

UNIVERSITATEA BABEȘ-BOLYAI  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
SPECIALIZAREA INFORMATICĂ

## LUCRARE DE LICENȚĂ

**Evaluarea execuției exercițiilor  
fizice pe baza posturii corpului  
utilizând rețele neuronale**

Coordonator științific  
Prof. dr. Czibula Gabriela

Student  
Călin Anda-Maria

2025

## **Abstract**

The subject of this thesis is the development of a web-based application that integrates artificial intelligence for the purpose of evaluating the execution of physical exercises. The project addresses a growing need for intelligent, user-friendly tools that assist individuals in performing exercises correctly, especially in contexts where access to professional supervision is limited.

The thesis is structured in three main chapters. The first chapter presents the theoretical foundations of posture evaluation and biomechanical correctness, along with an overview of the deep learning models commonly used in this domain. The second chapter introduces the proposed methodology, detailing the architecture of three AI models: a convolutional neural network (CNN) for static images, a CNN-LSTM hybrid for video sequences, and a gradient boosting classifier (XGBoost) based on biomechanical features extracted with MediaPipe. The third chapter is focused on the software implementation, describing both the frontend (ReactJS) and backend (FastAPI) components and their integration with the AI models.

The main contributions of this thesis consist in designing and implementing an application capable of evaluating exercise execution based on AI models. The originality of the project lies in its ability to support different input formats (images and videos), offer both visual and numerical feedback, and integrate multiple analysis methods within a single scalable system. Moreover, the application was tested using real data and demonstrates high accuracy in classifying exercise performance quality.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

Călin Anda-Maria

Cluj-Napoca  
15-06-2025

# Cuprins

<b>Listă de figuri</b>	<b>2</b>
<b>Listă de tabele</b>	<b>4</b>
<b>Introducere</b>	<b>6</b>
<b>1 Fundamente teoretice</b>	<b>8</b>
1.1 Context teoretic . . . . .	8
1.1.1 Importanța corectării posturii în timpul exercițiilor fizice . . . .	8
1.1.2 Anatomia și biomecanica posturii corpului uman . . . . .	9
1.1.3 Studii și publicații anterioare . . . . .	10
1.1.3.1 Corectarea posturii cu YOLOv7-pose . . . . .	11
1.1.3.2 Corectare posturală cu OpenPose . . . . .	13
1.1.3.3 Detectarea posturilor-cheie cu CNN . . . . .	15
1.2 Inteligența artificială în domeniul evaluării posturii . . . . .	18
1.2.1 Tehnici de interpretare a poziției corpului în mișcare . . . . .	19
1.2.1.1 Definiția și principiile detectării posturii . . . . .	19
1.2.1.2 Sisteme de evaluare a execuției exercițiilor pe baza ero- rilor . . . . .	20
1.2.2 Rolul rețelelor neuronale în evaluarea execuției exercițiilor . . .	21
<b>2 Abordarea propusă</b>	<b>26</b>
2.1 Metodologie . . . . .	26
2.1.1 Colectarea și preprocesarea datelor de antrenament . . . . .	26
2.1.2 Descrierea arhitecturii modelelor și a procesului de antrenare pentru clasificare . . . . .	27
2.1.2.1 Clasificarea pe imagini statice cu CNN . . . . .	27
2.1.2.2 Clasificarea pe secvențe video cu CNN-LSTM . . . . .	30
2.1.2.3 Clasificarea imaginilor pe baza landmark-urilor folo- sind XGBoost . . . . .	32
2.1.3 Metode de evaluare a performanței modelului . . . . .	34
2.2 Rezultate experimentale și analiza acestora . . . . .	35
2.3 Îmbunătățiri ulterioare . . . . .	40

<b>3</b>	<b><i>FitPose AI</i> - Aplicația software</b>	<b>42</b>
3.1	Descrierea sistemului . . . . .	42
3.2	Analiză și proiectare . . . . .	43
3.3	Implementare . . . . .	45
3.3.1	Frontend . . . . .	45
3.3.2	Backend . . . . .	47
3.4	Manual de utilizare . . . . .	48
3.5	Extinderi posibile . . . . .	52
	<b>Concluzii</b>	<b>54</b>

# Listă de figuri

1.1	Poziționarea corpului în raport cu axa verticală . . . . .	9
1.2	Detecția punctelor cheie articulare utilizând modelul YOLOv7-pose [KKDT23] . . . . .	11
1.3	Evoluția unghiurilor articulare în timp pentru genuflexiuni [KKDT23] . . . . .	12
1.4	Evoluția unghiurilor articulare în timp pentru ridicări laterale pentru umeri [KKDT23] . . . . .	13
1.5	Exemple de feedback vizual pentru execuția genuflexiunii analizate cu <i>OpenPose</i> [Boy21] . . . . .	14
1.6	Diagrame de flux care descriu pașii sistemului propus [Pan22]. . . . .	15
1.7	Structura rețelei convoluționale propuse de Pan pentru extragerea caracteristicilor posturale din cadrele video selectate [Pan22]. . . . .	16
1.8	Comparație între metoda propusă și alte metode tradiționale privind acuratețea recunoașterii posturii sportive [Pan22]. . . . .	17
1.9	Flux de prelucrare IA pentru evaluarea posturii corporale . . . . .	19
1.10	Cele 33 de puncte anatomice ( <i>landmark-uri</i> ) MediaPipe Pose [SARG23] . . . . .	20
1.11	Arhitectura unei rețele CNN cu straturi convoluționale, pooling și complet conectate . . . . .	23
1.12	Arhitectura CNN-LSTM pentru clasificarea secvențială a posturilor în videoclipuri . . . . .	24
1.13	Funcționarea algoritmului XGBoost . . . . .	25
2.1	Arhitectura CNN pentru clasificarea imaginilor statice . . . . .	28
2.2	Arhitectura CNN-LSTM pentru clasificarea în secvențe video . . . . .	31
2.3	Matrice de confuzie pentru modelul CNN – Barbell row . . . . .	36
2.4	Matrice de confuzie pentru modelul CNN+LSTM – Overhead Press . . . . .	37
2.5	Matricea de confuzie pentru modelul XGBoost . . . . .	38
3.1	Diagrama de cazuri de utilizare pentru aplicația FitPose AI . . . . .	43
3.2	Aplicarea principiului <i>Separation of Concerns</i> în arhitectura aplicației . . . . .	44
3.3	Fluxul de interacțiune între componentele frontend în React . . . . .	46
3.4	Fluxul backend – selecție și inferență model AI în <i>FitPose AI</i> . . . . .	48
3.5	<b>Home Page</b> - dropdown pentru aplicația FitPose AI . . . . .	49
3.6	<b>Home Page</b> - butonul "Load File" pentru aplicația FitPose AI . . . . .	49
3.7	<b>Încărcarea imaginii</b> pentru exercițiul Barbell Row . . . . .	50
3.8	<b>Încărcarea imaginii</b> - previzualizare pentru exercițiul Barbell Row . . . . .	50
3.9	<b>Încărcarea videoclipului</b> pentru exercițiul Squat . . . . .	51

3.10	<b>Încărcarea videoclipului</b> - alertă pentru exercițiul Squat . . . . .	51
3.11	Mesaje de feedback afișate în funcție de rezultatul predicției . . . . .	52

# Listă de tabele

2.1	Raport de clasificare pentru modelul CNN - Barbell Row . . . . .	35
2.2	Raport de clasificare pentru modelul CNN+LSTM – Overhead Press .	37
2.3	Raport de clasificare pentru modelul XGBoost - Barbell Row . . . . .	38
2.4	Comparația performanțelor celor trei modele de clasificare . . . . .	39

# Introducere

În ultimii ani, progresele din domeniul inteligenței artificiale au permis dezvoltarea unor sisteme capabile să execute sarcini din ce în ce mai complexe în arii precum sănătatea, educația și sportul. Unul dintre domeniile în care IA și-a demonstrat aplicabilitatea practică este analiza și evaluarea posturii corporale în timpul activităților fizice. Capacitatea de a detecta automat execuția incorectă a exercițiilor oferă multiple beneficii, de la prevenirea accidentărilor până la optimizarea performanței și facilitarea antrenamentelor personalizate, fără supravegherea directă a unui antrenor.

Creșterea interesului pentru activitățile de antrenament desfășurate acasă, alături de extinderea soluțiilor tehnologice în domeniul sportului, a generat o nevoie tot mai mare pentru sisteme inteligente care să sprijine utilizatorii în executarea corectă a exercițiilor fizice. În acest context, modelele de viziune computerizată bazate pe învățare profundă, în special *rețelele neuronale convoluționale* (CNN), s-au dovedit eficiente în interpretarea datelor vizuale și în clasificarea modului de realizare a exercițiilor ca fiind corect sau incorect. Mai mult, prin analiza secvențială a imaginilor sau a videoclipurilor, aceste modele pot identifica abateri de la modelele biomecanice recomandate și pot furniza feedback în timp real.

Lucrarea de față propune un sistem bazat pe tehnici de inteligență artificială pentru detectarea execuției incorecte a exercițiilor fizice, concentrându-se pe trei mișcări fundamentale: genuflexiune, presă deasupra capului și ramat cu bara. Aceste exerciții au fost selectate datorită gradului ridicat de complexitate biomecanică și importanței posturii în execuția lor corectă. Pentru fiecare mișcare, a fost utilizat un set de date etichetate, ce conține atât exemple corecte cât și incorecte, adnotate pe baza unor erori comune, precum proiecția excesivă a genunchilor, pierderea curburii lombare sau alinierea incorectă a coatelor.

Sistemul propus integrează un model de învățare profundă antrenat pentru a clasifica date vizuale (imagini sau videoclipuri) în funcție de corectitudinea executării mișcărilor. Aplicația este completată de o interfață web prietenoasă, dezvoltată în *React*, care permite utilizatorilor să încarce fișiere pentru evaluare și să primească feedback instant. Scopul principal este de a oferi un instrument care să contribuie la practicarea în siguranță a exercițiilor fizice, mai ales în cazul începătorilor sau al celor care nu au acces la supraveghere de specialitate.

Lucrarea este structurată în trei capitole principale. Capitolul 1 oferă o fundamente teoretică privind importanța corectării posturii în timpul exercițiilor fizice, anatomia implicată și tehnologiile bazate pe inteligență artificială utilizate în acest scop. Capitolul 2 descrie metodologia propusă pentru dezvoltarea modelului AI, incluzând



procesul de colectare și preprocesare a datelor, arhitectura rețelei neuronale utilizate și analiza performanței modelului. În Capitolul 3 este prezentată aplicația software *FitPose AI*, detaliind componentele sistemului, aspectele de implementare, precum și posibilitățile de extindere ulterioară. Lucrarea se încheie cu evidențierea concluziilor obținute și a direcțiilor de dezvoltare viitoare.

**Utilizarea instrumentelor IA generative** În timpul elaborării acestei lucrări, autoarea a folosit *ChatGPT* ca instrument de sprijin în redactarea și optimizarea codului sursă al aplicației, precum și în generarea unor imagini schematice și reformularea unor pasaje textuale pentru coerență. După utilizarea acestui instrument, autoarea a verificat, testat și adaptat codul generat și își asumă întreaga responsabilitate pentru implementarea finală.

# Capitolul 1

## Fundamente teoretice

Scopul acestui capitol este să introducă principalele concepte utilizate în cadrul lucrării, oferind astfel o bază teoretică necesară înțelegerii problemei corectării și evaluării posturii și a rolului acestora în contextul exercițiilor fizice. Capitolul este structurat astfel: Secțiunea 1.1.1 discută importanța corectării posturii în timpul exercițiilor fizice și efectele negative ale unei posturi incorecte asupra sănătății și performanței fizice. Secțiunea 1.1.2 prezintă aspecte fundamentale privind anatomia și biomecanica posturii corpului uman. Secțiunea 1.1.3 oferă o sinteză a studiilor și publicațiilor anterioare relevante pentru domeniul abordat în lucrare. Ulterior, în Secțiunea 1.2 este explorată utilizarea inteligenței artificiale în evaluarea și corectarea posturii. Sunt prezentate tehnici de analiză a poziției corpului în mișcare (Secțiunea 1.2.1) și modalități de evaluare a execuției exercițiilor pe baza erorilor (Secțiunea 1.2.1.2), precum și rolul rețelelor neuronale în această abordare (Secțiunea 1.2.2). Scopul final al capitolului este de a fundamenta teoretic soluția propusă în această lucrare pentru detectarea automată a posturii corecte în timpul exercițiilor fizice, prin utilizarea tehnicilor de procesare video și rețele neuronale.

### 1.1 Context teoretic

#### 1.1.1 Importanța corectării posturii în timpul exercițiilor fizice

Adoptarea unei posturi corecte în timpul exercițiilor fizice este un factor esențial pentru prevenirea accidentărilor, pentru maximizarea eficienței mișcărilor și pentru menținerea sănătății sistemului musculo-scheletal. Corectitudinea posturii influențează modul în care sunt distribuite forțele în corp, impactând în mod direct echilibrul, stabilitatea și solicitarea articulațiilor și a mușchilor. O aliniere corectă contribuie la activarea adecvată a grupelor musculare, reduce tensiunile neuniforme și permite o execuție controlată și precisă a mișcărilor.

Postura incorectă, în schimb, poate cauza o serie de probleme, cum ar fi dezechilibre musculare, compresii articulare, tensiuni cronice și chiar traumatisme de suprasolicitare. În cazul persoanelor active sau al sportivilor de performanță, aceste disfuncții

pot conduce la scăderea randamentului, întreruperea antrenamentelor și apariția unor perioade lungi de recuperare. Chiar și în cazul exercițiilor cu intensitate moderată, menținerea unei posturi greșite repetate în timp poate avea efecte negative semnificative asupra sănătății coloanei vertebrale, a genunchilor sau a umerilor.

În acest context, corectarea posturii devine un obiectiv prioritar în cadrul programelor de antrenament, al sesiunilor de kinetoterapie și în orice activitate fizică organizată. Tehnologia modernă permite astăzi identificarea și corectarea posturii în timp real, fie prin senzori, fie prin analiză video automatizată. Feedback-ul imediat ajută utilizatorul să-și conștientizeze poziția corporală și să corecteze greșelile de execuție, prevenind astfel accidentările și îmbunătățind eficiența mișcării.

Integrarea inteligenței artificiale în acest proces aduce un plus semnificativ de precizie și automatizare. Prin algoritmi de recunoaștere vizuală și învățare automată, sistemele moderne pot analiza postura unui utilizator și pot oferi recomandări personalizate în funcție de deviațiile detectate. Astfel, se creează premisele pentru dezvoltarea unor aplicații accesibile și eficiente, care pot susține atât amatorii, cât și sportivii profesioniști în realizarea exercițiilor într-o manieră corectă și sigură.

### 1.1.2 Anatomia și biomecanica posturii corpului uman

Postura corpului uman este rezultatul unui echilibru fin între elementele structurale și funcționale ale organismului, incluzând sistemul osos, sistemul muscular, sistemul nervos și țesuturile conjunctive. Aceste componente acționează împreună pentru a menține alinierea segmentară a corpului și pentru a permite executarea eficientă a mișcărilor, cu minim de stres asupra articulațiilor și țesuturilor. Anatomia posturală presupune poziționarea corectă a capului, trunchiului, pelvisului și membrilor în raport cu axa verticală a corpului, cu scopul de a menține o postură neutră și echilibrată [Bud10].

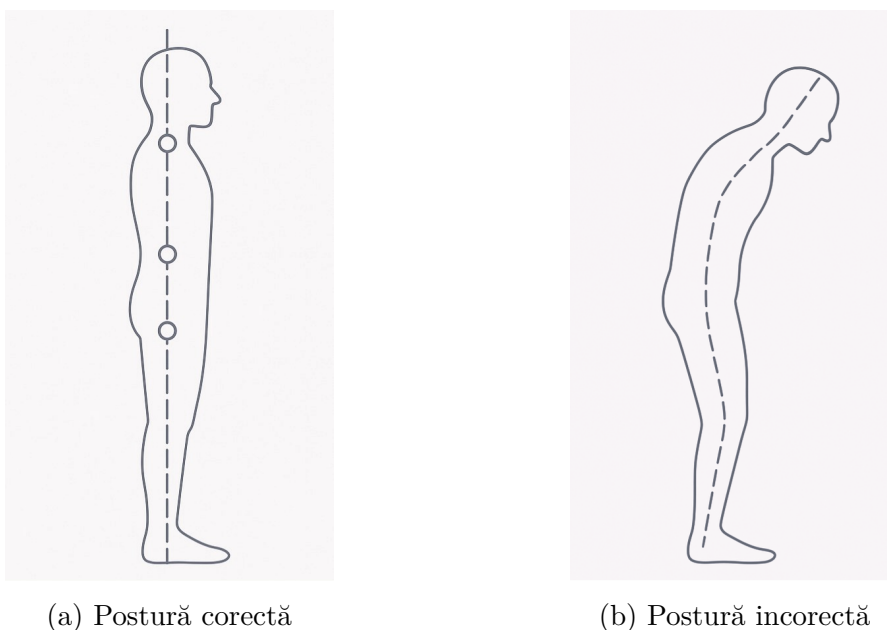


Figura 1.1: Poziționarea corpului în raport cu axa verticală

Din punct de vedere anatomic, postura este susținută de structuri precum coloana vertebrală, centura pelviană și centura scapulară, care oferă suport și mobilitate corpului. Coloana vertebrală, în special, joacă un rol crucial, deoarece trebuie să asigure atât stabilitatea cât și flexibilitatea, permițând mișcările naturale și protejând măduva spinării. Poziționarea corectă a acesteia – cu curburi fiziologice păstrate (lordoză cervicală și lombară, cifoza toracală) – este esențială pentru distribuirea echilibrată a greutății corporale.

În context biomecanic, postura este privită ca o stare dinamică ce implică adaptări continue ale corpului la solicitări interne și externe. Biomecanica posturii studiază interacțiunea dintre forțele generate de mișcare, reacțiile solului și gravitația, analizând vectorii de forță, centrele de greutate și momentele articulare. Prin aceste analize se identifică factorii perturbatori ai echilibrului postural și se stabilesc măsuri pentru restabilirea acestuia. O postură deficitară este deseori asociată cu suprasolicitarea anumitor grupe musculare și cu compensări nesănătoase care pot duce la afecțiuni cronice.

Un alt aspect important este coordonarea neuromusculară, care asigură răspunsuri rapide la modificările poziției corpului. Sistemul nervos central, creierul și măduva spinării, colaborează cu receptorii proprioceptivi localizați în mușchi și articulații pentru a menține echilibrul și stabilitatea posturală. Această relație dintre sistemul nervos și aparatul locomotor este vitală în menținerea unei posturi funcționale.

Din punct de vedere muscular, stabilitatea posturală este asigurată de mușchii trunchiului, inclusiv mușchii abdominali profunzi, mușchii paravertebrali, fesierii, ischiogambierii și cvadricepsii. Aceștia lucrează împreună pentru a păstra coloana vertebrală într-o poziție neutră și pentru a permite mobilitatea fără pierderea echilibrului. Dezechilibrele musculare, cum ar fi slăbirea gluteilor sau rigiditatea hamstringilor, pot modifica aliniamentul corporal și pot genera patologii posturale precum hiperlordoza, scolioza sau cifoza [KTB 22].

Evaluarea biomecanică a posturii implică utilizarea unor tehnologii moderne precum analiza video 2D/3D, platformele de forță, senzori de presiune sau camere de profunzime. Aceste metode permit o cuantificare precisă a devierilor posturale și oferă un suport obiectiv în diagnostic și intervenție [PIE21]. În ultimii ani, integrarea acestor tehnologii în aplicații mobile și platforme de inteligență artificială a revoluționat modul în care postura este monitorizată și corectată, atât în mediul clinic, cât și în cel sportiv.

În concluzie, o înțelegere aprofundată a anatomiei și biomecanicii posturii corpului uman constituie fundamentul intervențiilor eficiente pentru corectarea posturii. Aceasta oferă baza științifică necesară dezvoltării unor soluții inteligente, adaptate nevoilor individuale ale utilizatorilor.

### 1.1.3 Studii și publicații anterioare

Studierea posturii și a corectării acesteia în contextul activităților fizice a fost un subiect abordat frecvent în literatura de specialitate, mai ales odată cu dezvoltarea tehnologiilor digitale și a interesului crescut pentru prevenția accidentărilor. În lucrările

tradiționale din domeniul kinetoterapiei și al fizioterapiei, accentul este pus pe identificarea dezechilibrelor musculare și pe aplicarea de exerciții corective personalizate. Studiile recente însă explorează din ce în ce mai mult rolul tehnologiilor emergente în acest proces.

### 1.1.3.1 Corectarea posturii cu YOLOv7-pose

Unul dintre cele mai recente și cuprinzătoare studii aplicate în domeniul corectării posturii este realizat de Kotte [KKDT23], *Real-Time Posture Correction in Gym Exercises: A Computer Vision-Based Approach for Performance Analysis, Error Classification and Feedback*, în care autorii propun o arhitectură de tip *intelligent tutoring system* ce utilizează **YOLOv7-pose** pentru urmărirea articulațiilor și analiza execuției exercițiilor fizice în timp real.

