll que aporta la resta d'informació de la imatge [18].

Per entrenar el model LSTM s'utilitza el model d'estimació Sapiens de Meta [13]. Sapiens ofereix una precisió al nivell de l'estat de l'art, a canvi d'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.

6.2 Extracció de pose

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. 11)



Fig. 11: Comptador de repeticions

Això s'aconsegueix a través de la estimació de pose 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 segons per frame amb Sapiens i 0.5 segons amb YOLO11). Amb els resultats d'aquesta estimació, es busca un estat inicial i un estat final de l'exercici. Per exemple, en el press de banca, 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 espatlles 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 sense moviment de càmera ni parts del cos ocultades, 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).

6.2.1 Transformació de keypoints per homografia

Un dels principals problemes que afronten aquest tipus d'algoritmes que analitzen la postura humana és la variabilitat de pose, dimensions i angle de càmera que hi pot haver entre diferents vídeos. La posició dels píxels es detecta sobre la imatge, i no tenen en compte la profunditat. Si un mateix exercici es grava des de dos angles diferents, la posició dels keypoints canviaria tot i realitzar la mateixa acció. Per a poder afegir un marge de variabilitat, es transformen els keypoints de les articulacions detectades a un domini comú, un pla de referència, des d'on es poden fer càlculs estandarditzats. S'ha optat per a utilitzar el mètode de l'homografia, que transforma keypoints en qualsevol pla de la imatge a un pla comú, a través d'una matriu de homografia.



Fig. 12: Transformació de punts per homografia

Per realitzar una homografia es multiplica per una matriu d'homografia (3×3) per a transformar coordenades a qualsevol pla de la imatge (x, y, z) a coordenades al pla de referència ($x, y, z=1$) referents al pla que es forma al visualitzar la imatge a una pantalla. Aquesta matriu es calcula enllaçant punts a un pla d'una imatge a un altres punts dins del pla de referència on es vol transformar. El procés de càlcul de la matriu es realitza al primer frame, i posteriorment tots els keypoints als següents frames es calcularan sobre aquesta matriu.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \simeq \begin{pmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

A la Figura 13 es pot observar com definint uns punts a transformar (vermell) 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 14. Es veu com la imatge queda transformada a un pla observat des del cel, tot i que la xarxa queda mal reconstruïda ja que només es transformen els píxels de la imatge, la transformació no pot recrear com són els píxels darrere de la xarxa al no tenir-los a la vista.



Fig. 13: Selecció de punts des d'on calcular la matriu d'homografia

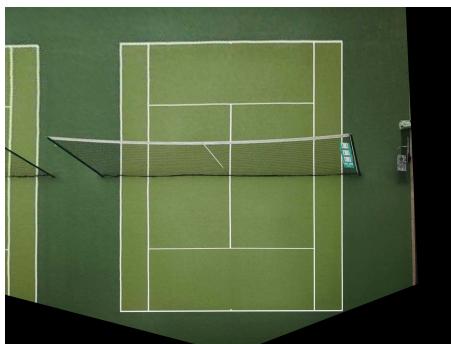


Fig. 14: Transformació de imatge per homografia

Per aplicar l'homografia a aquest projecte, cal definir com es transformarà al pla de referència per a cada exercici. S'agafen unes coordenades dins del 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. Una 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.

Respecte a la planificació original, abans d'emprar el mètode de l'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 o la barra de peses al press de banca. Aquest mètode va ser descartat aviat, degut a la inestabilitat a l'hora de trobar aquests objectes en espais amb reflexos i oclusió (Fig. 15), 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 qualsevol vídeo d'entrada a un pla de referència.



Fig. 15: Inestabilitat a la predicció YOLO11seg (colors vermell i rosa)

A la Figura 16 es pot observar el pipeline complet que segueix l'aplicació, des del vídeo d'entrada fins l'output amb el comptador de repeticions i l'anàlisi del moviment.