Arhitectura propusă se bazează pe mai multe componente esențiale. În primul rând, se utilizează YOLOv7-pose (You Only Look Once), un model avansat de detecție a punctelor cheie ale corpului, derivat din YOLOv7, capabil să identifice 17 articulații esențiale, așa cum este ilustrat în Figura 1.2.

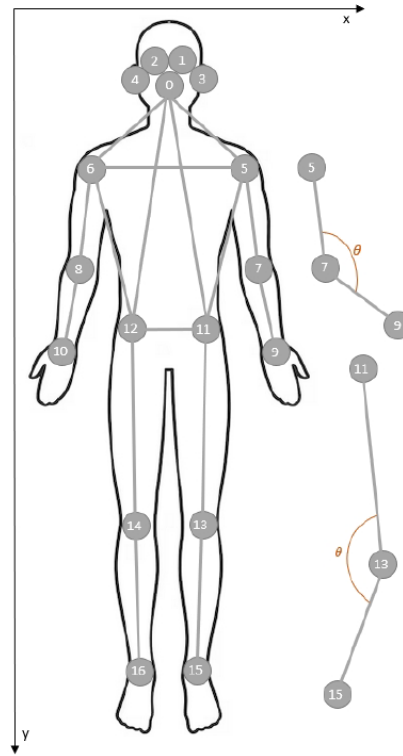


Figura 1.2: Detecția punctelor cheie articulare utilizând modelul YOLOv7-pose [KKDT23]

Pe baza acestor puncte, se calculează unghiurile articulare formate în jurul fiecărei articulații. Aceste unghiuri sunt determinate prin utilizarea a trei puncte  $(u, v, p)$ , conform relațiilor geometrice dintre vectori în planul cartezian. Formula utilizată este:

$$\theta(u, v, p) = \frac{180}{\pi} \cdot \frac{\phi(y_p - y_v, x_p - x_v) - \phi(y_u - y_v, x_u - x_v)}{\pi} \quad (1.1)$$

unde funcția  $\phi(y, x)$  este definită astfel:

$$\phi(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right), & \text{dacă } x > 0 \\ \frac{\pi}{2} - \arctan\left(\frac{x}{y}\right), & \text{dacă } y > 0 \\ -\frac{\pi}{2} - \arctan\left(\frac{x}{y}\right), & \text{dacă } y < 0 \\ \arctan\left(\frac{y}{x}\right) \pm \pi, & \text{dacă } x < 0 \\ \text{nedefinit}, & \text{dacă } x = 0 \text{ și } y = 0 \end{cases}$$

Fiecare exercițiu analizat are asociată o plajă ideală de unghiuri articulare, definite ca hiperparametri. Acestea sunt determinate prin analiza video a demonstrațiilor realizate de antrenori profesioniști (precum cei din rețeaua McFIT), reprezentând standardul de execuție corectă față de care sunt comparate mișcările utilizatorului. Pe baza acestor comparații, sistemul generează un *feedback vizual personalizat*, evidențiind în timp real articulațiile sau segmentele corporale poziționate incorect. Utilizatorul este astfel ghidat prin sugestii de corectare afișate direct pe interfață, ceea ce contribuie la îmbunătățirea execuției exercițiului fără intervenție umană suplimentară.

Evaluarea sistemului propus a fost realizată pe un grup de cinci utilizatori fără experiență anterioară în exerciții de forță, utilizând ca referință de comparație videoclipuri profesionale cu execuții corecte. Rezultatele obținute au indicat o acuratețe ridicată în detectarea posturii corecte, ceea ce confirmă robustețea modelului de clasificare în condiții reale. Participanții au perceput sistemul ca fiind intuitiv și util, oferind un feedback imediat asupra erorilor de execuție. În special, pentru exerciții precum genuflexiunile, împinsul de la umeri și fandările, modelul CNN+LSTM a demonstrat performanțe notabile în identificarea abaterilor de la forma ideală. Această observație este susținută și de analiza grafică a variațiilor unghiulare dintre punctele cheie ale corpului, ilustrată în Figura 1.3 și Figura 1.4. Acolo se poate observa cum unghiurile dintre articulații evoluează pe parcursul execuției exercițiilor, în funcție de cadru. Linia roșie evidențiază traseul postural, permițând identificarea momentelor-cheie și a eventualelor erori sistematice. Astfel, graficele contribuie la înțelegerea detaliată a modului în care postura utilizatorului se modifică în timp și oferă o bază obiectivă pentru validarea automată a execuției exercițiilor fizice.

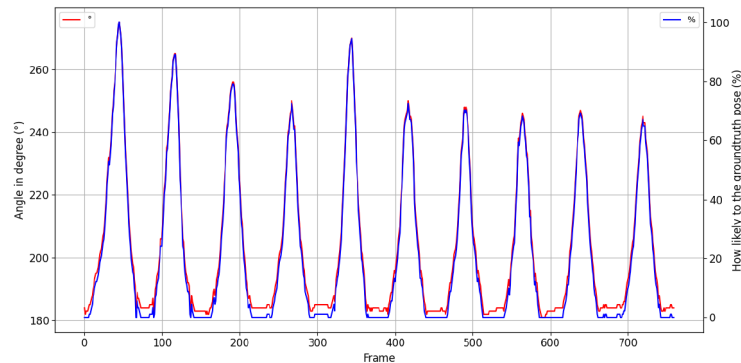


Figura 1.3: Evoluția unghiurilor articulare în timp pentru genuflexiuni [KKDT23]

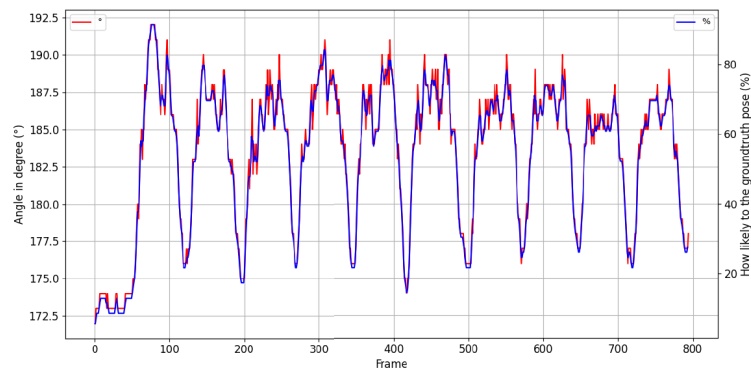


Figura 1.4: Evoluția unghiurilor articulare în timp pentru ridicări laterale pentru umeri [KKDT23]

Interfața grafică a aplicației a fost concepută pentru a fi intuitivă și *user friendly*, oferind un panou de control care include o bară de progres, un contor de repetări și o reprezentare vizuală sub forma unui schelet suprapus peste imaginea utilizatorului, cu evidențierea articulațiilor executate incorect. Acest tip de feedback vizual în timp real contribuie la o învățare mai eficientă și la corectarea imediată a erorilor. De asemenea, este în curs de dezvoltare integrarea cu fluxuri video live de la webcam, care ar permite analiza execuției în timp real. În plus, se are în vedere implementarea unui sistem de feedback audio pentru a spori nivelul de interactivitate și implicare al utilizatorului în timpul antrenamentelor.

Această lucrare demonstrează aplicabilitatea viziunii computerizate moderne în corectarea automată a posturii și în antrenamentul asistat de AI. Utilizarea **YOLOv7-pose** împreună cu tehnici de învățare transferabilă permite o integrare eficientă în aplicații reale fără nevoia de reantrenare extinsă. Este un exemplu relevant de sistem scalabil, aplicabil atât în medii educaționale, cât și în antrenamente personale.

### 1.1.3.2 Corectare posturală cu OpenPose

Lucrarea lui Boyalla [Boy21], *Real-time Exercise Posture Correction Using Human Pose Detection Technique*, propune o soluție de corectare a posturii în timp real utilizând modelul *OpenPose* antrenat cu seturile de date *COCO* și *MPII*. Obiectivul lucrării este dezvoltarea unui sistem capabil să identifice în mod automat abaterile posturale în timpul execuției genuflexiunilor prin analizarea unghiurilor formate între șold, genunchi și gleznă.

Modelul *OpenPose* este antrenat pe două seturi de date standard, *COCO* și *MPII*, și este implementat folosind framework-ul **Caffe**. În cadrul etapei de evaluare cu *COCO*, sistemul este testat pe exercițiul *genuflexiune*, cu focus pe calculul unghiurilor articulare pentru analiza corectitudinii execuției. Detecția punctelor cheie ale corpului permite extragerea coordonatelor articulațiilor, în special ale șoldului, genunchiului și gleznei, pentru fiecare picior. Pe baza coordonatelor celor trei puncte, se calculează unghiul articular format între segmentele  $\vec{ba}$  și  $\vec{bc}$ , utilizând funcția `getAngle()`. Aceasta implementează o metodă geometrică bazată pe funcția `atan2`, care determină unghiul

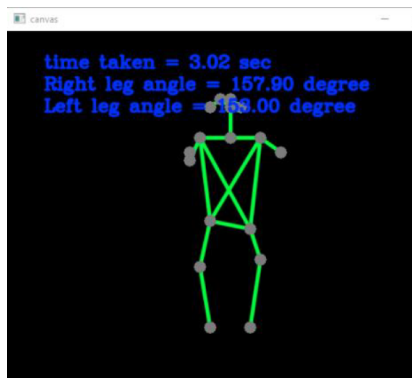
relativ în planul cartezian și returnează rezultatul în grade, conform relației:

$$\theta = [\arctan 2(y_c - y_b, x_c - x_b) - \arctan 2(y_a - y_b, x_a - x_b)] \cdot \frac{180}{\pi} \quad (1.2)$$

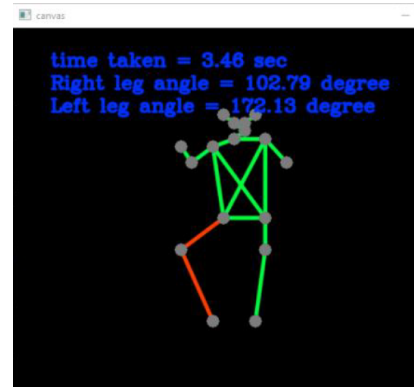
În cazul în care  $\theta$  este negativ, acesta este transformat în echivalentul său pozitiv prin adăugarea a  $360^\circ$ , pentru a păstra rezultatul în intervalul  $[0^\circ, 360^\circ)$ .

În secțiunea următoare a lucrării, autorul analizează cazurile în care punctele cheie nu sunt complet detectate în imagine, o situație frecvent întâlnită în datele provenite din setul *MP11*. Dacă, spre exemplu, glezna nu este vizibilă în cadrul analizat, sistemul atribuie un unghi invalid cu valoarea  $-1$ , semnalând imposibilitatea evaluării posturii pe acel segment. Această abordare permite excluderea automată a articulațiilor nesigure din procesul de feedback, contribuind la robustețea metodei.

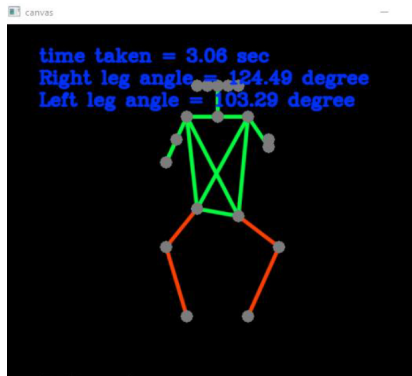
Pentru reprezentarea vizuală a poziției utilizatorului, sistemul generează o diagramă tip stick-figure colorată: dacă unghiurile calculate se încadrează într-o marjă de eroare de  $\pm 30^\circ$  față de forma ideală, segmentele articulare sunt afișate în verde, indicând o execuție corectă. În caz contrar, segmentele sunt colorate în roșu, iar utilizatorului i se afișează în interfață un mesaj de avertizare de forma *"Please correct your right leg angle"*. Această combinație de analiză geometrică și feedback vizual în timp real facilitează corectarea autonomă a execuției exercițiilor fizice, fără intervenție umană suplimentară (vezi Figura 1.5).



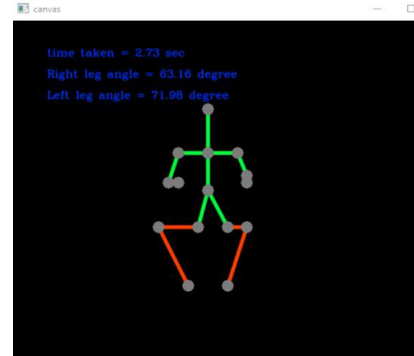
(a) Execuție corectă



(b) Execuție incorectă



(c) Execuție incorectă



(d) Execuție incorectă

Figura 1.5: Exemple de feedback vizual pentru execuția genuflexiunii analizate cu *OpenPose* [Boy21]



Rezultatele indică faptul că sistemul este capabil să detecteze corect execuțiile incorecte, iar abordarea sa modulară permite extinderea către alte tipuri de exerciții. Concluzia autorului subliniază potențialul acestui tip de tehnologie în contextul antrenamentelor personalizate, prevenției accidentărilor și dezvoltării unor platforme de fitness inteligente.

### 1.1.3.3 Detectarea posturilor-cheie cu CNN

Lucrarea lui Pan [Pan22], *A Method of Key Posture Detection and Motion Recognition in Sports Based on Deep Learning*, propune o arhitectură de tip CNN specializată pentru analiza mișcărilor sportive, concentrându-se pe detecția posturilor-cheie și recunoașterea activităților din videoclipuri, structurată în mai multe etape. După preprocesarea secvențelor video, sunt extrase automat cadrele relevante utilizând o metodă denumită RoI\_KP (Region of Interest for Key Postures), care permite identificarea momentelor semnificative dintr-un exercițiu. Figura 1.6 ilustrează diagrama de flux a întregului sistem propus, începând cu segmentarea temporală a videoclipului și terminând cu clasificarea mișcării prin rețea neuronală. Această abordare optimizează performanța rețelei CNN prin reducerea redundanței în datele de intrare.

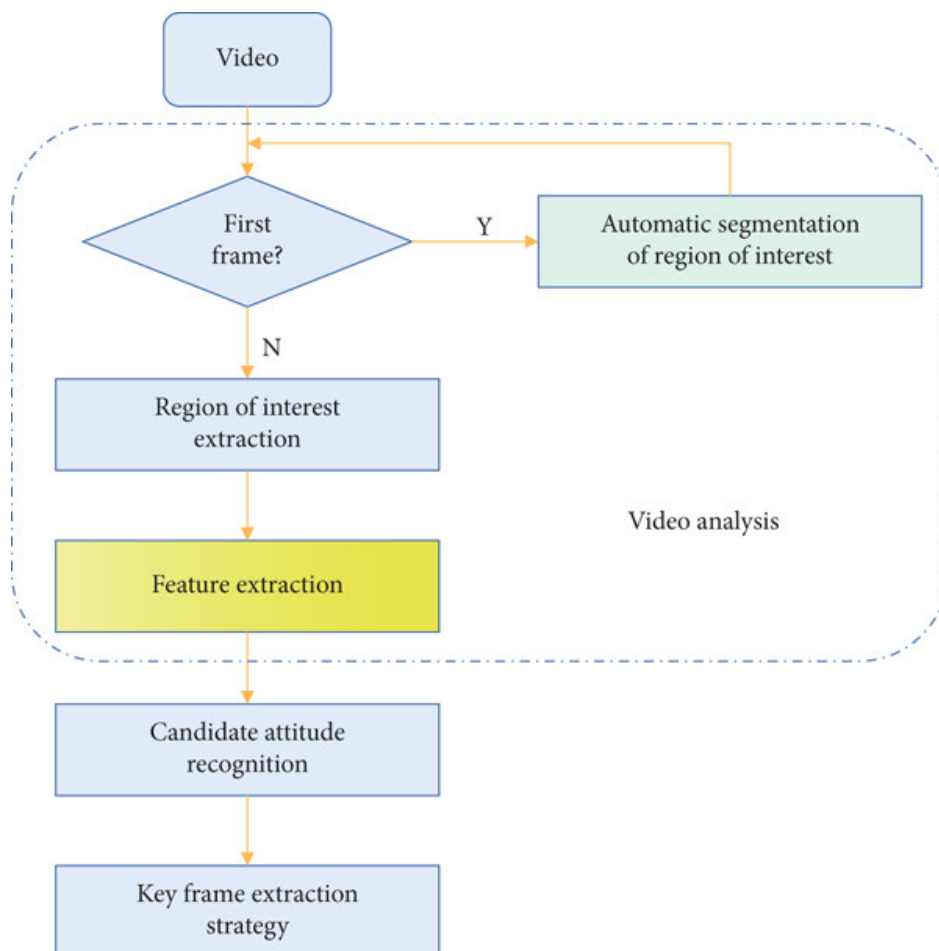


Figura 1.6: Diagrame de flux care descriu pașii sistemului propus [Pan22].

Rețeaua neuronală convoluțională proiectată de autor (Figura 1.7) este adaptată

pentru extragerea caracteristicilor spațiale ale posturii din cadrele selectate de modulul RoI. Aceasta este alcătuită din mai multe straturi convoluționale și de pooling, urmate de un strat complet conectat care realizează clasificarea. Procesarea se face la nivel de cadru individual (*frame-wise*), iar apoi caracteristicile extrase sunt agregate într-o manieră secvențială pentru recunoașterea coerentă a mișcării. Această arhitectură este optimizată pentru recunoașterea precisă a tipului de postură, fiind antrenată pe seturi de date provenite din sporturi de performanță. Prin combinarea acestei arhitecturi CNN cu procesul de selecție a cadrelor relevante, sistemul propus reușește să identifice corect posturile-cheie fără a necesita procesarea tuturor cadrelor dintr-un video.

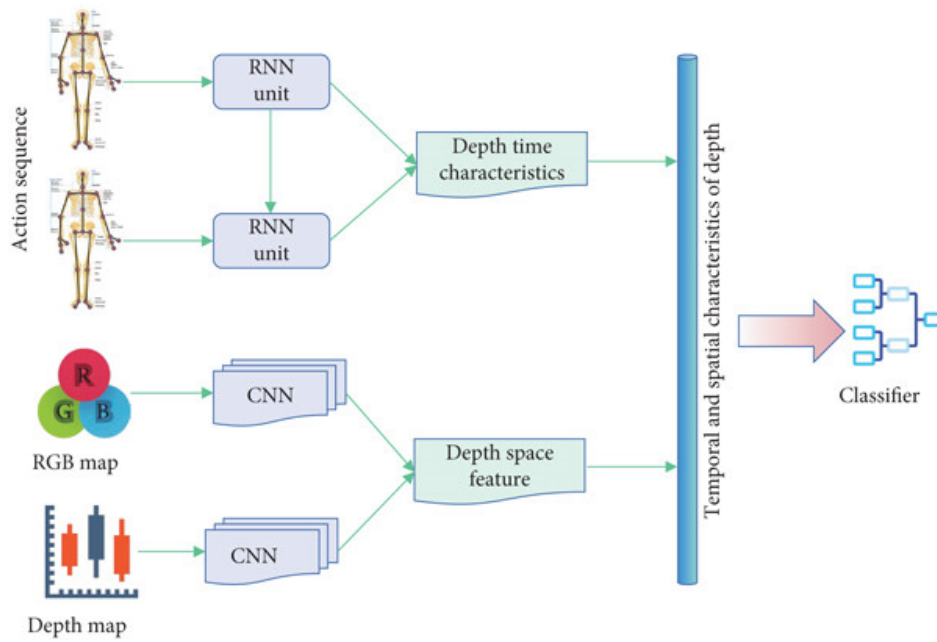


Figura 1.7: Structura rețelei convoluționale propuse de Pan pentru extragerea caracteristicilor posturale din cadrele video selectate [Pan22].

Pentru măsurarea gradului de similaritate între secvențele de postură, autorul utilizează distanța cosinus (Ecuația 1.3), o metrică frecvent folosită în procesarea vectorilor de caracteristici, care indică apropierea dintre execuția curentă și o secvență etalon.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (1.3)$$

Rezultatele experimentale prezentate în Figura 1.8 arată că metoda propusă, bazată pe selecția automată a cadrelor relevante (RoI\_KP) și clasificarea prin CNN, depășește performanțele tehnicilor tradiționale precum HOG+SVM, obținând o acuratețe superioară în recunoașterea mișcărilor sportive.