6.3 API Web

Per poder emprar el comptador de repeticions desenvolupat, s'utilitza una API web, la qual té assignada dos ports al servidor remot (a dos entorns virtuals 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 respuestes del servidor de forma més ràpida respecte als seus equivalents, a més d'ofrir una programació en Python, 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. 17). Amb /docs a la url de entrada al backend, permet provar les funcions i introduir entrades per comprovar en

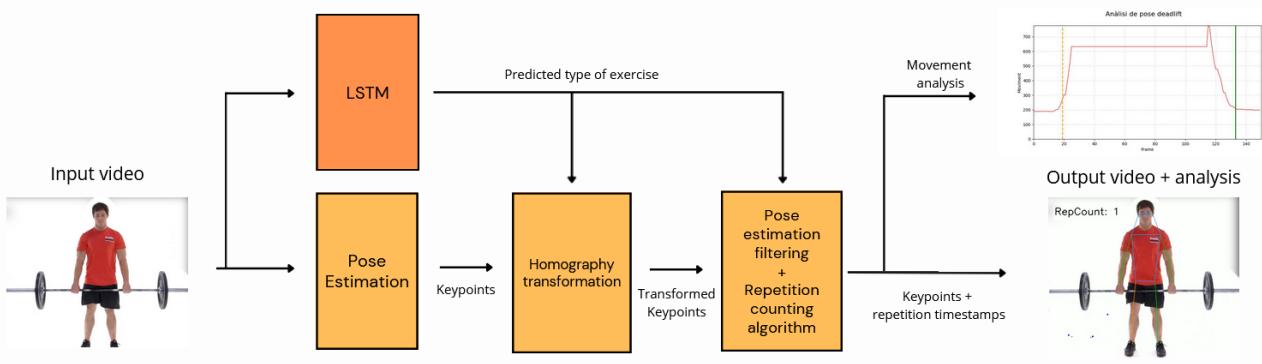


Fig. 16: Pipeline complet del comptador de repeticions

fase de desenvolupament les respostes que proporcionaran al frontend.

La base de dades sobre la qual es treballa és SQLite, la qual a través de la llibreria integrada amb Python SQLAlchemy permet interacció amb l'entorn FastAPI. En aquesta base de dades s'emmagatzemen els usuaris registrats a la web, amb les seves dades de perfil, i el seu historial d'ús de l'aplicació, amb detalls sobre cada execució.

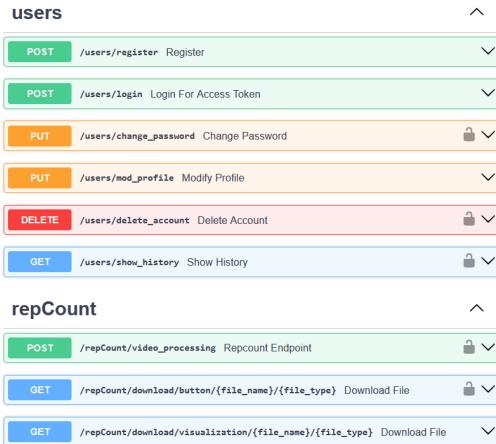


Fig. 17: Documentació automàtica interactiva de les diferents funcions del backend

7 RESULTATS

A continuació es mostren els resultats obtinguts, tant la classificació d'exercici, el comptador de repeticions i l'aplicació web final que permet interactuar amb l'eina.

7.1 Classificació d'exercici

Després d'entrenar el model LSTM, s'ha obtingut la matriu de confusió (Fig. 18) 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 vídeos.

El model és capaç de predir de forma correcta aproximadament un 85% dels exercicis al conjunt de test. Els errors restants s'atribueixen principalment a grans variacions de l'angle de gravació al conjunt d'entrenament i de test, que introduceixen grans variacions entre seqüències.

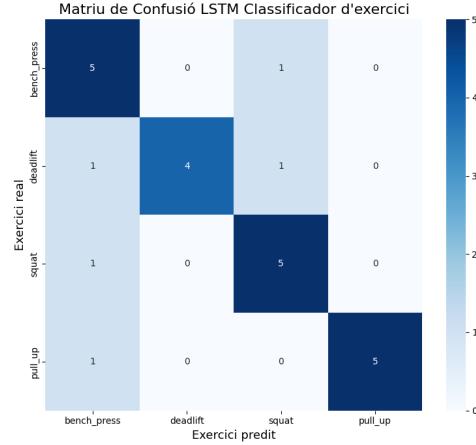


Fig. 18: Matriu de confusió LSTM

7.2 Comptador de repeticions

A la imatge de la figura 19 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.