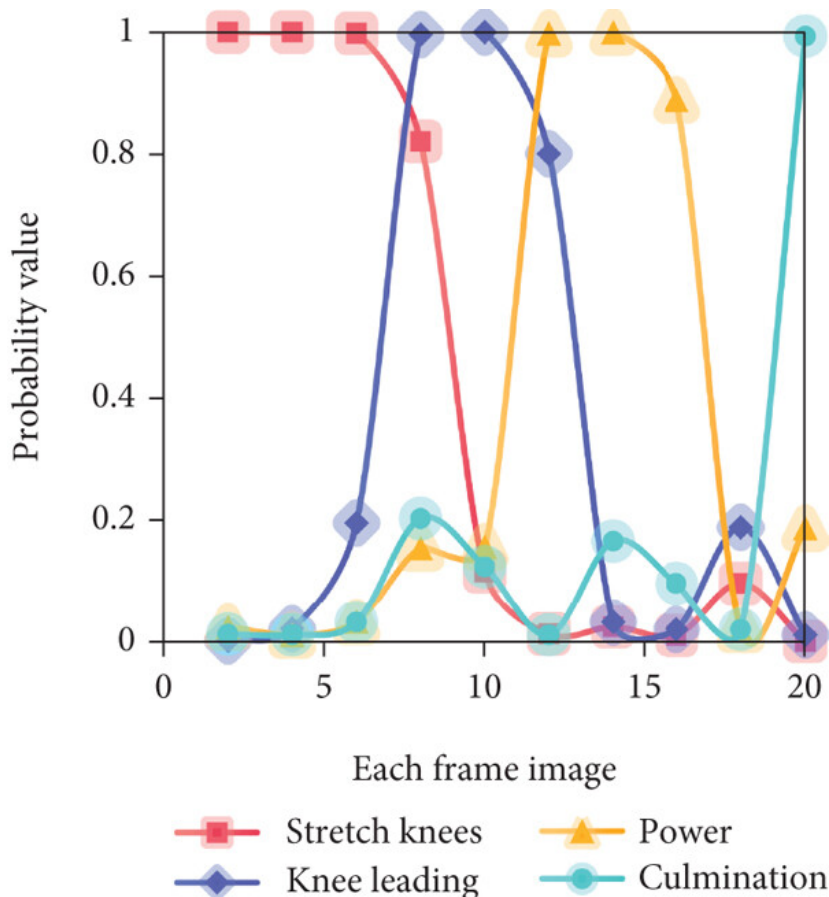


Figura 1.8: Comparație între metoda propusă și alte metode tradiționale privind acuratețea recunoașterii posturii sportive [Pan22].

Lucrarea lui Pan [Pan22] evidențiază aplicabilitatea tehnicilor de învățare profundă în detecția posturilor-cheie și recunoașterea mișcărilor sportive, demonstrând că selecția automată a cadrelor relevante, combinată cu o arhitectură CNN optimizată, permite o analiză eficientă și precisă a exercițiilor fizice. Prin reducerea redundanței din datele video și focalizarea pe momentele esențiale ale mișcării, sistemul oferă o soluție viabilă pentru asistența digitală în procesul de antrenament, putând fi integrat în aplicații interactive pentru feedback vizual sau evaluare educațională, scalabile și aplicabile în contexte reale.

În concluzie, literatura de specialitate evidențiază o tranziție clară de la metodele tradiționale de corectare posturală, bazate pe observație subiectivă și intervenție umană, către soluții automatizate ce integrează viziunea computerizată, rețele neuronale și evaluare geometrică. Studiile analizate demonstrează eficiența unor modele precum *YOLOv7-pose*, *OpenPose* și CNN-uri adaptate, care permit detecția în timp real a deviațiilor posturale și oferirea de feedback vizual imediat. Aceste abordări contribuie semnificativ la dezvoltarea unor sisteme de antrenament inteligente, capabile să sprijine utilizatorii în corectarea autonomă a execuției exercițiilor fizice, atât în scop recreațional, cât și în contexte clinice sau educaționale. Integrarea acestor tehnologii în aplicații practice oferă perspective promițătoare pentru creșterea accesibilității și

eficienței antrenamentelor asistate de inteligență artificială.

## 1.2 Inteligența artificială în domeniul evaluării posturii

Această secțiune își propune să evidențieze fundamentele tehnologice și metodele de inteligență artificială utilizate în dezvoltarea aplicației pentru corectarea posturii în timpul exercițiilor fizice. Vor fi prezentate conceptele teoretice esențiale legate de detectarea poziției corpului uman, precum și abordările moderne de analiză a execuției mișcărilor, cu accent pe utilizarea etichetelor de tip eroare.

În prima parte a secțiunii este discutat rolul inteligenței artificiale în domeniul evaluării posturii, urmată de o analiză a principalelor metode de interpretare a poziției corpului în mișcare, incluzând modele de detecție și sisteme de evaluare bazate pe erori. Ultima secțiune este dedicată **rețelelor neuronale convoluționale** (CNN), prezentând importanța acestora în procesarea atât a imaginilor cât și a videoclipurilor și aplicabilitatea lor în clasificarea posturilor corecte și incorecte. De asemenea, este abordată și utilizarea **modelelor de tip XGBoost** aplicate pe trăsături extrase din *landmark-urile corpului*, oferind o alternativă eficientă la rețelele neuronale în contexte în care reprezentarea numerică a posturii permite clasificarea directă prin modele tradiționale de învățare automată.

**Inteligența artificială** (IA), definită ca ansamblul tehnicilor care permit sistemelor informatice să simuleze inteligența umană, este astăzi utilizată cu succes în multiple domenii, de la procesarea limbajului natural până la recunoașterea imaginilor și automatizarea deciziilor. În contextul activităților fizice și al sănătății posturale, IA oferă instrumente inovatoare pentru monitorizarea, evaluarea și corectarea poziției corpului uman. Această descriere se aplică și în cazul sistemelor care asistă utilizatorii în corectarea posturii prin analiză video sau imagine.

În ultimii ani, progresele în învățarea automată (machine learning) și în special în învățarea profundă (deep learning), prin utilizarea rețelelor neuronale convoluționale, au condus la apariția unor soluții precise pentru detecția și evaluarea posturii corporale. Aceste tehnologii permit extragerea automată a caracteristicilor relevante din date vizuale, identificând tipare recurente asociate cu execuții corecte sau incorecte ale exercițiilor fizice [LBH15]. În comparație cu metodele tradiționale, bazate pe expertiza antrenorului sau terapeutului, sistemele inteligente oferă o evaluare obiectivă, consistentă și replicabilă. Acestea pot oferi feedback în timp real, detectând deviațiile de la postura ideală și semnalând utilizatorului necesitatea ajustării mișcărilor. Astfel, inteligența artificială contribuie semnificativ la prevenirea accidentărilor, optimizarea performanței fizice și promovarea unei practici corecte a exercițiilor [ZWXX21]. De exemplu, sistemele de monitorizare a exercițiilor fizice pot fi folosite în cadrul unui mediu privat, permițând utilizatorilor să efectueze antrenamente fără supravegherea directă a unui specialist, beneficiind totodată de feedback automatizat. De asemenea, aceste soluții pot sprijini activitatea antrenorilor, kinetoterapeuților sau profesorilor de

educație fizică, oferind evaluări obiective ale execuției exercițiilor. În plus, integrarea inteligenței artificiale în aplicații mobile și platforme de tip e-learning sportiv contribuie la crearea unor experiențe interactive, personalizate în funcție de performanțele și nevoile utilizatorului, facilitând astfel accesul larg la programe de antrenament asistate tehnologic.

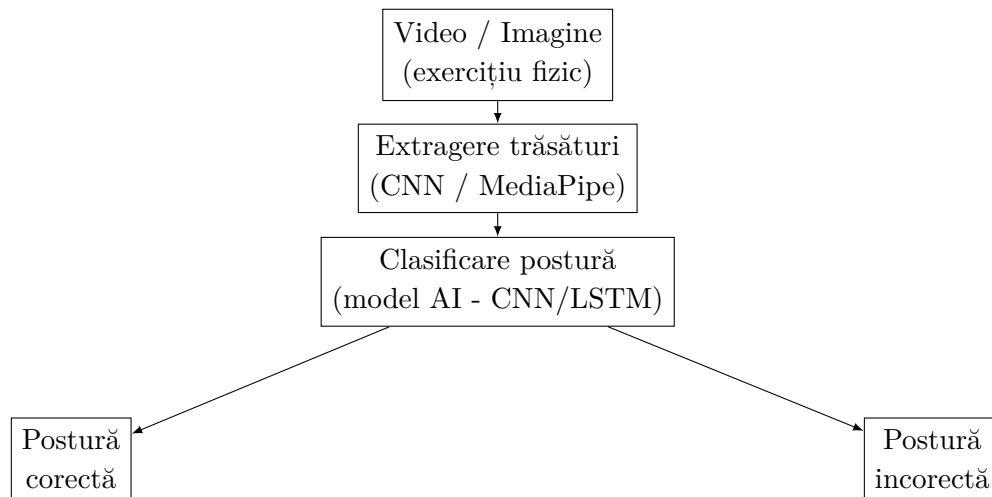


Figura 1.9: Flux de prelucrare IA pentru evaluarea posturii corporale

Prin urmare, integrarea inteligenței artificiale în evaluarea și corectarea posturii reprezintă un pas important spre digitalizarea antrenamentelor fizice și accesibilizarea acestora pentru un public larg, oferind un cadru tehnologic solid pentru susținerea sănătății și performanței fizice.

### 1.2.1 Tehnici de interpretare a poziției corpului în mișcare

Evaluarea posturii corporale implică un set complex de tehnici menite să extragă și să interpreteze informații din imagini sau videoclipuri cu utilizatori care execută mișcări specifice. Această secțiune analizează fundamentele teoretice ale detectării posturii și ale sistemelor automatizate de evaluare a execuției exercițiilor, cu accent pe modalitățile prin care algoritmi de inteligență artificială pot identifica și clasifica tipare de mișcare. În mod special, sunt abordate metodele de extragere a trăsăturilor relevante din punct de vedere biomecanic și logica utilizării erorilor semnificative în cadrul unui sistem de evaluare.

#### 1.2.1.1 Definiția și principiile detectării posturii

Detectarea posturii se referă la procesul de estimare automată a poziției și orientării segmentelor corporale în spațiu, pornind de la imagini statice sau cadre video. Aceasta presupune identificarea unui set de puncte de interes (keypoints), corespunzător articulațiilor majore, care sunt apoi folosiți pentru a reconstrui o reprezentare scheletică a corpului. Un exemplu notabil este **MediaPipe Pose**, o arhitectură dezvoltată de Google, capabilă să identifice 33 de puncte anatomice într-un cadru video [BLG<sup>+</sup>20].

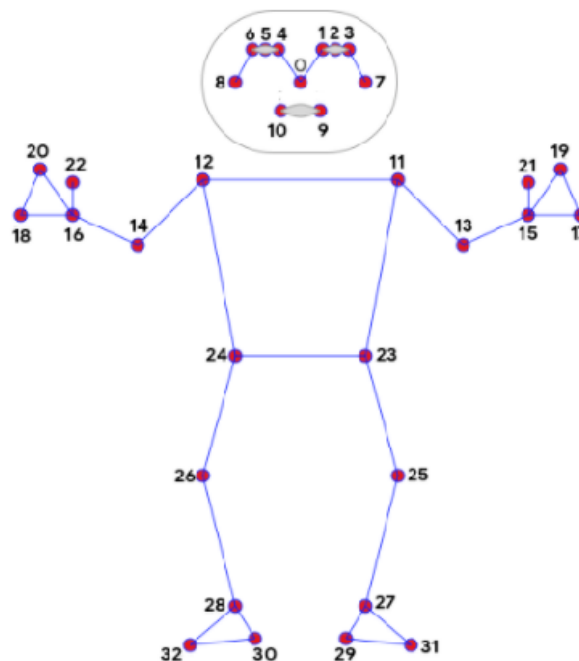


Figura 1.10: Cele 33 de puncte anatomice (*landmark-uri*) MediaPipe Pose [SARG23]

Informațiile obținute din aceste puncte cheie pot fi ulterior procesate pentru extragerea de trăsături numerice precum unghiuri între segmente osoase, distanțe relative sau orientări spațiale. Aceste trăsături pot servi drept intrare pentru diverse modele de clasificare sau evaluare, fie de tip rețea neuronală, fie bazate pe învățare automată tradițională.

MediaPipe oferă un pipeline eficient și ușor de integrat, fiind optimizat pentru rulare în timp real pe CPU, ceea ce îl face potrivit pentru aplicații educaționale sau de antrenament în timp real. Detectarea eficientă a posturii presupune:

- extragerea automată a trăsăturilor relevante din imagine sau video;
- antrenarea pe seturi de date etichetate care conțin poziții scheletice corecte;
- compararea ieșirilor modelului cu valori de referință (ground truth).

Aceste principii permit analiza biomecanică a mișcării și constituie baza oricărui sistem care oferă feedback corectiv sau de antrenament.

### 1.2.1.2 Sisteme de evaluare a execuției exercițiilor pe baza erorilor

Sistemele de evaluare a execuției exercițiilor se bazează frecvent pe paradigmele învățării supervizate, în care modelele sunt antrenate pe baza unui set de date etichetat, adică fiecare instanță de antrenament este însoțită de o etichetă ce descrie execuția corectă sau incorectă. Învățarea supervizată (Supervised Learning) este un subdomeniu fundamental al învățării automate în care algoritmi învață să prezică o etichetă de ieșire pe baza unor exemple de intrare, având ca scop generalizarea corectă la date noi nevăzute. Această tehnică presupune existența unei funcții de mapare care corelează un set de variabile de intrare (cum ar fi pozițiile articulațiilor, unghiuri ale corpului,

sau traiectorii ale segmentelor) cu o variabilă de ieșire cunoscută, adesea de natură binară sau ordinală. Performanța unui model supervizat este evaluată prin metrice precum acuratețea, precizia sau scorul F1, în funcție de complexitatea problemei și de distribuția dezechilibrată a claselor.

Spre deosebire de învățarea nesupervizată, care grupează date pe baza similarității fără a avea etichete, sau învățarea prin întărire (reinforcement learning), care se bazează pe interacțiune și recompense, învățarea supervizată permite antrenarea eficientă a clasificatorilor pentru sarcini bine definite, cum ar fi clasificarea posturii. Printre avantajele învățării supervizate se numără simplitatea conceptuală, eficiența în procesul de antrenare și capacitatea de interpretare a rezultatelor prin intermediul matricelor de confuzie sau al analizelor de eroare.

În mod particular, în multe seturi de date recente utilizate în cercetare sau aplicații reale, adnotările realizate de experți sunt salvate în fișiere de tip JSON, organizate pe categorii de erori, cum ar fi erori de aliniere a trunchiului, poziționarea incorectă a genunchilor sau lipsa menținerii curburii coloanei lombare. Aceste fișiere acționează ca sursă de ground truth, indicând pentru fiecare instanță de date (ex. o imagine sau o secvență video) dacă execuția corespunde unei mișcări corecte sau prezintă una sau mai multe abateri biomecanice. Prin parcurgerea acestor fișiere și asocierea etichetelor corespunzătoare instanțelor vizuale, se poate construi o bază de date bine structurată pentru antrenarea clasificatorilor. De exemplu, un sistem automat poate marca drept „incorectă” orice secvență în care este detectată cel puțin o eroare, iar drept „corectă” orice secvență lipsită de erori. Această abordare binară este frecvent utilizată în fazele inițiale de prototipare, întrucât simplifică problema de clasificare și permite antrenarea rapidă a modelelor. În etape ulterioare, dacă este disponibilă o etichetare mai granulară (ex. tipul și severitatea erorii), se pot implementa clasificatoare multi-clasă sau modele de regresie, care oferă scoruri continue de performanță. Aceste sisteme sunt mai flexibile și pot furniza feedback adaptat nivelului utilizatorului, fiind utile atât în aplicații de antrenament sportiv, cât și în programe de reabilitare medicală.

### 1.2.2 Rolul rețelelor neuronale în evaluarea execuției exercițiilor

Rețelele neuronale artificiale (ANN) sunt modele computaționale inspirate de rețeaua de neuroni biologici din creierul uman. Acestea sunt alcătuite din straturi de neuroni artificiali organizați în mod ierarhic: un strat de intrare, unul sau mai multe straturi ascunse și un strat de ieșire. Fiecare neuron artificial primește ca intrare un set de valori  $x_i$ , le înmulțește cu ponderi  $w_i$ , adună rezultatul cu un termen de bias  $b$  și aplică o funcție de activare  $f$ , generând astfel o ieșire [Hay98]:

$$y = f \left( \sum_{i=1}^n w_i x_i + b \right) \quad (1.4)$$

Funcțiile de activare frecvent utilizate includ ReLU (Rectified Linear Unit), funcțiile sigmoid sau tanh, utile în contexte secvențiale sau de clasificare binară. Funcția ReLU



este una dintre cele mai utilizate funcții în rețelele neuronale profunde, datorită simplității sale computaționale și eficienței în antrenarea modelelor complexe. Matematic, aceasta este definită ca  $f(x) = \max(0, x)$ , unde  $x$  reprezintă inputul către neuron. Această funcție introduce non-liniaritate în rețea, permițând modelului să învețe reprezentări complexe ale datelor. Spre deosebire de funcțiile sigmoide sau tangente hiperbolice, care suferă de fenomenul de vanishing gradient, ReLU menține un gradient constant pentru valorile pozitive, facilitând astfel propagarea eficientă a erorii în rețele profunde (deep networks). Cu toate acestea, o limitare a acestei funcții este că neuronii pot „muri” în timpul antrenării dacă intrarea este mereu negativă, ceea ce duce la un gradient zero. Pentru a adresa această problemă, au fost propuse variante precum Leaky ReLU sau Parametric ReLU (PReLU) [NH10].

Totuși, rețelele complet conectate nu sunt eficiente în procesarea imaginilor, din cauza numărului foarte mare de parametri necesari pentru fiecare conexiune și a pierderii structurii spațiale a datelor vizuale. Pentru a remedia aceste probleme, au fost introduse rețelele neuronale convoluționale, care exploatează proprietățile locale ale imaginilor prin restrângerea conexiunilor la regiuni locale și partajarea ponderilor între neuroni [LBBH98].

**Convoluția** Operația de bază a unei CNN este *convoluția*, care constă în aplicarea unui filtru (kernel) peste o regiune locală a imaginii de intrare. Kernelul este o matrice mică, de obicei 3x3 sau 5x5, ale cărei valori sunt învățate în timpul antrenării. La fiecare pas, filtrul este aplicat pe o fereastră a imaginii, iar produsul scalar dintre valorile pixelilor și valorile kernelului este stocat într-o hartă de trăsături (feature map):

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \quad (1.5)$$

unde  $I$  reprezintă imaginea de intrare (matricea de pixeli),  $K$  kernelul sau filtrul aplicat, iar  $S$  harta de trăsături rezultată. Această operație permite extragerea automată de trăsături precum muchii, colțuri sau modele complexe, păstrând totodată informația spațială din imagine.

**Pooling și dimensionalitate.** După fiecare strat de convoluție este de obicei inserat un strat de *pooling*, utilizat pentru a reduce dimensiunea spațială a reprezentării și a preveni supraînvățarea. Cele mai comune tehnici sunt *max pooling* (păstrează valoarea maximă dintr-o regiune) și *average pooling* (păstrează media valorilor).

**Straturi complet conectate (fully connected layers).** După trecerea prin straturile convoluționale și de pooling, rețelele CNN includ de obicei unul sau mai multe straturi complet conectate. Aceste straturi sunt similare cu cele din *rețelele neuronale artificiale* clasice (ANN), în care fiecare neuron dintr-un strat este conectat la toți neuronii din stratul anterior. Scopul lor este de a combina trăsăturile extrase și de a realiza clasificarea finală. Straturile complet conectate sunt responsabile pentru decizia finală a rețelei, având ca ieșire scoruri de probabilitate pentru fiecare clasă posibilă.

Rețelele CNN utilizează astfel o arhitectură compusă din:



- straturi convoluționale pentru extragerea trăsăturilor;
- straturi de pooling pentru comprimarea datelor;
- straturi complet conectate pentru clasificare.

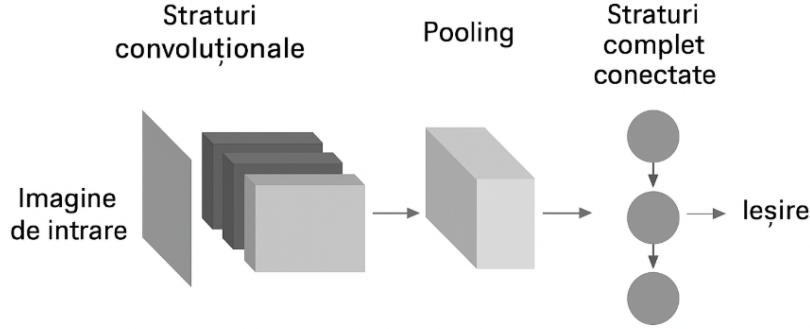


Figura 1.11: Arhitectura unei rețele CNN cu straturi convoluționale, pooling și complet conectate

Pentru procesarea videoclipurilor sau a datelor temporale, CNN-urile pot fi extinse cu rețele recurente de tip *Long Short-Term Memory* (LSTM), capabile să păstreze o “memorie” a informației din cadrul anterior. Această combinație (CNN+LSTM) este frecvent utilizată în aplicații care implică analiza mișcărilor sau acțiunilor în timp. Rețelele de tip LSTM reprezintă o extensie a **rețelelor neuronale recurente clasice** (RNN), concepute pentru a gestiona secvențe de date și a reține informații pe termen lung. Acestea includ trei componente principale: poarta de uitare (forget gate) care decide ce informație este uitată, poarta de intrare (input gate) care decide ce este adăugat în memorie și poarta de ieșire (output gate) care decide ce informație este transmisă mai departe. Aceste componente controlează fluxul de informație în și din starea internă a celulei (cell state). Acest mecanism permite LSTM-ului să decidă ce informație este relevantă și trebuie păstrată și ce informație poate fi uitată, evitând astfel problema dispariției gradientului întâlnită la RNN-urile tradiționale. Matematic, funcționarea unei celule LSTM poate fi descrisă prin următoarele formule:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1.6)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (1.7)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (1.8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (1.9)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (1.10)$$

$$h_t = o_t \odot \tanh(c_t) \quad (1.11)$$

unde  $x_t$  este intrarea la momentul  $t$ ,  $h_{t-1}$  este starea ascunsă precedentă,  $c_t$  este starea celulei, iar  $\sigma$  reprezintă funcția sigmoid [HS97]. Aceste componente lucrează împreună pentru a permite memorarea și actualizarea eficientă a informației într-o secvență.