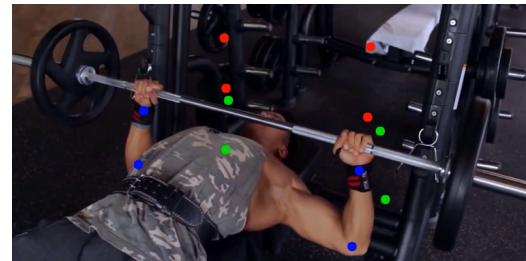


Fig. 19: Transformació de keypoints per homografia

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 proporcional 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. 20, 21, 22, 23) 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 l'homografia, mentres que a la de sota es mostra després d'una suavització dels màxims i mínims pensada per filtrar outliers provenint d'errors de l'estimació de pose.

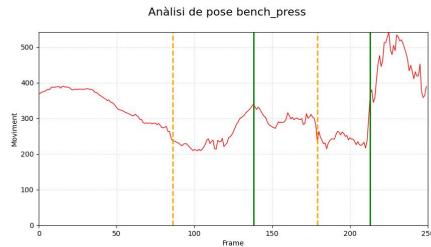


Fig. 20: Anàlisi a un exemple de press de banca

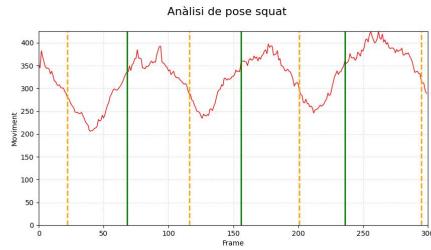


Fig. 21: Anàlisi a un exemple de squat

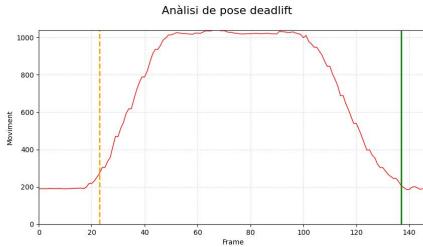


Fig. 22: Anàlisi a un exemple de pes mort

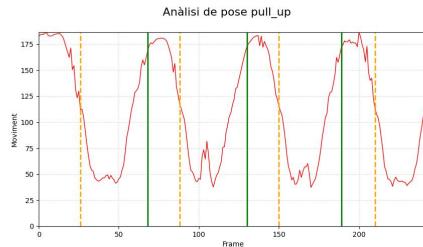


Fig. 23: Anàlisi a un exemple de dominades

Gràcies a l'ús de YOLOv11 per a l'estimació de pose i d'utilitzar keypoints com a entrada de la xarxa LSTM, s'obté una velocitat mitjana de 32.7 frames per segon (FPS), és a dir, frames que es processen a cada segon. Per a vídeos de 30 FPS o menys, la durada del procés és menor que la del vídeo. Si es té en compte la generació del vídeo de sor-

tida, amb la estimació de pose i el comptador de repeticions representats, s'obtenen 19.5 FPS com a mitjana.

7.3 Ús de l'API Web

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. 24).

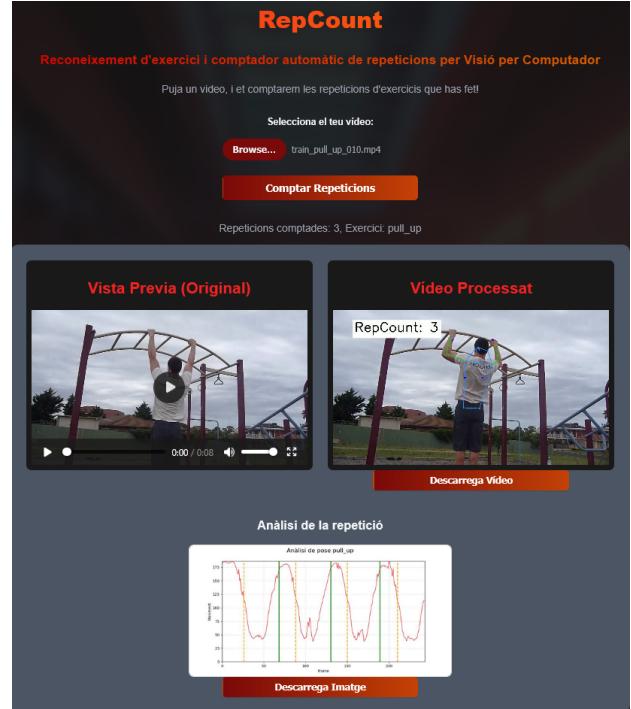


Fig. 24: Pàgina principal web

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. A l'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 maliciós 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 amb TypeScript.