Rețelele LSTM sunt deosebit de utile în analiza secvențială, precum în procesarea video, deoarece pot învăța contextul temporal al mișcărilor, oferind o interpretare mai coerentă și precisă a execuției exercițiilor fizice.

Structura tipică include:

- CNN pentru extragerea trăsăturilor vizuale din fiecare cadru;
- un modul LSTM care primește secvența de trăsături pentru a înțelege dinamica temporală;
- un strat final de clasificare.

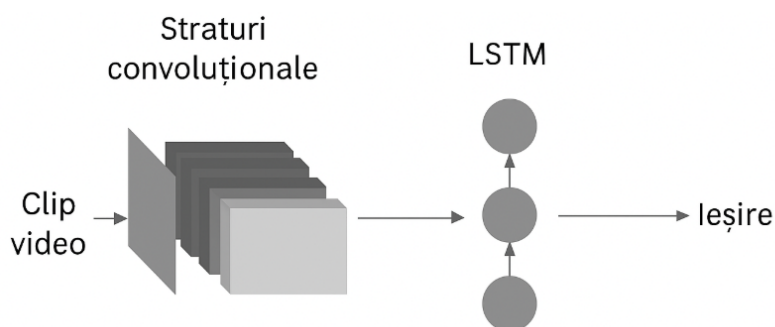


Figura 1.12: Arhitectura CNN-LSTM pentru clasificarea secvențială a posturilor în videoclipuri

Combinarea CNN cu rețelele neuronale recurente de tip LSTM aduce beneficii semnificative în analiza execuției exercițiilor fizice, întrucât permite atât extragerea trăsăturilor vizuale esențiale din fiecare cadru (prin CNN), cât și modelarea relațiilor temporale dintre cadre (prin LSTM). Acest tip de arhitectură hibridă facilitează interpretarea contextului biomecanic al unei mișcări și oferă posibilitatea realizării de predicții în timp real asupra secvențelor video. Prin urmare, modelele CNN-LSTM sunt aplicate cu succes în domenii precum monitorizarea performanței sportive, reabilitarea medicală asistată, realitatea augmentată sau sistemele inteligente de antrenament, contribuind la îmbunătățirea posturii și prevenirea accidentărilor prin evaluări automatizate și precise [DAHG<sup>+</sup>15].

**XGBoost (Extreme Gradient Boosting).** Pe lângă rețelele neuronale convoluționale, o alternativă eficientă și adesea complementară o reprezintă metodele tradiționale de învățare automată, cum este algoritmul **XGBoost** [CG16]. Acesta este un model de tip *ensemble*, bazat pe arbori de decizie antrenați secvențial, în cadrul unui proces de optimizare a funcției de pierdere prin tehnici de gradient boosting. Fiecare arbore nou construit are scopul de a corecta erorile făcute de arborii anteriori, contribuind astfel la un model final puternic și stabil.

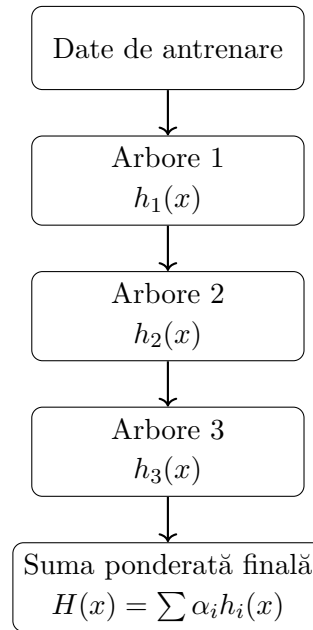


Figura 1.13: Funcționarea algoritmului XGBoost

În contextul analizei posturii, XGBoost poate fi utilizat în mod eficient atunci când datele de intrare constau în trăsături geometrice numerice, extrase din imagini sau videoclipuri. Aceste trăsături nu sunt obținute direct din pixelii imaginii, ci printr-o etapă intermediară de detecție a punctelor anatomice (landmark-uri), realizată cu ajutorul bibliotecii *MediaPipe Pose*.

Combinarea dintre MediaPipe și XGBoost creează un pipeline eficient: MediaPipe extrage informația posturală brută din imagine, iar XGBoost o interpretează prin clasificare. Această abordare are avantajul separării clar definite între etapa de detecție (bazată pe viziune computerizată) și etapa de decizie (bazată pe modele de clasificare explicabile), permițând astfel o mai bună optimizare și o interpretabilitate crescută a rezultatelor. De asemenea, dimensiunea redusă a setului de trăsături face ca antrenarea modelului să fie rapidă și eficientă chiar și pe sisteme fără accelerare hardware.

# Capitolul 2

## Abordarea propusă pentru evaluarea posturii

În acest capitol este prezentată metodologia practică de construire a unui sistem automatizat pentru identificarea execuției incorecte a exercițiilor fizice, cu accent pe dezvoltarea modelului de învățare automată utilizat în procesul de clasificare posturală. Sunt detaliate etapele esențiale care au stat la baza construirii modelului, începând cu procesul de colectare și preprocesare a datelor, urmat de selecția și descrierea arhitecturii rețelei neuronale utilizate, până la configurarea și antrenarea efectivă a acesteia.

De asemenea, sunt prezentate strategiile de evaluare aplicate pentru a cuantifica performanța modelului, utilizând metrice relevante din domeniul clasificării binare. Această abordare practică asigură o bază solidă pentru integrarea ulterioară a modelului într-un sistem complet de monitorizare a posturii.

### 2.1 Metodologie

#### 2.1.1 Colectarea și preprocesarea datelor de antrenament

În vederea antrenării unui model de învățare automată pentru clasificarea execuției exercițiilor fizice, a fost utilizat un subset de date din cadrul setului public **Fitness Action Quality Assessment (Fitness-AQA)** [PGR22]. Acest set de date este conceput pentru a evalua calitatea execuției exercițiilor de fitness și conține atât videoclipuri, cât și imagini statice, adnotate de experți în domeniu.

În cadrul lucrării, atenția a fost concentrată pe trei exerciții fundamentale de fitness: *ramat cu bara aplecat (barbell row)*, *genuflexiuni (squat)* și *presa deasupra capului (overhead press)*. Fiecare exercițiu este însoțit de date vizuale și adnotări privind erorile biomecanice frecvente întâlnite în execuția acestora. Acestea au fost selectate datorită relevanței lor în evaluarea posturii, gradului ridicat de complexitate biomecanică implicată și frecvenței erorilor de execuție întâlnite în practică. Pentru ramat cu bara, setul de date oferă imagini statice etichetate cu fișiere de tip JSON care indică prezența unor erori precum „eroare lombară” (*labels\_lumbar\_error.json*), care vizează curbura incorectă a zonei lombare a spatelui în timpul mișcării, și „unghi incorect al trunchiului” (*labels\_torso\_angle\_error.json*), care reflectă o înclinare neadecvată a

trunchiului față de sol în timpul execuției exercițiului. Pentru exercițiile genuflexiune și presa deasupra capului, datele sunt furnizate sub formă de videoclipuri, fiecare fișier fiind însoțit de un set de adnotări structurate în fișiere *JSON* care semnalează tipuri de greșeli identificate de experți. În cazul genuflexiunilor, erorile urmărite includ proiecția excesivă a genunchilor în față (*error\_knees\_forward.json*) și colapsul genunchilor spre interior (*error\_knees\_inward.json*), ambele fiind indicatori biomecanici ai unei execuții deficitare. Pentru exercițiul presa deasupra capului, etichetările vizează poziționarea incorectă a coatelor (*error\_elbows.json*) și instabilitatea în alinierea genunchilor (*error\_knees.json*), factori care afectează controlul mișcării și siguranța posturală. Aceste adnotări indică prezența sau absența erorilor de execuție. În cadrul acestei lucrări, etichetarea a fost realizată în mod binar: o secvență a fost considerată „incorectă” dacă a fost identificată cel puțin o eroare, iar „corectă” în absența acestora. Astfel, pentru fiecare exercițiu, ramat cu bara, genuflexiuni și presa deasupra capului, au fost definite două clase distincte de date: „corectă” și „incorectă”, această clasificare binară fiind esențială pentru antrenarea eficientă a modelelor de învățare automată.

## 2.1.2 Descrierea arhitecturii modelelor și a procesului de antrenare pentru clasificare

Pentru clasificarea posturii în execuția exercițiilor fizice, au fost dezvoltate trei modele distincte, adaptate formatului și naturii datelor disponibile. Prima arhitectură este destinată antrenării pe imagini statice și este implementată sub forma unui model CNN simplificat. A doua arhitectură combină CNN cu rețele de tip LSTM, pentru a gestiona secvențele temporale din videoclipuri. În completare, a fost implementată o abordare alternativă bazată pe algoritmul XGBoost, care clasifică postura pe baza trăsăturilor geometrice extrase din landmark-urile detectate cu ajutorul MediaPipe.

Toate cele trei arhitecturi au fost dezvoltate astfel încât să fie ușor de extins, testat și comparat, având la bază o clasă abstractă comună pentru modelele neurale, respectiv un pipeline separat pentru clasificatorul tradițional. Această diversitate metodologică permite evaluarea avantajelor și limitărilor fiecărei abordări în funcție de tipul de date de intrare și contextul de utilizare.

### 2.1.2.1 Clasificarea pe imagini statice cu CNN

**Arhitectura modelului.** Pentru clasificarea execuției exercițiului *ramat cu bara* pe baza imaginilor statice, a fost proiectat și implementat un model de rețea neuronală convoluțională (CNN), adaptat pentru detectarea trăsăturilor vizuale relevante în postura corporală. Alegerea arhitecturii CNN se justifică prin eficiența acestui tip de rețea în procesarea imaginilor, fiind capabil să extragă automat trăsături spațiale relevante fără a necesita o etapă manuală de selecție sau preprocesare a caracteristicilor.

Modelul CNN implementat este derivat dintr-o clasă abstractă *AbstractClassifier*, care definește metodele esențiale pentru încărcarea și filtrarea imaginilor, preprocesarea acestora și împărțirea datelor în seturi de antrenare și testare. Această arhitectură specifică este implementată în clasa *CNNPostureClassifier* și este alcătuită din următoarele

componente:

- **trei blocuri convoluționale**, fiecare alcătuit dintr-un strat Conv2D cu kernel  $3 \times 3$  și activare ReLU, urmat de BatchNormalization și un strat MaxPooling2D - aceste blocuri extrag progresiv trăsături vizuale relevante și reduc dimensiunea spațială, contribuind totodată la prevenirea overfitting-ului.
- **un strat de aplatizare (flattening)** - care transformă datele bidimensionale în vectori unidimensionali pentru a putea fi transmise către partea complet conectată a rețelei;
- **două straturi complet conectate (fully connected)** - responsabile pentru învățarea combinațiilor complexe de trăsături și pentru decizia finală de clasificare;
- **strat de dropout** - care are un rol important în prevenirea fenomenului de overfitting. Această tehnică constă în dezactivarea aleatorie a unui procent (în acest caz, 50%) dintre neuronii unui strat complet conectat în timpul fiecărei iterații de antrenare. Astfel, modelul este forțat să învețe reprezentări robuste și să nu se bazeze excesiv pe anumite conexiuni specifice;
- **funcție de activare sigmoid** în stratul de ieșire, utilizată pentru clasificare binară între postura corectă și incorectă.

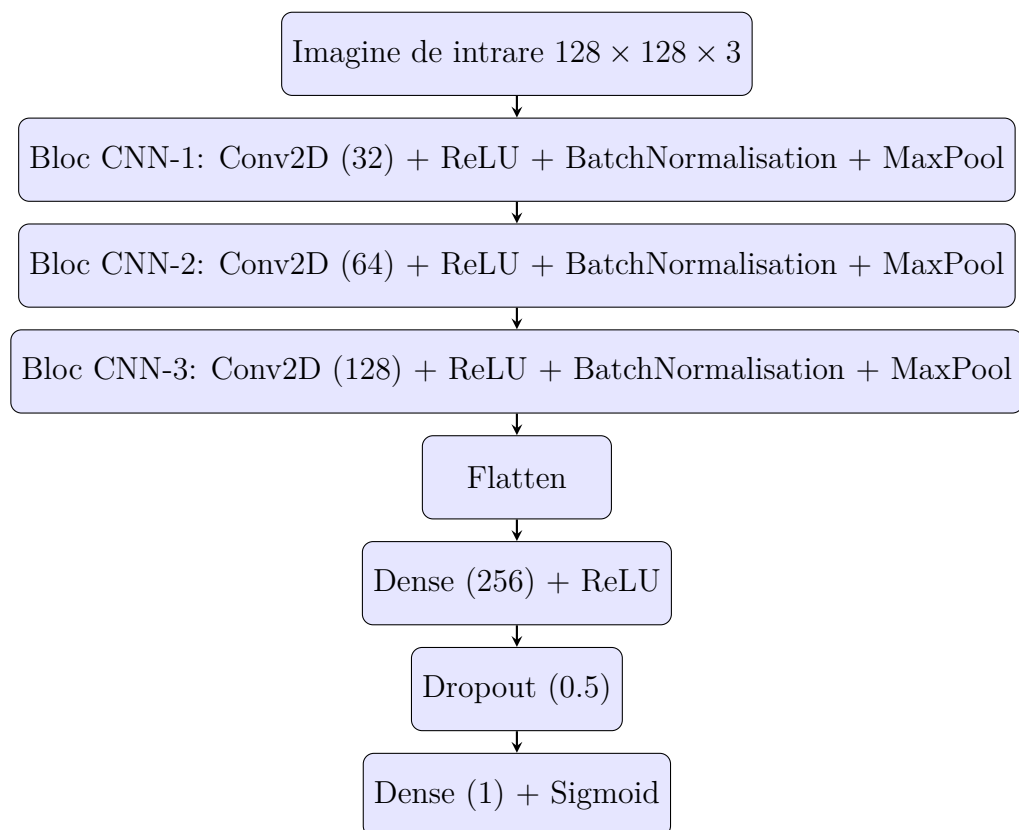


Figura 2.1: Arhitectura CNN pentru clasificarea imaginilor statice

**Pregătirea datelor și procesul de antrenare.** Procesul de antrenare al unui model de învățare profundă presupune ajustarea parametrilor rețelei neuronale astfel încât eroarea dintre predicțiile modelului și etichetele reale să fie minimizată. Acest proces este iterativ și implică parcurgerea unui set de date de antrenament în mai multe epoci, fiecare epocă reprezentând o trecere completă prin întregul set de date. Scopul este eficientizarea unei funcții de pierdere (loss function) utilizând un algoritm de optimizare, precum Adam (Adaptive Moment Estimation), care este adaptiv și stabil din punct de vedere numeric pentru rețele neuronale profunde.

Antrenarea modelului CNN s-a realizat pe un set de date etichetat manual, alcătuit din imagini statice corespunzătoare execuției exercițiului *ramat cu bara*. Fiecare imagine are dimensiunea standardizată  $128 \times 128$  pixeli, iar setul de date a fost împărțit în proporție de 80% pentru antrenare și 20% pentru testare, cu o distribuție echilibrată între clasele „corect” și „incorect”.

Pentru a evita dezechilibrul inițial între cele două clase, s-au aplicat tehnici de augmentare doar asupra clasei „incorect”, astfel încât numărul total de exemple din fiecare clasă să ajungă la 10.000. Scriptul de augmentare a inclus operații aleatoare precum:

- rotații ale imaginilor cu până la  $15^\circ$ ;
- flip-uri orizontale aleatorii (cu o probabilitate de 50%);
- ajustări de luminozitate și contrast (*ColorJitter*);
- decupaje redimensionate aleatorii (*RandomResizedCrop*) cu scale între  $[0.8, 1.0]$ .

Astfel, datele de antrenare au fost extinse artificial pentru a îmbunătăți robustețea modelului față de variații naturale precum unghiuri diferite de captură, lumini neuniforme sau poziționări parțiale ale subiectului în cadru.

Modelul este compilat folosind optimizatorul *Adam*, iar funcția de pierdere aleasă este *binary\_crossentropy*, potrivită pentru sarcini de clasificare binară. Antrenarea s-a desfășurat pe parcursul a maximum 30 de epoci, utilizând un batch size de 32, iar performanța a fost monitorizată printr-o strategie de oprire timpurie (*early stopping*) aplicată pe pierderea de validare. Dacă nu s-au înregistrat îmbunătățiri timp de 5 epoci consecutive, antrenarea a fost oprită automat și au fost restaurate cele mai bune greutăți.

Algoritmul de optimizare utilizat pentru ajustarea parametrilor rețelei a fost *Adam*, o metodă care combină avantajele algoritmului *AdaGrad* și ale metodei *RMSProp*. Adam adaptează rata de învățare pentru fiecare parametru în parte, folosind estimări ale primului moment (media gradientului) și al doilea moment (media pătratelor gradientului). Astfel, Adam accelerează convergența și îmbunătățește stabilitatea antrenării în rețele neuronale profunde. La fiecare pas  $t$ , parametrii  $w$  ai modelului sunt

actualizați conform relațiilor:

$$\begin{aligned}
 m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \\
 v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \\
 \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\
 \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\
 w_{t+1} &= w_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}
 \end{aligned} \tag{2.1}$$

unde  $g_t$  este gradientul la pasul  $t$ ,  $\eta$  este rata de învățare, iar  $\beta_1$ ,  $\beta_2$  și  $\varepsilon$  sunt hiperparametri controlați manual. Valori tipice sunt  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  și  $\varepsilon = 10^{-8}$ .

La finalul procesului de antrenare, modelul salvează greutatea optime într-un fișier *.h5*, putând fi utilizat ulterior în etapa de inferență pe imagini noi. Formatul *HDF5* (*.h5*) este standard în TensorFlow/Keras pentru salvarea atât a arhitecturii rețelei, cât și a parametrilor învățați, facilitând încărcarea rapidă și reutilizarea modelului fără reantrenare.

### 2.1.2.2 Clasificarea pe secvențe video cu CNN-LSTM

**Arhitectura modelului.** Pentru evaluarea execuției exercițiilor fizice în format video, a fost dezvoltată o arhitectură hibridă de tip *CNN-LSTM*, implementată în clasa *CNNLSTMClassifier*, care extinde funcționalitățile de bază ale clasei abstracte *AbstractClassifierVideo*. Această arhitectură este concepută pentru a combina avantajele rețelelor convoluționale în extragerea trăsăturilor spațiale cu cele ale rețelelor neuronale recurente (LSTM) în modelarea dependențelor temporale dintre cadrele unui videoclip.

Modelul prelucrează secvențele video compuse din 20 cadre selectate uniform de-a lungul fiecărui fișier video. Fiecare cadru este redimensionat la  $64 \times 64$  pixeli și normalizat, formând astfel un tensor de intrare cu forma  $[batch\_size, 20, 64, 64, 3]$ . Prelucrarea fiecărui cadru individual este realizată printr-o rețea convoluțională cu trei blocuri CNN asemănătoare cu cele utilizate în arhitectura pentru imagini statice. Aceste blocuri constau din straturi *Conv2D* urmate de *ReLU*, *BatchNormalization* și *MaxPooling2D*, fiind aplicate asupra fiecărui cadru cu ajutorul stratului *TimeDistributed*, care menține alinierea temporală a trăsăturilor extrase.

După extragerea trăsăturilor spațiale, rezultatul este o secvență de vectori de trăsături pe care modelul o transmite către un strat *LSTM* cu 64 de neuroni. Acesta are rolul de a învăța pattern-uri temporale din dinamica execuției, cum ar fi ritmul mișcării, succesiunea articulațiilor implicate sau stabilitatea poziției. Stratului *LSTM* îi urmează un strat complet conectat cu 64 de neuroni și activare *ReLU*, urmat de un strat de *Dropout* cu rată 0.5 pentru regularizare.

Clasificarea finală este realizată printr-un strat *Dense* cu activare *sigmoid*, care returnează o probabilitate asociată clasei „execuție corectă”. Întregul flux arhitectural este prezentat schematic în Figura 2.2, unde sunt ilustrate atât etapele spațiale aplicate individual fiecărui cadru, cât și agregarea secvențială a acestora pentru luarea deciziei finale.