Per poder mostrar una barra de progrés al processar el vídeo pujat, s'activa una crida al backend que crea un WebSocket. Aquest s'empra per transmetre els missatges de progrés activats al processament del vídeo d'una forma molt més ràpida i que no requereix peticions constants des del frontend.

La web emmagatzema un historial d'ús per a cada usuari. 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. 25).



Fig. 25: Historial d'ús de l'aplicació web

8 TREBALL FUTUR I CONCLUSIONS

La característica prioritizada en el desenvolupament de l'eña és la robustesa cap a diferents angles de càmera. El mètode utilitzat per calcular els instants de les repeticions té en compte pocs factors, només el moviment vertical que es realitza al dur a terme un exercici. Això, tot i que permet establir un mètode senzill per generalitzar a molts angles de càmera, no realitza un anàlisi profund, només comptabilitza les repeticions. Així que, per a altres tipus d'ús més complexos, com l'anàlisi de postura, càlculs addicionals com d'angle entre articulacions podrien ser necessaris, a costa d'una pèrdua de generalització.

Els treballs futurs per millorar la funció desenvolupada, a més d'afegir diferents funcionalitats com poder analitzar més tipus d'exercicis, podrien incloure una base de dades d'entrenament del LSTM més gran o bé l'ús de models basats en transformers. També hi hauria potencial per integrar el comptador de repeticions a una aplicació mòbil, la qual podria ser utilitzada de forma personal o de forma professional per a anàlisi d'exercicis a competicions.

Gràcies a la ràpida evolució dels models d'intel·ligència artificial, ja existeixen nous models molt recents que milloren els models emprats a l'inici d'aquest projecte. Per exemple, al novembre de 2025, SAM3 de Meta [24] es va llançar, oferint segmentació de màscara a través d'una petició de text o "prompt" amb una precisió al nivell de píxels (Fig. 26). Aquest mètode es podria integrar afegint o substituint la estimació de pose per trobar amb precisió la posició relativa de les articulacions o les peses.

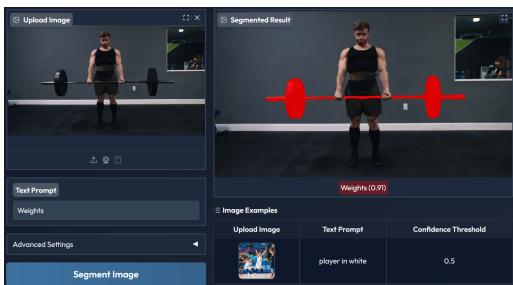


Fig. 26: Exemple d'ús de SAM3

Actualment, l'algoritme desenvolupat presenta una forma d'aplicar l'estimació de pose per al comptatge de repeticions d'exercicis de força, amb una alta capacitat de gene-

ralització per a qualsevol angle de càmera del vídeo d'entrada, sense necessitat d'especificar l'exercici realitzat i en un entorn web de fàcil l'ús per a l'usuari.

AGRAÏMENTS

Voldria donar les gràcies al tutor d'aquest treball, Coen Antens, per haver-me donat l'oportunitat de realitzar aquest treball i per haver-ho tutoritzat amb seguiments setmanals. Així com també m'agradaria agraïr al Centre de Visió per Computador de la Universitat Autònoma de Barcelona per oferir els seus servidors per al desenvolupament d'aquest projecte, i al departament d'Enginyeria Informàtica de la pròpia universitat per haver-me ofert la possibilitat de realitzar aquest treball de forma remota durant el meu intercanvi internacional.

DECLARACIÓ D'ÚS D'INTEL·LIGÈNCIA ARTIFICIAL GENERATIVA

Aquest document ha estat redactat **completament** per l'autor. La IA generativa, específicament Gemini 3.0 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 estat 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 3.0 s'han utilitzat com a font de revisió, ajuda a la comprensió i a la implementació, amb redacció manual del codi final.