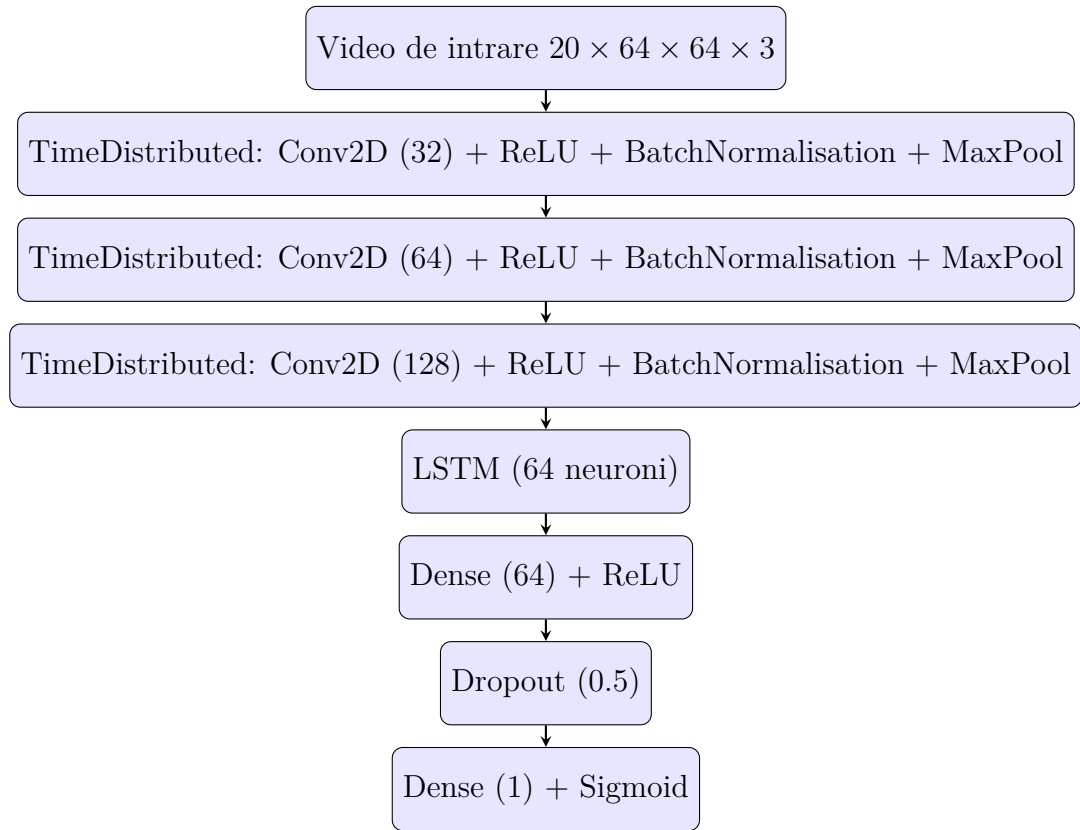


Figura 2.2: Arhitectura CNN-LSTM pentru clasificarea în secvențe video

**Pregătirea datelor și procesul de antrenare.** Preprocesarea datelor video este esențială pentru performanța unui model de tip CNN-LSTM, întrucât acesta trebuie să interpreteze corect atât trăsăturile vizuale ale fiecărui cadru, cât și succesiunea lor în timp. În cadrul acestei aplicații, preprocesarea este gestionată prin clasa abstractă *AbstractClassifierVideo*, care oferă metode pentru încărcarea fișierelor video, extragerea cadrelor relevante, redimensionarea acestora și normalizarea valorilor de pixeli. Fiecare videoclip este convertit într-o secvență de 20 cadre alese la intervale egale pe durata clipului. Acest tip de sampling uniform permite păstrarea structurii temporale a mișcării, evitând acumularea de cadre redundante sau pierderea unor faze esențiale ale execuției. Fiecare cadru extras este redimensionat la rezoluția standard  $64 \times 64$  pixeli, convertit în format RGB (dacă este necesar) și normalizat prin scalare în intervalul  $[0, 1]$ . Acest proces asigură consistența dimensională a inputului pentru stratul *TimeDistributed*.

Structura de date de antrenare este organizată în perechi  $(X, y)$ , unde  $X$  este tensorul de formă  $[batch\_size, 20, 64, 64, 3]$ , iar  $y$  este o etichetă binară asociată întregii secvențe. Etichetarea este realizată pe baza unei convenții clare: 1 pentru execuții corecte și 0 pentru execuții incorecte. Lista de ID-uri video și etichete este încărcată dintr-un fișier *.json*, utilizat de clasa *VideoDataGenerator*, care gestionează mini-batch-urile și amestecarea eșantioanelor la începutul fiecărei epoci. Generatorul permite antrenarea eficientă pe GPU chiar și în condițiile unui set de date de dimensiune mare, evitând încărcarea completă în memorie.

Un aspect esențial în antrenarea secvențială este păstrarea ordinii cronologice a cadrelor pentru ca rețeaua LSTM să poată învăța tipare de mișcare. Spre deosebire de CNN simplu, unde ordinea datelor nu influențează ieșirea, în cazul CNN-LSTM o permutare greșită a cadrelor ar afecta complet dinamica internă a celulelor LSTM. De aceea, pipeline-ul de preprocesare asigură menținerea secvenței originale și sincronizarea cu eticheta corectă a fiecărui videoclip.

Modelul este compilat utilizând optimizatorul *Adam*, cu o rată de învățare de 0.0001, parametru ales empiric pentru a asigura o convergență lentă și stabilă, fără oscilații bruște ale pierderii. Funcția de pierdere utilizată este *binary\_crossentropy*, adecvată clasificării binare. Pentru regularizare, în rețea este integrat un strat *Dropout* cu rată 0.5, aplicat înainte de ieșire. În plus, este activată strategia *EarlyStopping*, care întrerupe antrenarea după 5 epoci consecutive fără îmbunătățire a performanței pe setul de validare și restaurează automat greutatea modelului care au obținut scorul optim.

Antrenarea completă s-a realizat pe 25 de epoci, cu batch-uri de 8 secvențe video, parametri aleși astfel încât să optimizeze utilizarea resurselor disponibile. În timpul antrenării, sunt monitorizate metricele de performanță: acuratețea, precizia și recall-ul, iar la final modelul este salvat în format *.h5*, standardul Keras pentru stocarea eficientă a greutăților și structurii rețelei, facilitând reutilizarea ulterioară în etapa de inferență sau integrarea într-un API.

### 2.1.2.3 Clasificarea imaginilor pe baza landmark-urilor folosind XGBoost

**Arhitectura modelului.** În contrast cu modelele CNN și CNN-LSTM care procesează direct imagini sau cadre video, clasificatorul XGBoost folosește o abordare bazată pe caracteristici extrase numeric din datele vizuale. Aceste trăsături sunt generate cu ajutorul bibliotecii *MediaPipe Pose*, care identifică automat 33 de puncte anatomice (landmark-uri) pentru fiecare imagine de intrare. Din aceste puncte, sunt calculate unghiuri relevante pentru evaluarea posturii: unghiuri de flexie ale spatelui, genunchilor, brațelor și înclinări ale trunchiului, pentru ambele părți ale corpului. În total, pentru fiecare imagine se generează un vector de 10 trăsături numerice:

- 5 unghiuri calculate pe partea stângă: *back\_angle\_L*, *knee\_angle\_L*, *arm\_angle\_L*, *trunk\_angle\_L*, *torso\_bend\_L*;
- 5 unghiuri echivalente pentru partea dreaptă: *back\_angle\_R*, *knee\_angle\_R*, *arm\_angle\_R*, *trunk\_angle\_R*, *torso\_bend\_R*.

Acești indicatori biomecanici sunt ulterior folosiți ca intrare pentru modelul *XGBoost* – o metodă de clasificare de tip *gradient boosting*, cunoscută pentru acuratețea ridicată și rezistența la overfitting. XGBoost construiește un ansamblu de arbori de decizie, fiecare nou arbore încercând să corecteze erorile celor anteriori. Modelul învață astfel o funcție complexă de decizie care separă postura corectă de cea incorectă.

**Preprocesarea datelor și procesul de antrenare.** Procesul de antrenare al modelului XGBoost începe cu aplicarea algoritmului *MediaPipe Pose* asupra fiecărei imagini

din setul de date. Pentru fiecare imagine procesată, sunt extrase 33 de puncte articulare (landmark-uri), reprezentând pozițiile normalizate (x, y) ale articulațiilor esențiale ale corpului uman. Pe baza acestor coordonate, sunt calculate automat 10 unghiuri biomecanice cheie, reprezentând flexii, înclinări și aliniamente relevante pentru evaluarea execuției exercițiului fizic. Calculul se realizează prin funcții trigonometrice aplicate pe triplete de landmark-uri (de exemplu, umăr–șold–genunchi pentru unghiul spatelui), folosind formula:

$$\theta = \arccos \left( \frac{\vec{BA} \cdot \vec{BC}}{\|\vec{BA}\| \cdot \|\vec{BC}\|} \right) \quad (2.2)$$

unde  $\cdot$  reprezintă produsul scalar, iar  $\|\cdot\|$  este norma vectorului. Rezultatul este exprimat în grade și reflectă unghiul biomecanic dintre segmentele corporale relevante.

Ulterior, rezultatele sunt salvate într-un fișier *CSV*, unde fiecare rând corespunde unei imagini, iar coloanele conțin valorile unghiurilor extrase.

Etichetarea exemplelor nu se face doar manual sau binar, ci este rafinată printr-o metodă numerică de scorare. Se definesc valori „ideale” pentru fiecare unghi biomecanic, corespunzătoare unei execuții corecte. Pentru fiecare imagine, se calculează distanța medie pătratică (Mean Squared Error) între vectorul unghiurilor sale și valorile ideale. Acest scor de abatere este apoi convertit într-un scor calitativ pe o scală de la 1 la 10, folosind praguri prestabilite. Ulterior, scorurile sunt grupate în trei clase:

- 1–4.9: Execuție slabă;
- 5–7.9: Execuție parțial corectă;
- 8–10: Execuție corectă.

După procesul de etichetare automată a posturilor în trei clase (corect, parțial corect și slab), distribuția setului de date a fost puternic dezechilibrată, cu o predominanță a clasei „corect”. Pentru a elimina acest dezechilibru și a asigura o învățare echitabilă, s-a aplicat o tehnică de tip *oversampling* cu înlocuire (*sampling with replacement*). Mai exact, pentru fiecare clasă inferioară numeric au fost generate instanțe suplimentare prin replicarea aleatorie a exemplorilor existente până la atingerea unui număr țintă comun de aproximativ 7000 de instanțe per clasă. Rezultatul a fost un set de date echilibrat, cu distribuție uniformă între cele trei categorii, salvat în format *CSV* pentru a putea fi utilizat în etapa de antrenare a modelului XGBoost. Această metodă simplă, dar eficientă, contribuie la reducerea biasului de învățare și permite clasificatorului să învețe distincții semnificative între clase, fără a fi influențat disproporționat de clasa majoritară.

După echilibrarea setului de date, fișierul *CSV* a fost utilizat ca sursă principală pentru antrenarea modelului XGBoost. Acest fișier conține, pentru fiecare imagine, un vector de 10 trăsături numerice reprezentând unghiuri biomecanice calculate din pozițiile landmark-urilor corporale. Coloana *eticheta* indică clasa din cele trei posibile: *corect*, *mediu* sau *slab*.

Setul de date a fost încărcat într-un obiect *DataFrame*, iar trăsăturile (X) au fost separate de etichete (y). Deoarece etichetele erau inițial de tip string, acestea au fost codificate numeric folosind *LabelEncoder*, rezultând o mapare de forma:

$$\text{corect} \rightarrow 0, \quad \text{mediu} \rightarrow 1, \quad \text{slab} \rightarrow 2$$

Setul complet de date a fost împărțit în proporție de 80% pentru antrenare și 20% pentru testare, cu stratificare după clase pentru a menține distribuția proporțională între seturi. A fost utilizat clasificatorul *XGBClassifier* cu metrica de evaluare *mlogloss* (log-loss pentru clasificare multi-clasă).

După antrenare, modelul obținut, împreună cu instanța *LabelEncoder* necesară decodificării predicțiilor, au fost salvate într-un fișier *.joblib*, care este optimizată pentru stocarea obiectelor Python de dimensiuni mari (cum sunt arborii de decizie sau modelele de tip ensemble). Formatul *.joblib* este preferat în cazul modelelor XGBoost datorită vitezei de serializare superioare și compatibilității directe cu aplicațiile de producție.

### 2.1.3 Metode de evaluare a performanței modelului

Evaluarea performanței modelelor de învățare automată este esențială pentru a determina eficiența și aplicabilitatea acestora într-un context real. În cadrul acestei lucrări, pentru toate cele trei modele dezvoltate (CNN, CNN-LSTM și XGBoost), au fost utilizate metrice standard din domeniul clasificării, aplicate pe seturile de testare separate în prealabil din datele inițiale. Cele mai relevante metode de evaluare utilizate au fost:

- *Acuratețea* (eng. *accuracy*) – reprezintă proporția de predicții corecte în raport cu numărul total de exemple testate. Este un prim indicator global al performanței, însă poate fi înșelător în cazul datelor dezechilibrate.
- *Precizia* (eng. *precision*) – măsoară proporția de exemple prezise ca „pozitive” care sunt într-adevăr corecte. Este relevantă în special atunci când penalizările pentru clasificări false pozitive sunt mari.
- *Rapel* (eng. *recall*) – reflectă proporția de exemple pozitive din totalul real care au fost corect identificate de model. Este esențială în contexte unde este important să detectezi toate cazurile pozitive.
- *Scorul F1* (eng. *F1 score*) – media armonică între precizie și recall, fiind o metrică echilibrată care penalizează extrem de sever discrepanțele mari între cele două.
- *Matricea de confuzie* – oferă o reprezentare detaliată a numărului de predicții corecte și incorecte pentru fiecare clasă, permițând analiza tipurilor de erori comise de model (de exemplu, confundarea clasei „slab” cu „mediu”).

Pentru modelele CNN și CNN-LSTM, evaluarea a fost realizată pe seturi separate de imagini și videoclipuri, etichetate manual. Pentru XGBoost, evaluarea a fost efectuată pe trăsăturile extrase numeric din imagini folosind MediaPipe, iar interpretarea a fost realizată folosind un raport complet de clasificare generat cu ajutorul bibliotecii *scikit-learn*.

Aceste metode oferă o imagine de ansamblu clară asupra performanței fiecărui model și permit compararea obiectivă între diferitele arhitecturi și abordări testate. În

contextul unei aplicații educaționale, prioritizarea scorului F1 și a recall-ului este esențială, întrucât o predicție incorectă asupra unei posturi greșite poate însemna lipsa unui feedback corectiv necesar.

## 2.2 Rezultate experimentale și analiza acestora

În această secțiune sunt prezentate rezultatele obținute în urma antrenării și evaluării fiecărui model de clasificare implementat în cadrul aplicației. Evaluările au fost realizate separat, pe seturi de date distincte, în funcție de tipul de intrare și arhitectura utilizată. În continuare sunt detaliate performanțele celor trei modele: modelul CNN pentru imagini statice, modelul CNN-LSTM pentru secvențe video și modelul XGBoost antrenat pe trăsături numerice extrase cu MediaPipe. Rezultatele obținute sunt cuantificate utilizând metricile de evaluare discutate în Subsecțiunea anterioară 2.1.3, respectiv acuratețea, precizia, recall-ul, scorul F1 și matricea de confuzie, care oferă o perspectivă completă asupra performanței fiecărui model.

**Analiza rezultatelor modelului CNN** Modelul *CNN* simplificat a fost antrenat pe imagini statice corespunzătoare exercițiului *barbell row*, având ca scop clasificarea execuțiilor în două clase: *corect* și *incorect*. Fiecare imagine a fost preprocesată prin redimensionare la dimensiunea  $128 \times 128$  pixeli și scalată în intervalul  $[0, 1]$ . Evaluarea performanței modelului a fost realizată pe un set de testare echilibrat, iar rezultatele sunt prezentate în Tabela 2.1. Precizia generală a modelului a fost de aproximativ 58.3%, ceea ce indică o capacitate limitată a arhitecturii CNN de a generaliza pe date noi, în special în cazul imaginilor în care postura este parțial mascată sau capturată din unghiuri variate.

Tabela 2.1: Raport de clasificare pentru modelul CNN - Barbell Row

Clasă	Precizie	Recall	F1-score	Support
<i>corect</i>	0.58	0.37	0.45	624
<i>incorect</i>	0.58	0.77	0.66	712
<b>Acuratețe</b>	0.583			

Se observă o acuratețe globală modestă de aproximativ 58.3%, ceea ce indică o capacitate limitată a modelului de a generaliza pe datele de test. Scorul F1 obținut pentru clasa „*incorect*” (0.66) este semnificativ mai mare decât cel pentru clasa „*corect*” (0.45), sugerând că modelul a învățat mai bine să identifice execuțiile greșite, cel mai probabil datorită caracteristicilor vizuale mai evidente ale acestor instanțe.

Matricea de confuzie prezentată în Figura 2.3 oferă o imagine detaliată a distribuției predicțiilor. Din totalul de 624 exemple corecte, doar 232 au fost clasificate corect, în timp ce 392 au fost etichetate greșit ca incorecte. În schimb, dintre cele 712 execuții incorecte, 547 au fost clasificate corect. Acest comportament indică o tendință a modelului de a favoriza clasa „*incorect*”, ceea ce poate fi explicat atât prin arhitectura

relativ simplă a CNN-ului, cât și prin dificultatea de a extrage trăsături relevante din imagini statice fără o analiză contextuală temporală.

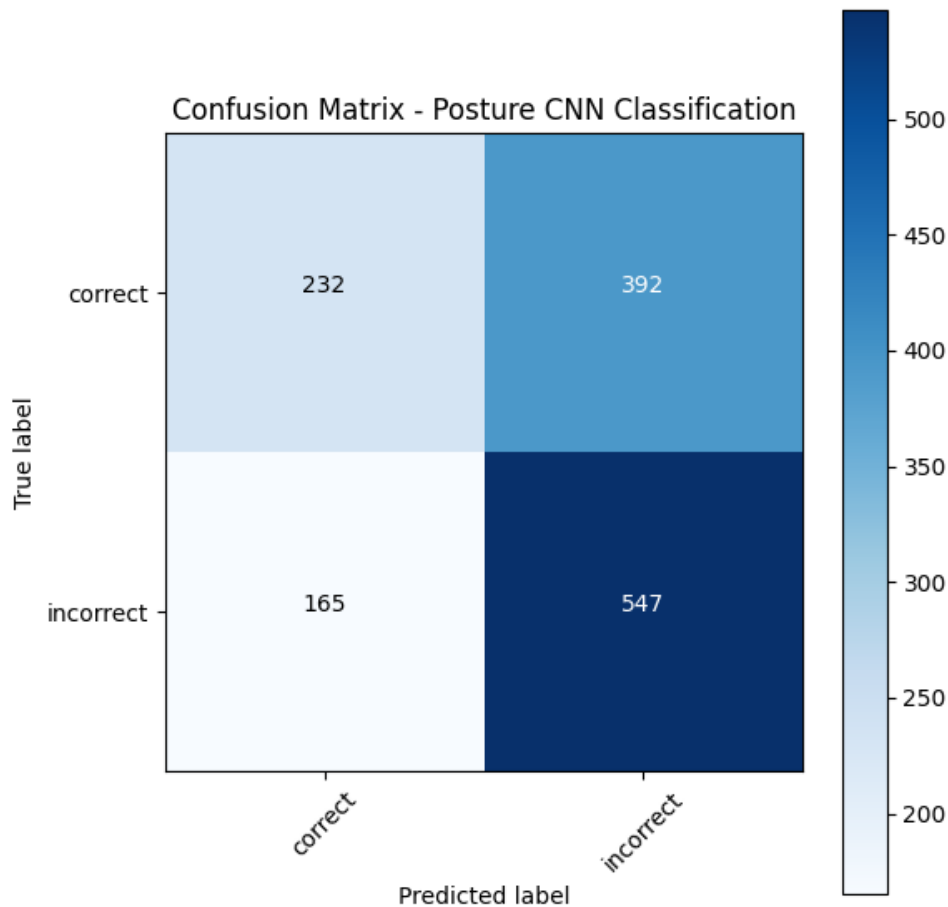


Figura 2.3: Matrice de confuzie pentru modelul CNN – Barbell row

Deși modelul CNN reușește să identifice într-o anumită măsură execuțiile incorecte, performanța sa generală este limitată, fapt ce evidențiază necesitatea unor arhitecturi mai complexe sau a unor strategii suplimentare de preprocesare a imaginilor pentru a obține rezultate competitive.

**Analiza rezultatelor modelului CNN+LSTM** Modelul *CNN+LSTM* a fost conceput pentru a analiza succesiuni video formate din 20 de cadre extrase uniform dintr-o înregistrare, fiecare cadru fiind redimensionat la  $128 \times 128$  pixeli și normalizat în intervalul  $[0, 1]$ . Arhitectura modelului combină trei blocuri convoluționale, fiecare urmat de operații de normalizare și pooling, integrate într-o structură *TimeDistributed* care asigură aplicarea consistentă a CNN-ului peste fiecare cadru individual. Ulterior, secvența de trăsături extrase este transmisă unui strat LSTM care învață relațiile temporale dintre cadre, modelând astfel dinamica mișcării în timp.

Evaluarea modelului s-a realizat pe setul de date aferent exercițiului *Overhead Press*, cu clasificare binară între execuții corecte și incorecte. După antrenarea modelului timp

de 25 de epoci, s-au obținut următoarele rezultate pe setul de test: o acuratețe globală de 54.5%, un scor F1 de 0.60 pentru clasa „corect” și de 0.48 pentru clasa „incorect” (vezi Tabelul 2.2). Aceste performanțe sugerează că modelul are dificultăți în a distinge precis execuțiile incorecte, probabil din cauza diversității scăzute a erorilor în setul de date sau a dezechilibrului între clase.

Tabela 2.2: Raport de clasificare pentru modelul CNN+LSTM – Overhead Press

Clasă	Precizie	Recall	F1-score	Support
<i>correct</i>	0.61	0.58	0.60	195
<i>incorrect</i>	0.47	0.49	0.48	144
<b>Acuratețe</b>	0.55			

Matricea de confuzie din Figura 2.4 oferă o perspectivă mai detaliată: modelul a reușit să clasifice corect 114 din cele 195 de instanțe etichetate ca *corect*, dar a etichetat greșit 81 dintre acestea ca *incorect*. Pe de altă parte, dintre cele 144 de exemple incorecte, doar 71 au fost recunoscute ca atare, restul fiind clasificate eronat. Acest comportament indică o ușoară tendință a modelului de a favoriza clasa „corect”, posibil și din cauza unei distribuții dezechilibrate în timpul antrenării, sau din cauza similarității vizuale între unele execuții corecte și cele incorecte.

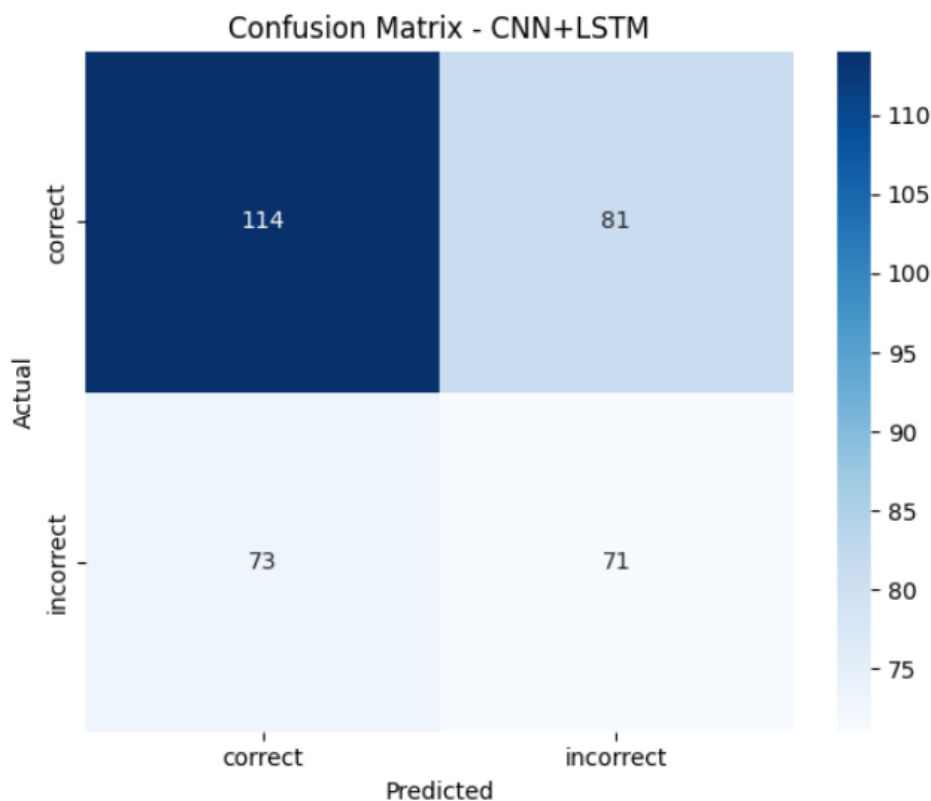


Figura 2.4: Matrice de confuzie pentru modelul CNN+LSTM – Overhead Press

De remarcat este faptul că această arhitectură a fost utilizată și pentru analiza exer-

cițiului *Squat*, folosind un set de date video structurat identic. Deși rezultatele pentru *Squat* nu sunt prezentate în această secțiune, testele preliminare au arătat o performanță comparabilă, cu o ușoară îmbunătățire a scorului F1 pentru clasa incorectă, datorită variației mai vizibile între execuții corecte și greșite.

**Analiza rezultatelor modelului XGBoost** Modelul *XGBoost* a fost antrenat pe date extrase prin MediaPipe din imagini etichetate în funcție de calitatea execuției exercițiului *barbell row*, clasificarea având trei clase: *corect*, *mediu* și *slab*. Rezultatele obținute pe setul de test sunt prezentate în Tabelul 2.3.

Tabela 2.3: Raport de clasificare pentru modelul XGBoost - Barbell Row

Clasă	Precizie	Recall	F1-score	Support
<i>corect</i>	0.96	0.99	0.97	1501
<i>mediu</i>	0.99	0.95	0.97	1501
<i>slab</i>	0.99	1.00	0.99	1501
<b>Acuratețe</b>	0.98			

Se observă o acuratețe generală de 98%, cu scoruri F1 foarte ridicate pentru toate cele trei clase. Clasa „*slab*” este recunoscută perfect de model, având un *recall* de 1.00 și o precizie de 0.99, ceea ce sugerează că exemplele din această categorie sunt foarte bine delimitate. Pentru clasa „*corect*”, modelul obține un *recall* de 0.99, dar o precizie mai scăzută (0.96), ceea ce înseamnă că unele instanțe „medii” sunt uneori clasificate greșit ca „corecte”.

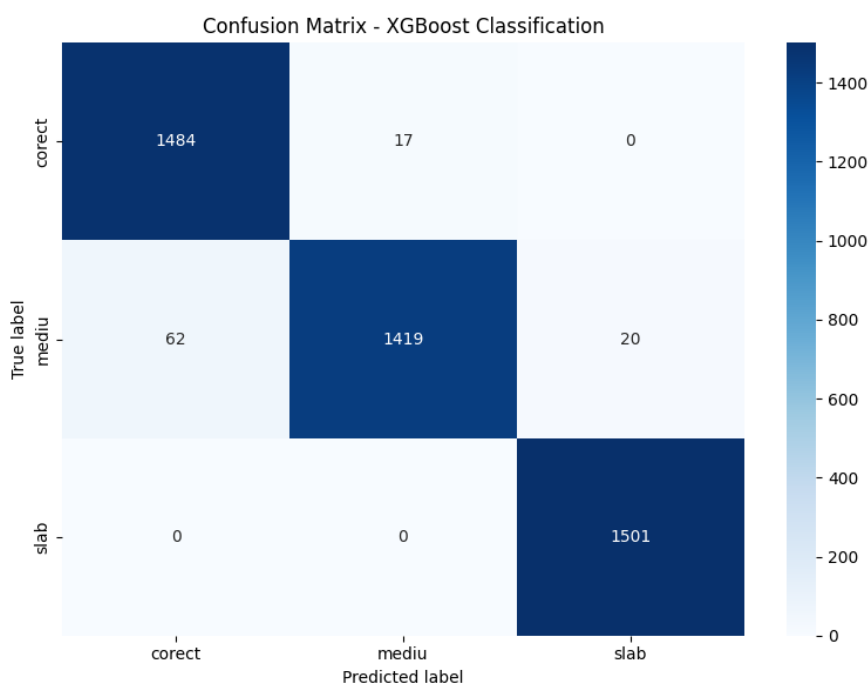


Figura 2.5: Matricea de confuzie pentru modelul XGBoost



Matricea de confuzie din Figura 2.5 oferă o perspectivă detaliată asupra distribuției predicțiilor. Cele mai multe erori apar între clasele „corect” și „mediu”, unde 62 de exemple „mediu” au fost clasificate greșit ca „corect”, și 17 cazuri „corect” au fost clasificate ca „mediu”. Nu s-au observat confuzii între clasa „slab” și celelalte, ceea ce validează capacitatea modelului de a identifica execuțiile semnificativ greșite.

Aceste rezultate confirmă robustețea modelului XGBoost în clasificarea posturilor bazate pe trăsături biomecanice extrase numeric. Mai mult, timpul de antrenare redus și dimensiunea compactă a modelului salvat în format *.joblib* îl recomandă pentru integrarea în aplicații web, fiind ideal pentru execuții rapide pe servere fără GPU.

**Compararea rezultatelor** Analizând performanțele celor trei modele implementate în cadrul aplicației *FitPose AI*, se poate observa o diferență semnificativă între arhitecturile bazate pe rețele neuronale convoluționale și clasificatorul XGBoost. Modelul XGBoost, antrenat pe trăsături biomecanice extrase numeric din imagini, a atins o acuratețe generală de 98% și scoruri F1 peste 0.97 pentru toate cele trei clase. Această performanță remarcabilă este susținută de claritatea trăsăturilor extrase (unghiuri și distanțe între articulații), care oferă o reprezentare abstractă, dar eficientă a execuției exercițiilor.

În schimb, modelele CNN și CNN+LSTM au înregistrat rezultate modeste, cu o acuratețe de 58.3% și respectiv 55%. În cazul CNN-ului aplicat pe imagini statice, dificultatea majoră constă în lipsa unei secvențialități temporale și în ambiguitățile vizuale generate de unghiuri de captură diferite. De asemenea, modelul CNN+LSTM, deși teoretic capabil să capteze dinamica execuției, nu a atins performanțe comparabile cu XGBoost, posibil din cauza unui set de date video limitat sau a etichetării subiective a execuțiilor. Totuși, rezultatele obținute demonstrează că arhitectura este promițătoare, cu scoruri F1 echilibrate între clase și un comportament stabil, ceea ce justifică extinderea acestei abordări și pentru alte exerciții.

Tabela 2.4: Comparația performanțelor celor trei modele de clasificare

Model	Tip date	Nr. clase	Acuratețe	Scor F1 (mediu)
CNN	Imagini	2 (corect/incorect)	58.3%	0.55
CNN+LSTM	Video	2 (corect/incorect)	55.0%	0.54
XGBoost	Valori numerice	3 (corect/mediu/slab)	98.0%	0.97

Comparativ cu lucrările din literatura de specialitate, performanțele obținute cu XGBoost se aliniază sau chiar depășesc unele rezultate raportate. Spre exemplu, în studiul realizat de Pan [Pan22], pentru exercițiul *squat*, o rețea CNN combinată cu RoI Align și un sistem de evaluare bazat pe scoruri obține o acuratețe de 92%. Totodată, Kotte [KKDT23], utilizând YOLOv7-pose și un sistem de clasificare bazat pe estimarea unghiurilor, au obținut o acuratețe de aproximativ 95%. Comparativ, modelul XGBoost propus în această lucrare atinge o performanță superioară, fără a necesita procesări vizuale complexe sau rețele adânci de detecție.

În ceea ce privește modelele CNN și CNN+LSTM, acestea oferă o soluție viabilă pentru detecție directă pe imagini brute sau video, dar cu costuri computaționale mai ridicate și o nevoie mai mare de date etichetate și augmentate. Spre exemplu, în lucrarea lui Boyalla [Boy21], care utilizează OpenPose pentru analiza posturii, scorul F1 obținut pentru detectarea execuțiilor incorecte nu depășește 0.85. În comparație, arhitectura CNN+LSTM propusă în această lucrare, deși mai simplă, reușește să capteze unele tipare temporale, iar acuratețea sa poate fi crescută printr-o extindere a setului de date sau prin fine-tuning specific pe exerciții.

Așadar, modelul XGBoost se remarcă drept cel mai performant în contextul aplicației actuale, datorită generalizării bune, a timpului redus de inferență și a interpretabilității rezultatelor. Totuși, modelele neuronale oferă potențial ridicat pentru analiză vizuală avansată, mai ales în combinație cu rețele de detecție și estimare a poziției, urmând a fi perfecționate în lucrări viitoare.

## 2.3 Îmbunătățiri ulterioare

În actuala implementare a aplicației, au fost utilizate trei tipuri de modele de clasificare: rețele neuronale convoluționale (CNN) pentru imagini statice, arhitecturi hibride CNN-LSTM pentru videoclipuri și un clasificator XGBoost antrenat pe un set redus de trăsături numerice. Deși aceste modele oferă rezultate satisfăcătoare, există multiple direcții de optimizare și extindere care pot contribui la creșterea performanței și a capacității de generalizare.

În cazul modelului CNN, utilizat pentru clasificarea imaginilor statice, una dintre direcțiile principale de îmbunătățire este creșterea profunzimii rețelei prin adăugarea de noi straturi convoluționale sau utilizarea unor arhitecturi pre-antrenate precum MobileNetV2, EfficientNet sau ResNet, aplicate prin tehnici de transfer learning. Aceste rețele oferă un grad ridicat de generalizare datorită antrenării pe seturi de date complexe precum ImageNet. O altă variantă de extindere constă în realizarea unui fine-tuning pe seturi specializate, cu exerciții fizice, pentru a adapta rețeaua la specificul domeniului postural. În plus, se poate îmbunătăți procesul de augmentare a datelor, înlocuind transformările simple (rotații, flip-uri) cu distorsiuni sintetice realiste sau simulări de zgomot vizual. De asemenea, creșterea rezoluției imaginilor de la  $128 \times 128$  pixeli la dimensiuni mai mari, precum  $224 \times 224$ , poate furniza rețelei un spațiu vizual mai bogat pentru învățarea trăsăturilor fine.

Pentru arhitectura CNN-LSTM, utilizată în clasificarea videoclipurilor, performanța poate fi îmbunătățită prin creșterea numărului de cadre procesate pe secvență, oferind astfel o acoperire mai completă a mișcării analizate. În locul stratului LSTM pot fi utilizate unități GRU (Gated Recurrent Unit), care au un comportament similar, dar sunt mai eficiente din punct de vedere computațional. O alternativă modernă constă în înlocuirea LSTM cu un encoder de tip Transformer, ce poate modela mai eficient relațiile temporale fără pierderi de context. Integrarea unui mecanism de atenție (attention) ar permite modelului să identifice cadrele cele mai relevante pentru luarea deciziei finale, accentuând secvențele în care apar deviații posturale. În plus, modelul

poate fi extins cu rețele convoluționale 3D (ex. C3D, I3D), care tratează simultan informația spațială și temporală, oferind o alternativă complet integrată față de abordarea separată CNN + LSTM. De asemenea, este posibilă tranziția de la clasificare binară la evaluarea continuă, cu scoruri pe o scară de la 1 la 10, reflectând mai fidel calitatea execuției.

În ceea ce privește modelul XGBoost, bazat pe trăsături numerice extrase cu MediaPipe, o direcție importantă de extindere este creșterea numărului de trăsături utilizate. Pe lângă unghiurile statice actuale, se pot adăuga distanțe relative între articulații, variații de unghiuri în timp și derivate temporale, oferind o imagine mai detaliată a dinamicii mișcării. Modelul poate fi extins către învățarea multi-label, astfel încât să nu returneze doar un scor general, ci și să identifice regiunile anatomice responsabile de abaterea posturală. Pentru a crește interpretabilitatea, se pot integra tehnici de explicabilitate precum SHAP, care oferă o justificare numerică a deciziei modelului. În final, modelul poate fi generalizat pentru a funcționa pe mai multe tipuri de exerciții, fie prin antrenarea unor clasificatori separați, fie prin adăugarea unei trăsături suplimentare ce indică tipul exercițiului analizat.

Toate aceste direcții de îmbunătățire urmăresc creșterea performanței, robusteții și capacității de interpretare a modelelor integrate în aplicația dezvoltată. Prin extinderea seturilor de date, rafinarea arhitecturilor și integrarea unor mecanisme avansate de învățare, aplicația poate evolua către un instrument scalabil, validabil în contexte clinice și educaționale, cu o utilitate extinsă pentru utilizatori diverși.

# Capitolul 3

## *FitPose AI* - Aplicația software

În acest capitol este prezentată componenta practică a lucrării, și anume aplicația web *FitPose AI*, dezvoltată pentru a facilita evaluarea posturii utilizatorilor în timpul efectuării exercițiilor fizice. Această aplicație îmbină tehnici de procesare vizuală cu algoritmi de învățare profundă pentru a detecta automat execuții incorecte și a furniza feedback corectiv. Structura aplicației este modulară, permițând extinderea acesteia cu noi tipuri de exerciții sau algoritmi de analiză. În cele ce urmează, sunt detaliate componentele software utilizate, procesul de proiectare și implementare, precum și opțiunile de utilizare ale aplicației.

### 3.1 Descrierea sistemului

Acest capitol prezintă aplicația web propusă, denumită *FitPose AI*, destinată evaluării posturii în timpul execuției exercițiilor fizice, cu scopul de a sprijini utilizatorii în identificarea abaterilor posturale și în corectarea acestora. Aplicația permite încărcarea de fișiere vizuale (imagini sau videoclipuri), analiza acestora cu ajutorul unui model de inteligență artificială și furnizarea unui feedback interpretabil și personalizat. Soluția propusă își propune să faciliteze crearea unui sistem accesibil, scalabil și intuitiv, adaptabil la diferite tipuri de exerciții fizice.

Originalitatea acestei aplicații constă în integrarea unei arhitecturi complete frontend-backend-AI, care permite utilizatorului să interacționeze cu mai multe modele de clasificare a execuției exercițiilor fizice fără a necesita cunoștințe tehnice avansate. De asemenea, una dintre contribuțiile notabile constă în oferirea unui feedback bazat pe fișiere etichetate manual (fișiere *JSON*), care reflectă prezența sau absența unor erori specifice în execuția exercițiilor.

Pentru dezvoltarea aplicației au fost utilizate tehnologii moderne și optimizate. Interfața a fost dezvoltată folosind ReactJS, oferind o experiență fluidă și adaptabilă atât pe desktop, cât și pe dispozitive mobile. Aplicația este structurată în pagini individuale corespunzătoare etapelor de interacțiune: selecția exercițiului, încărcarea fișierului și vizualizarea rezultatului. Designul a fost realizat cu accent pe simplitate, claritate vizuală și accesibilitate.

Partea de backend este prevăzută pentru a integra trei modele de clasificare, doua

dintre ele bazate pe rețele neuronale convoluționale (CNN), iar cel de-al treilea un model de învățare automată (XGBoost), antrenate separat și salvate sub forma unor fișiere *.h5* sau *.joblib*. Acestea vor fi integrate ulterior într-un server Python (folosind FastAPI) care va permite transmiterea fișierelor din interfață către modelul ales, primirea rezultatului și returnarea feedback-ului.

În ansamblu, sistemul *FitPose AI* își propune să fie un instrument eficient și extensibil în domeniul evaluării posturii, ușor de utilizat și adaptabil unor scenarii variate (fitness individual, kinetoterapie, educație fizică etc.).

## 3.2 Analiză și proiectare

În urma cercetării literaturii de specialitate privind evaluarea posturii în contextul exercițiilor fizice, s-a constatat lipsa unei aplicații intuitive care să permită utilizatorilor să încarce imagini sau videoclipuri și să primească feedback automatizat în timp real, bazat pe modele de inteligență artificială. Aplicația propusă acoperă această nevoie, oferind posibilitatea de a analiza execuția exercițiilor cu ajutorul rețelelor neuronale sau al clasificatorilor de tip XGBoost, permițând astfel o evaluare obiectivă și scalabilă, aplicabilă atât în domeniul fitness-ului, cât și în cel educațional sau medical (kinetoterapie, prevenție posturală etc.).

Faza de planificare a constat în identificarea cerințelor funcționale și nefuncționale ale aplicației, materializate într-un model de cazuri de utilizare. Figura 3.1 prezintă diagrama de tip *Use Case Diagram* care surprinde principalele interacțiuni ale utilizatorului cu sistemul. Este important de menționat că operațiile de evaluare presupun existența unei sesiuni active și a unui fișier valid.