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. 8699, 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
- [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
- [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/>
- [24] N. Carion, L. Gustafson, , Y. T. Hu, S. Debnath, Hu, R. Suris, D. ... and C. Feichtenhofer, "Sam 3: Segment anything with concepts," *arXiv preprint arXiv:2511.16719*.
- [25] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [26] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, LLong-term recurrent convolutional networks for visual recognition and description,in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.
- [27] Django Software Foundation, Django. [Online]. Available: <https://www.djangoproject.com/>
- [28] Pallets Projects, Flask. [Online]. Available: <https://flask.palletsprojects.com/>
- [29] E. You, Vue.js. [Online]. Available: <https://vuejs.org/>
- [30] Google, Angular. [Online]. Available: <https://angular.dev/>
- [31] Meta Platforms, Inc., React. [Online]. Available: <https://react.dev/>
- [32] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, vol. 8693, Springer, Cham, 2014. doi: <https://doi.org/10.1007/978-3-319-10602-1-48>

APÈNDIX

A.1 Planificació i diagrama de Gantt

A la Figura 27 s'inclou la planificació del projecte a la plataforma Asana, on s'indiquen les diferents dates d'entrega.

Kick-off			
<input checked="" type="checkbox"/> Elaboració Planning + Github	3 oct		
<input checked="" type="checkbox"/> Bibliografia inicial	3 oct		
<input checked="" type="checkbox"/> Elaboració dataset inicial	3 oct		
<input checked="" type="checkbox"/> Instal·lació Sapiens	3 oct		
<input checked="" type="checkbox"/> YOLO11seg	3 oct		
<input checked="" type="checkbox"/> Informe seguiment 1	3 oct		
Aregar tarea...			
Sprint 1			
<input checked="" type="checkbox"/> LSTM	24 oct		
<input checked="" type="checkbox"/> Prova diferents models Pose	31 oct		
<input checked="" type="checkbox"/> Homografia	7 nov	Media	
<input checked="" type="checkbox"/> Informe seguiment 2	12 nov	Media	
Aregar tarea...			
Sprint 2			
<input checked="" type="checkbox"/> Contador de repeticions	Viernes	Alta	
<input checked="" type="checkbox"/> API Web	3 dic	Alta	
<input checked="" type="checkbox"/> Informe seguiment 3	10 dic	Baja	
Aregar tarea...			
Sprint 3			
<input checked="" type="checkbox"/> Revisió repCount	Viernes		
<input checked="" type="checkbox"/> Afegir exercicis	19 dic		
<input checked="" type="checkbox"/> Revisió dataset	16 ene de 2026		
<input checked="" type="checkbox"/> Entrega final informe	9 feb de 2026	Baja	
<input checked="" type="checkbox"/> Preparació defensa TFG	10 feb de 2026	Baja	
Aregar tarea...			

Fig. 27: Planificació Projecte a Asana

Es mostra el diagrama de Gantt conseqüent de l'anterior planificació (Fig. 28). Aquest ha estat realitzat a la plataforma ProjectLibre. Aquest mostra les dependències que es troben entre les diferents tasques dins de cada Sprint.

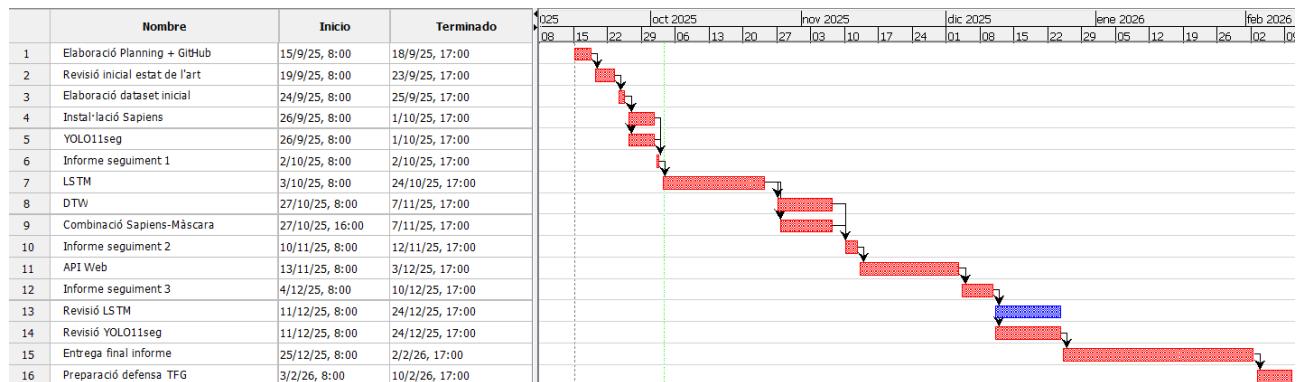


Fig. 28: Diagrama de Gantt Projecte - ProjectLibre