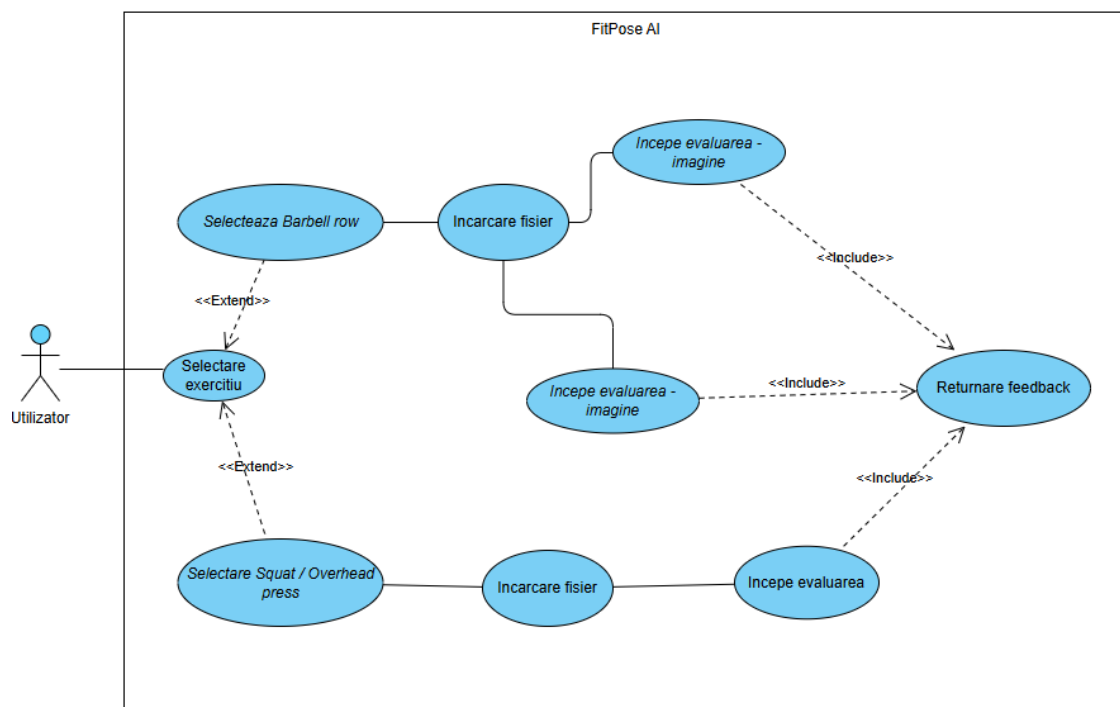


Figura 3.1: Diagrama de cazuri de utilizare pentru aplicația FitPose AI

Se poate observa utilizatorul ca fiind singurul actor al diagramei alături de cazurile de utilizare identificate:

- Selectare exercițiu – utilizatorul alege tipul de exercițiu ce urmează a fi analizat.
- Încărcare fișier – permite încărcarea unei imagini sau a unui videoclip.
- Începe evaluarea – utilizatorul poate alege tipul de model dorit pentru analiza imaginilor statice, CNN sau XGBoost, sau implicit pentru analiza videoclipurilor, CNN+LSTM.
- Returnare rezultat și afișare feedback – rezultatul este procesat și afișat sub formă de etichetă/score.

Diagrama cazurilor de utilizare oferă o privire de ansamblu asupra principalelor funcționalități din aplicația *FitPose AI*, dar nu reflectă direct modul în care acestea sunt implementate. Pentru a înțelege mai bine organizarea internă a aplicației, este necesară descrierea structurii logice și a principiilor de proiectare care au stat la baza dezvoltării sistemului.

Deși aplicația nu este construită folosind o arhitectură orientată obiect, organizarea sa respectă principiul fundamental al modularității. Codul este împărțit în module independente: antrenarea modelelor de inteligență artificială este realizată în fișiere *Python* separate, backend-ul este implementat folosind *FastAPI* într-un mod minimalist, iar interfața web este construită cu *React*, respectând separarea clară dintre logică și prezentare. Această structurare contribuie la scalabilitate și ușurința în mentenanță. De asemenea, sistemul respectă principiul *Separation of Concerns* (SoC), fiecare componentă având o responsabilitate unică și bine delimitată: interfața utilizatorului gestionează interacțiunea vizuală, backend-ul preia fișierele și direcționează cererile, iar modulele de inteligență artificială realizează inferența.

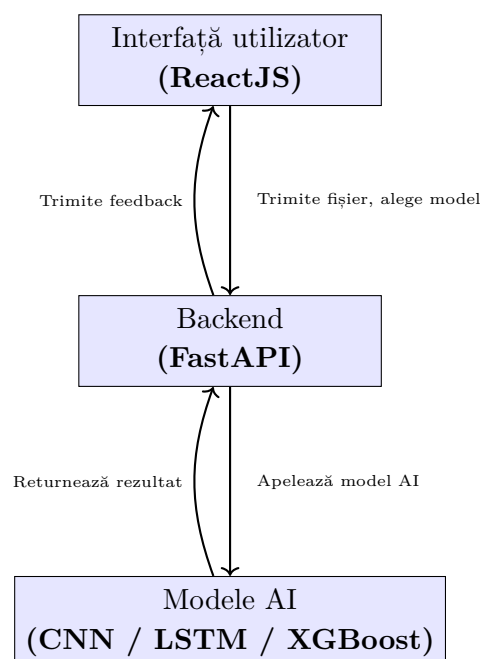


Figura 3.2: Aplicarea principiului *Separation of Concerns* în arhitectura aplicației

Această separare logică permite înlocuirea sau îmbunătățirea unei componente fără a afecta întregul sistem.

În ceea ce privește procesul de dezvoltare, aplicația *FitPose AI* a fost construită incremental, printr-o abordare centrată pe testarea componentelor individuale. Inițial au fost antrenate și validate modelele de inteligență artificială, urmate de implementarea și testarea endpoint-ului */predict* din backend-ul FastAPI, iar în final a fost integrată interfața ReactJS. Această succesiune a permis o dezvoltare controlată și o depanare eficientă, fiecare etapă fiind validată înainte de trecerea la următoarea.

Un aspect esențial al aplicației îl constituie interfața API standardizată, definită sub forma unui endpoint principal */predict*, care primește cereri de tip POST cu un fișier (image sau video) și metadatele asociate (exercițiu, tip model). Acesta reprezintă un punct de legătură între interfața utilizatorului și logica AI, permițând extinderea aplicației fără modificarea backend-ului, prin integrarea într-o aplicație mobilă sau într-un dashboard educațional.

Astfel, printre avantajele principale ale aplicației se numără modularitatea, scalabilitatea și claritatea arhitecturii software. Separarea componentelor (frontend, backend, modele AI) permite dezvoltarea și testarea lor independentă, iar utilizarea unor tehnologii moderne precum FastAPI și ReactJS contribuie la performanța și interope-rabilitatea aplicației. De asemenea, integrarea mai multor tipuri de modele AI (CNN, LSTM, XGBoost) oferă flexibilitate și adaptabilitate în funcție de scenariul analizat.

## 3.3 Implementare

Etapa de implementare reprezintă o componentă esențială în ciclul de viață al aplicației *FitPose AI*, întrucât presupune alegerea tehnologiilor, organizarea logicii aplicației și integrarea componentelor software dezvoltate. În această secțiune vor fi detaliate deciziile tehnice, structura modulară a aplicației și particularitățile implementării atât pentru partea de backend, cât și pentru interfața web (frontend).

### 3.3.1 Frontend

Interfața grafică a aplicației *FitPose AI* a fost realizată utilizând biblioteca *React*, una dintre cele mai populare soluții open-source pentru dezvoltarea aplicațiilor web moderne. ReactJS permite construirea de interfețe modulare, component-based, ceea ce facilitează reutilizarea codului și menținerea structurii aplicației. Alegerea ReactJS a fost motivată de capacitatea acestuia de a organiza logic aplicația în componente reutilizabile, fiecare gestionând propria stare (*state*) și propriile evenimente (*event handlers*). Datorită acestui model declarativ, React oferă o dezvoltare rapidă și ușor de întreținut, chiar și în aplicații cu interacțiuni complexe.

**Structura aplicației** În cadrul aplicației *FitPose AI*, interfața este structurată în patru componente principale, fiecare având o responsabilitate bine delimitată.

Componenta *HomePage.jsx* reprezintă pagina principală a aplicației și permite utilizatorului să selecteze tipul de exercițiu pentru evaluare, inițiind astfel fluxul logic

al interacțiunii. După alegerea exercițiului, utilizatorul este redirecționat către componenta adecvată pentru încărcarea fișierului. În cazul în care se dorește încărcarea unei imagini, este utilizată componenta *UploadImage.jsx*, care permite utilizatorului să selecteze o imagine statică de pe dispozitivul local, să aleagă modelul de analiză dorit (CNN sau XGBoost) și să trimită cererea către backend-ul FastAPI. Această componentă utilizează *hooks* precum *useState* pentru gestionarea locală a fișierului încărcat și *useEffect* pentru reacții la modificări în starea aplicației. În mod analog, componenta *UploadVideo.jsx* este responsabilă pentru încărcarea fișierelor video, fiind destinată analizării exercițiului cu ajutorul arhitecturii CNN-LSTM. Aceasta include un formular simplificat care acceptă fișiere video de dimensiuni moderate și gestionează trimiterea acestora la endpoint-ul dedicat procesării temporale. După primirea rezultatului din backend, utilizatorul este redirecționat către componenta *Feedback.jsx*, care are rolul de a afișa eticheta corespunzătoare execuției (corect/incorect/mediu/slab) și eventuale sugestii de îmbunătățire. Această componentă este reutilizabilă și comună atât pentru imaginile statice, cât și pentru videoclipuri, ceea ce reflectă aplicarea principiului *DRY* (Don't Repeat Yourself) în organizarea logicii aplicației.

Navigarea între pagini este implementată cu ajutorul componentelor `<Routes>` și `<Route>`, iar fiecare pagină este încărcată în funcție de calea URL (ex: `/upload-image`, `/feedback`).

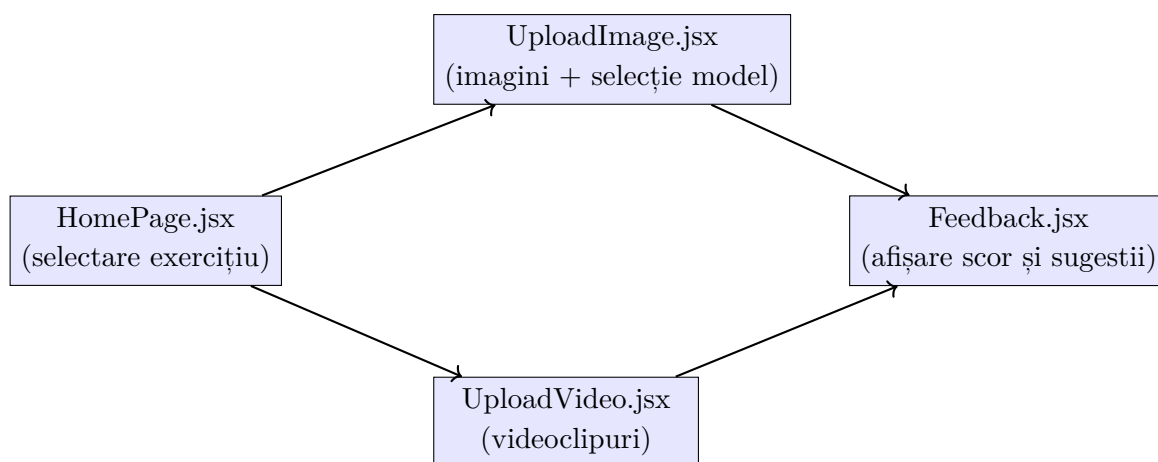


Figura 3.3: Fluxul de interacțiune între componentele frontend în React

**Stilizare și design vizual** Pentru definirea aspectului vizual al aplicației a fost utilizat limbajul *CSS* (Cascading Style Sheets), implementat separat în fișiere precum *HomePage.css*, *UploadImage.css*, *UploadVideo.css* și *Feedback.css*. Designul propus este minimalist și funcțional, punând accent pe lizibilitate și claritatea interacțiunii. Elementele sunt aranjate în mod intuitiv pe ecran, cu butoane evidențiate pentru acțiunile principale, utilizând culori contrastante pentru a ghida atenția utilizatorului. Fișierele CSS sunt modularizate per componentă, ceea ce permite menținerea stilurilor specifice fără interferențe între pagini.

Paleta cromatică a fost concepută plecând de la principiile de contrast vizual ridicat și codificare semantică a culorilor. Fundalul global al aplicației utilizează o nuanță bleumarin, aplicată uniform pentru a asigura o bază neutră și consistentă. Acest fun-



dal întunecat optimizează lizibilitatea elementelor luminoase și contribuie la reducerea solicitării vizuale în utilizare îndelungată, conform recomandărilor HCI privind *dark theme interfaces*. Elementele de interacțiune principală sunt stilizate într-un albastru saturat, care oferă vizibilitate ridicată și direcționează atenția utilizatorului către acțiunile relevante. Complementar, o nuanță de verde deschis este aplicată în mod deliberat asupra butoanelor de confirmare, sugerând acceptare sau progres pozitiv, în conformitate cu convențiile UI/UX pentru codarea culorilor în interfețe interactive.

Layout-ul este organizat folosind modelul (*Flexbox*) și aliniere verticală, permițând adaptabilitatea interfeței la rezoluții variate. Elementele sunt distribuite intuitiv, cu spațiere adecvată și dimensiuni constante ale butoanelor pentru a reduce încărcarea cognitivă a utilizatorului. De asemenea, dimensiunile și fonturile alese sunt optimizate pentru lizibilitate, inclusiv pe dispozitive mobile.

**Integrarea în arhitectura aplicației** Integrarea interfeței cu backend-ul este realizată prin intermediul cererilor *HTTP POST*, transmise către serverul FastAPI, care preia fișierele și le direcționează către modelul corespunzător pentru inferență. Răspunsul primit de la server este afișat prin intermediul componentei *Feedback.jsx*, care încorporează feedback-ul textual și stilizat (scoruri, mesaje, sugestii). Acest flux de comunicare bidirecțională dintre frontend și backend este esențial pentru menținerea unei experiențe fluide pentru utilizator.

### 3.3.2 Backend

Backend-ul aplicației *FitPose AI* este implementat utilizând *FastAPI*, un framework modern pentru dezvoltarea de aplicații web rapide în Python, conceput pentru a fi performant, asincron și compatibil cu standardul OpenAPI. Alegerea acestuia a fost motivată de mai multe avantaje: suport nativ pentru validarea tipurilor de date cu ajutorul bibliotecii *pydantic*, definirea intuitivă a endpointurilor RESTful, precum și capacitatea de a gestiona cereri asincrone – aspect esențial în contextul încărcării fișierelor mari (imagini, videoclipuri). De asemenea, FastAPI oferă integrare automată cu documentația interactivă Swagger, facilitând testarea API-ului pe durata dezvoltării.

În cadrul aplicației, backend-ul are rolul de a primi fișierele trimise de utilizator (imagini sau videoclipuri), de a le prelucra conform modelului ales și de a returna un rezultat interpretabil. Acest comportament este centralizat în fișierul *app.py*, unde este definit endpointul */predict*. Acest endpoint primește, printr-o cerere POST, un fișier și metadate asociate (*model\_type*, *exercise*), iar în funcție de valorile acestora, apelează unul dintre cele trei modele disponibile: *CNN*, utilizat pentru clasificarea imaginilor statice, încărcat la începutul aplicației din fișierul *barbellrow.h5*; *XGBoost*, utilizat pentru clasificarea pe baza trăsăturilor extrase cu MediaPipe, încărcat cu ajutorul bibliotecii *joblib* și *CNN+LSTM*, un model video, aplicat automat dacă exercițiul selectat este „Overhead Press” sau „Squat”.

Funcționalitatea de inferență este organizată modular în fișierul *predict.py*, care conține trei funcții principale: *predict\_image\_posture\_h5*, *predict\_image\_posture* și *predict\_video\_posture*. Aceste funcții sunt responsabile pentru preprocesarea fișie-

relor de intrare (inclusiv redimensionare, normalizare și extragerea trăsăturilor relevante), încărcarea modelului AI corespunzător și generarea predicției finale. Rezultatul returnat constă într-o etichetă de clasificare (ex. „corect”/„incorect” sau ”corect”/„mediu”/”slab”) și un scor numeric care reflectă încrederea modelului în predicție.

În cazul videoclipurilor, sistemul extrage automat un număr fix de 20 de cadre distribuite uniform pe durata înregistrării. Aceste cadre sunt preprocesate și grupate într-o structură de tip tensor 4D cu forma (1, 20, 64, 64, 3), corespunzătoare unui singur eșantion video format din 20 de imagini RGB de dimensiune  $64 \times 64$ . Această reprezentare este utilizată drept intrare pentru modelul CNN+LSTM, specializat în analiza secvențelor temporale.

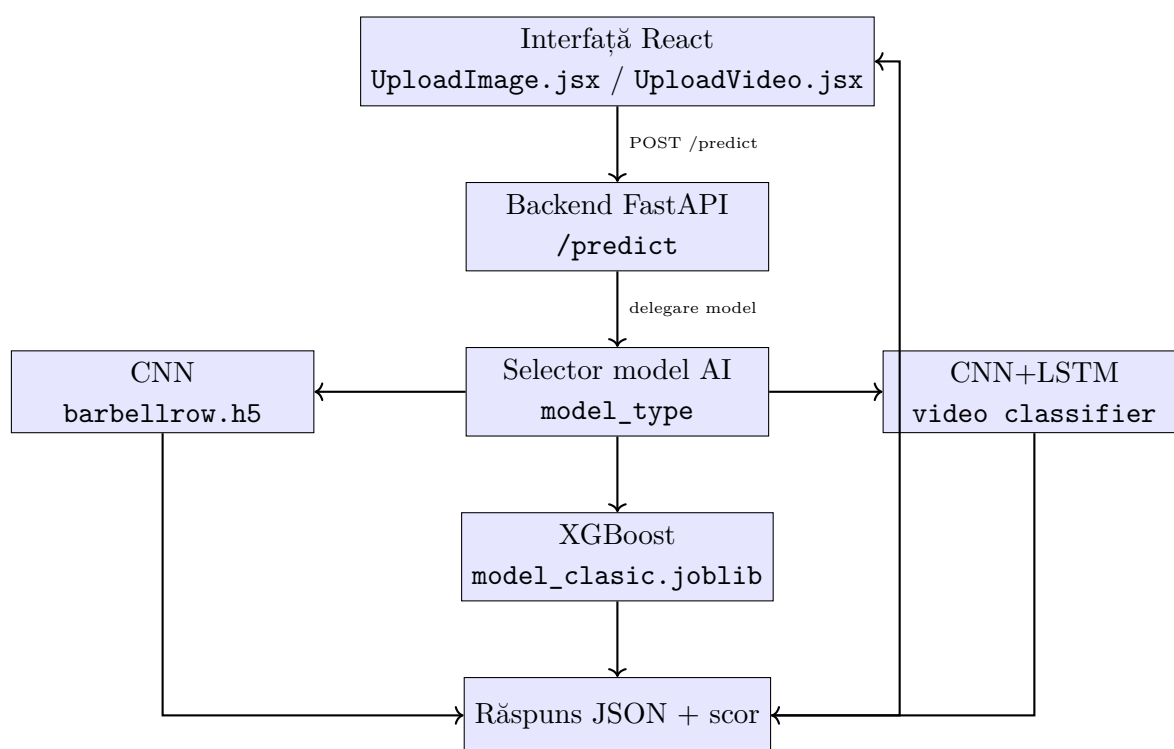


Figura 3.4: Fluxul backend – selecție și inferență model AI în *FitPose AI*

Prin urmare, backend-ul acționează ca un intermediar inteligent între interfața React și modelele de învățare automată, encapsulând logica de decizie și inferență într-o arhitectură flexibilă și scalabilă. FastAPI s-a dovedit a fi o alegere potrivită datorită performanței ridicate și a simplității în dezvoltare, iar integrarea sa cu bibliotecile TensorFlow, OpenCV, MediaPipe și XGBoost a permis o soluție completă și ușor de întreținut.

### 3.4 Manual de utilizare

Un manual de utilizare bine structurat este esențial pentru a facilita primele interacțiuni ale utilizatorului cu platforma *FitPose AI*. Această secțiune prezintă pașii necesari

pentru a utiliza aplicația și modul în care pot fi obținute rezultatele de evaluare a execuției pentru exercițiile fizice.

Pentru a utiliza aplicația, utilizatorul accesează interfața web locală la adresa *http://localhost:3000*, într-un browser precum Chrome sau Firefox. Pagina principală este afișată conform Figure 3.5, permițând selectarea dintr-un dropdown exercițiului ce urmează a fi analizat.

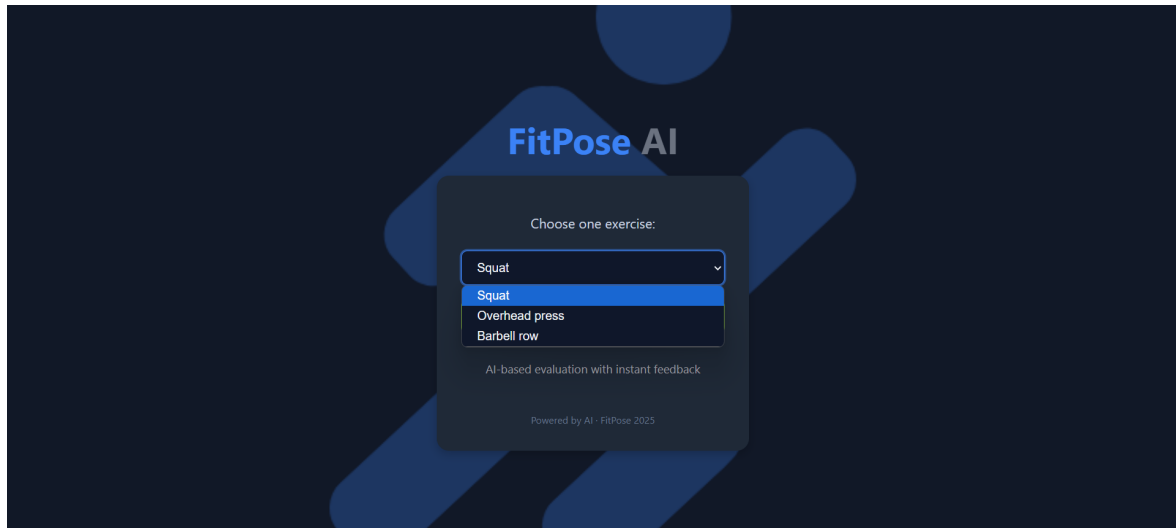


Figura 3.5: **Home Page** - dropdown pentru aplicația FitPose AI

După selectarea exercițiului, utilizatorul trebuie să confirme alegerea apăsând butonul *Load file*, care declanșează redirectionarea către pagina corespunzătoare pentru încărcarea fișierului.

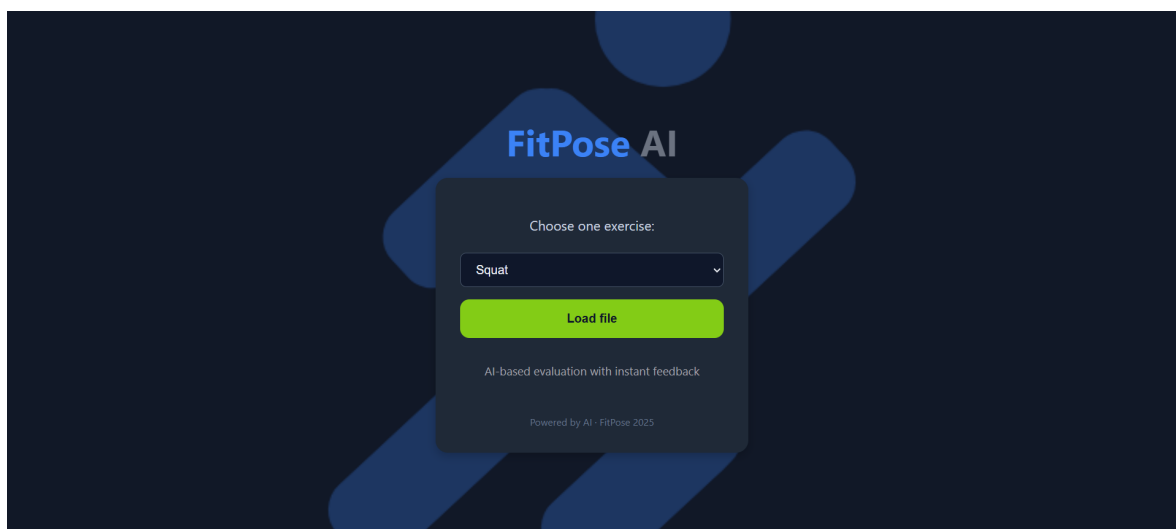


Figura 3.6: **Home Page** - butonul "Load File" pentru aplicația FitPose AI

După apăsarea butonului și alegerea exercițiului utilizatorul este redirectionat către pagina de încărcare a fișierului. În funcție de exercițiu, aplicația permite încărcarea fie a unei imagini statice, fie a unui videoclip.

Pentru exercițiul *Barbell Row*, aplicația oferă două opțiuni: analiza pe baza unui model CNN sau pe baza landmark-urilor extrase prin MediaPipe și procesate de un model XGBoost. Figura 3.7 prezintă interfața corespunzătoare.

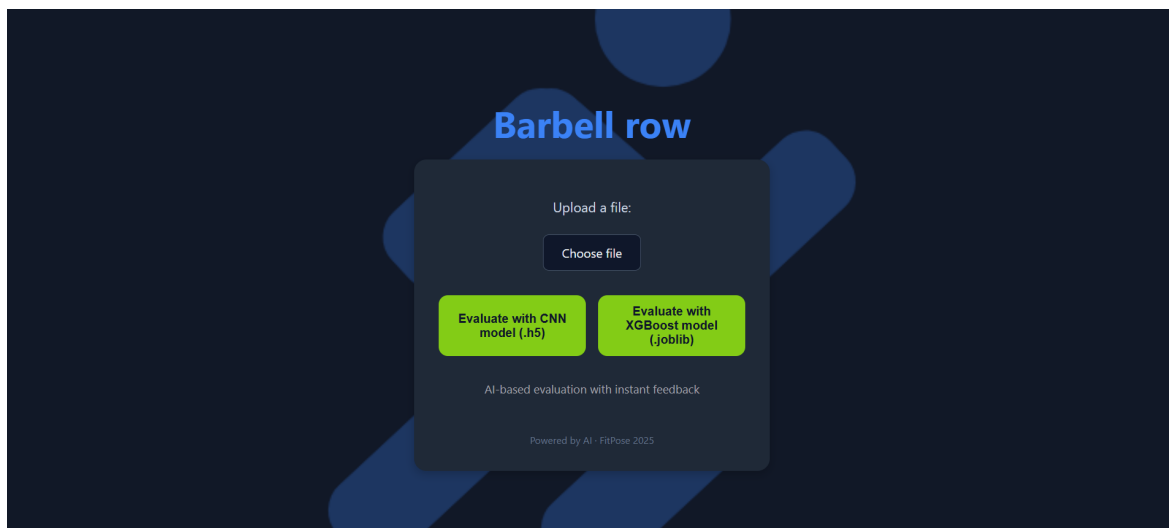


Figura 3.7: Încărcarea imaginii pentru exercițiul Barbell Row

După ce utilizatorul selectează fișierul dorit din sistemul local, imaginea este afișată automat în interfață sub forma unei previzualizări, așa cum se observă în Figura 3.8. Acest comportament permite utilizatorului să confirme că fișierul încărcat este cel corect.

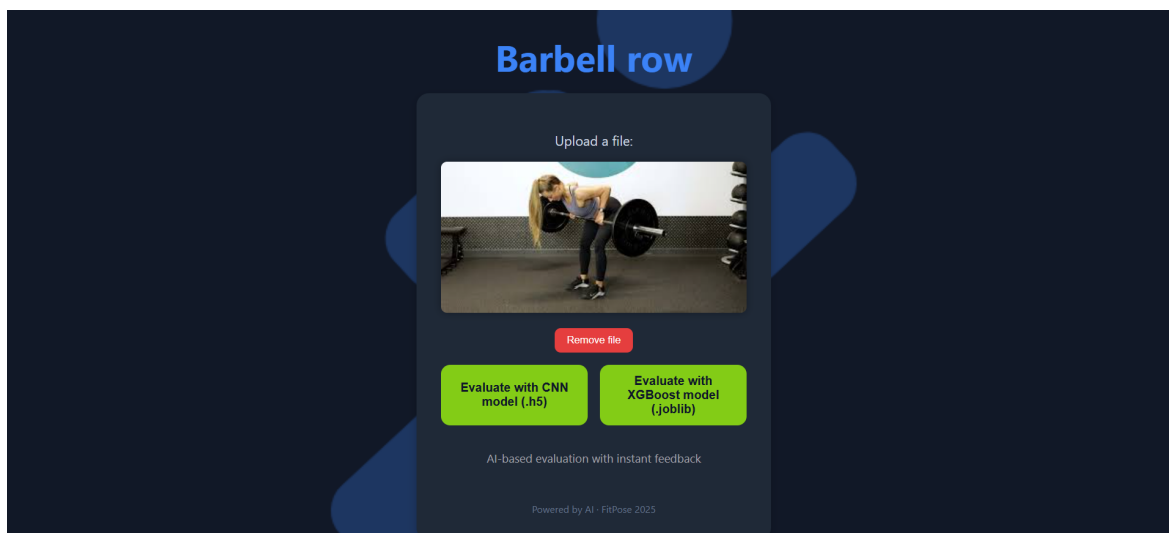


Figura 3.8: Încărcarea imaginii - previzualizare pentru exercițiul Barbell Row

Sub imagine apare un buton cu eticheta *Remove file*, care oferă posibilitatea de a șterge fișierul selectat în cazul unei alegeri gresite. După încărcarea fișierului, utilizatorul are la dispoziție două butoane pentru trimiterea imaginii către backend, corespunzătoare fiecăruia dintre cele două modele disponibile (CNN sau XGBoost). Fiecare acțiune trimite fișierul la server printr-o cerere POST, împreună cu metadatele relevante (*model\_type* și *exercise*), în vederea procesării.

Pentru exercițiile *Squat* și *Overhead Press*, utilizatorul trebuie să selecteze un fișier video (*.mp4*) corespunzător execuției exercițiului. După selectarea fișierului video din sistemul local, acesta este afișat direct în interfață, într-un player video HTML5, permițând utilizatorului să vizualizeze conținutul înainte de trimiterea pentru analiză. Comportamentul este ilustrat în Figura 3.9 și este similar cu cel pentru încărcarea imaginilor.

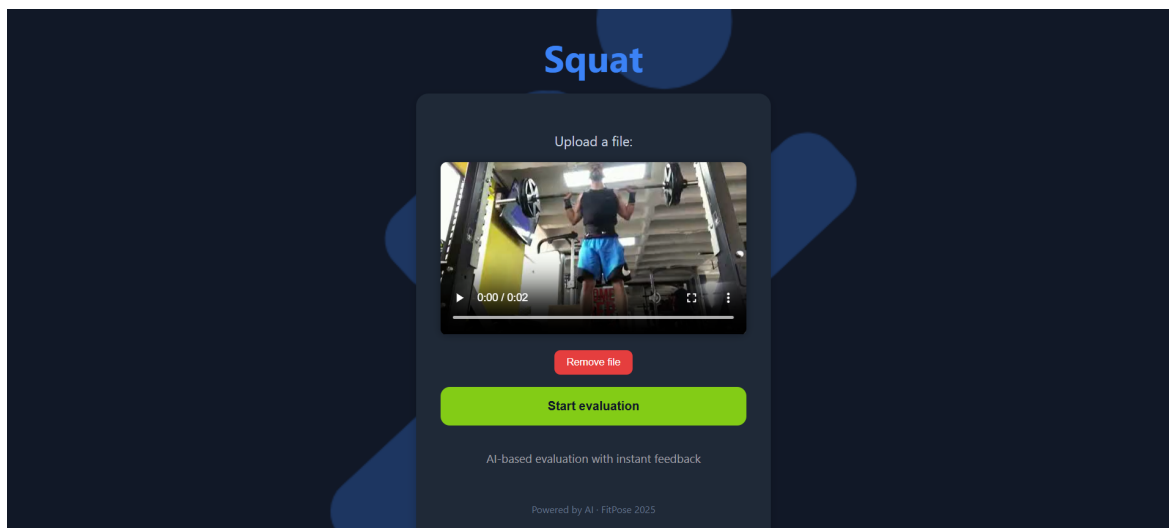


Figura 3.9: Încărcarea videoclipului pentru exercițiul Squat

După apăsarea butonului de analiză *Start evaluation*, aplicația trimite videoclipul către backend pentru procesare prin modelul CNN-LSTM, împreună cu informațiile despre exercițiul selectat. În cazul în care utilizatorul încearcă să inițieze analiza fără a fi selectat un fișier video, aplicația previne trimiterea cererii și declanșează un mesaj de alertă vizuală. Această alertă este generată prin metoda nativă *alert()* din JavaScript și se afișează în partea superioară a ecranului sub forma unei ferestre modale cu mesajul explicit „*Select a video file*”, așa cum este ilustrat în Figura 3.10.

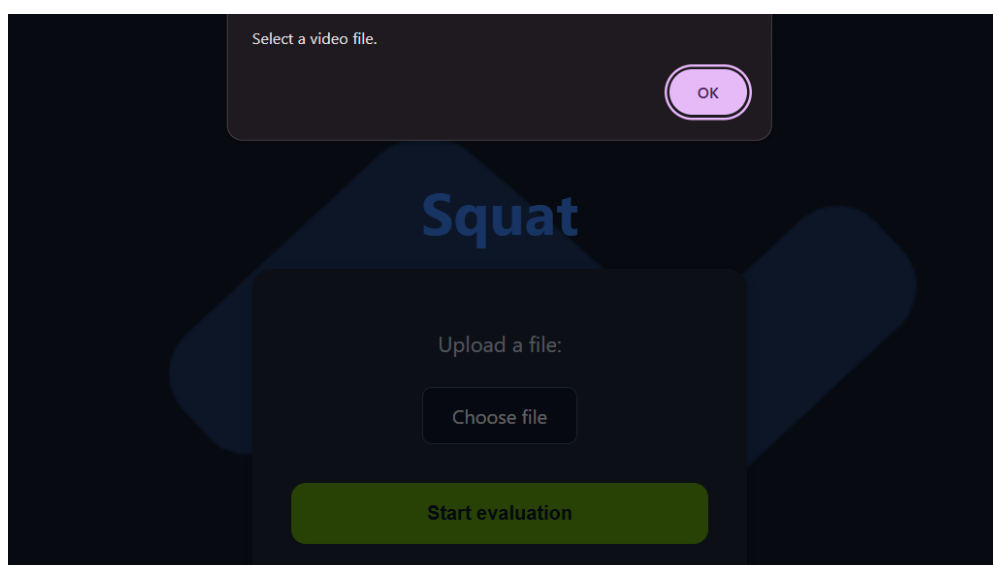
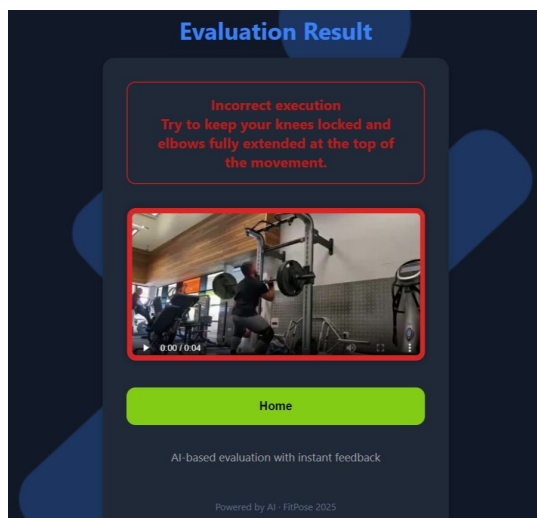


Figura 3.10: Încărcarea videoclipului - alertă pentru exercițiul Squat

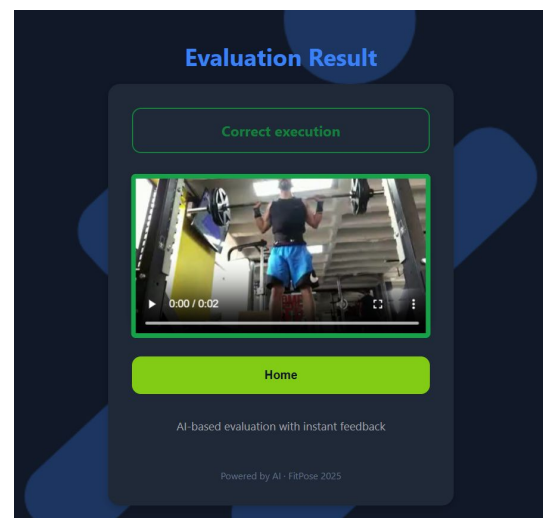
Comportamentul este identic și pentru secțiunea de imagini, unde aplicația refuză trimiterea către backend dacă nu este selectată o imagine validă, oferind utilizatorului aceeași alertă adaptată contextului. Acest mecanism de validare asigură robustețea aplicației și contribuie la prevenirea erorilor de la nivelul serverului, ghidând totodată utilizatorul în procesul de interacțiune.

După procesare, utilizatorul este redirecționat către pagina de feedback, unde este afișată eticheta generată de modelul AI, împreună cu un mesaj informativ despre calitatea execuției. În cazul unei predicții negative (execuție incorectă), mesajul include sugestii concrete de îmbunătățire, adaptate exercițiului analizat. Aceste sugestii sunt afișate într-un cadru de atenționare cu margine roșie, pentru a evidenția vizual verdictul nefavorabil. În schimb, atunci când modelul clasifică execuția ca fiind corectă, aplicația afișează un mesaj pozitiv de confirmare, fără sugestii de corectare. Caseta de feedback apare în acest caz evidențiată cu o margine verde, transmițând un răspuns vizual imediat, favorabil utilizatorului. Videoclipul analizat este redat similar, dar încadrat cu o bordură colorată, reflectând verdictul oferit de modelul AI. Această consistență vizuală între mesajul textual și culoarea de stare contribuie la o experiență de utilizare intuitivă și clară.

În partea de jos a interfeței, aplicația oferă un buton de întoarcere către pagina principală care încheie sesiunea de evaluare într-un mod coerent vizual.



(a) Feedback negativ



(b) Feedback pozitiv

Figura 3.11: Mesaje de feedback afișate în funcție de rezultatul predicției

### 3.5 Extinderi posibile

Deși aplicația *FitPose AI* îndeplinește obiectivele propuse și oferă o funcționalitate completă pentru evaluarea posturii prin imagini și videoclipuri, arhitectura actuală rămâne minimalistă, cu o implementare predominant procedurală și fără integrarea unei structuri de tip full-stack complet. Această abordare simplifică dezvoltarea și învățarea, dar limitează extensibilitatea pe termen lung.

Din perspectiva backend-ului, una dintre direcțiile principale de extindere o reprezintă integrarea unei baze de date relaționale (ex. PostgreSQL sau MySQL) pentru a permite gestionarea persistentă a utilizatorilor, sesiunilor de analiză și istoricului evaluărilor. Adăugarea unui sistem de autentificare și autorizare (bazat pe JWT sau OAuth2) ar permite personalizarea evaluărilor și accesul diferențiat la anumite funcționalități (ex. statistici avansate doar pentru utilizatorii înregistrați). De asemenea, backend-ul ar putea fi restructurat conform principiilor arhitecturii RESTful sau chiar migrat către o arhitectură cu microservicii, folosind containere Docker și orchestrare cu Kubernetes, pentru a asigura scalabilitate și modularitate.

În ceea ce privește frontend-ul, aplicația ar putea beneficia de o interfață mai interactivă și mai bogată vizual. O direcție importantă de îmbunătățire este includerea unui sistem de feedback vizual asupra imaginii sau videoclipului analizat, prin evidențierea directă a punctelor de interes (landmark-uri) și a zonelor cu deviații posturale. De asemenea, s-ar putea implementa un panou de control cu istoricul evaluărilor, filtrare pe exerciții și afișarea progresului în timp, sub formă de grafice sau tabele dinamice. Integrarea unui sistem de notificări și salvare locală temporară (prin LocalStorage sau IndexedDB) ar putea îmbunătăți experiența utilizatorului.

Din punct de vedere funcțional, aplicația ar putea fi extinsă pentru a susține un set mai variat de exerciții, inclusiv rutine personalizate de antrenament. Se poate adăuga o funcționalitate de *upload batch* (încărcare multiplă), precum și posibilitatea de a compara două execuții ale aceluiași exercițiu. O altă direcție interesantă este integrarea cu dispozitive mobile printr-o aplicație nativă (React Native sau Flutter), pentru evaluări rapide în sala de sport sau în context educațional.

În concluzie, deși implementarea actuală a fost aleasă pentru a permite testarea rapidă și eficientă a funcționalității modelelor implementate și pentru a permite o dezvoltare rapidă, aplicația oferă o bază solidă pentru extinderi ulterioare. Atingerea unui nivel de maturitate software mai ridicat presupune integrarea unor principii de arhitectură modulară, sisteme de management al utilizatorilor și funcționalități vizuale avansate, orientate spre o experiență de utilizare completă și personalizată.

# Concluzii

Lucrarea de față a urmărit dezvoltarea unui sistem inteligent denumit *FitPose AI*, destinat evaluării execuției exercițiilor fizice prin intermediul unor modele de învățare automată aplicate pe date vizuale. Într-un context marcat de digitalizarea tot mai accentuată a activităților de fitness și educație fizică, soluțiile automatizate de analiză a mișcării devin din ce în ce mai relevante, atât pentru utilizatorii individuali, cât și pentru profesioniștii din domeniul sportului sau al sănătății.

Contribuția majoră a acestei lucrări constă în integrarea și compararea a trei abordări diferite pentru evaluarea execuției: un model CNN pentru clasificarea imaginilor statice, un model hibrid CNN-LSTM pentru analiza videoclipurilor și un model XGBoost bazat pe trăsături numerice extrase cu MediaPipe Pose. Fiecare model a fost testat și validat pe seturi de date specifice, fiind antrenat cu grijă pentru a reflecta cât mai fidel diferențele dintre o execuție corectă și una incorectă. Rezultatele experimentale obținute confirmă aplicabilitatea fiecărei metode în scenarii diferite, oferind o bază solidă pentru extinderi viitoare.

Pe lângă partea de cercetare și dezvoltare a modelelor AI, o componentă esențială a proiectului a fost construirea aplicației web *FitPose AI*. Aceasta a fost realizată utilizând o arhitectură de tip client-server, cu un frontend dezvoltat în ReactJS și un backend minimalist în FastAPI. Deși structura aplicației este una modulară și procedurală, aceasta reușește să ofere o interfață accesibilă și funcțională pentru utilizator, permițând încărcarea de imagini sau videoclipuri și afișarea rapidă a unui scor de evaluare generat de modelul AI.

Concluzia generală care se desprinde din această lucrare este că un astfel de sistem poate oferi utilizatorilor un feedback obiectiv și rapid asupra execuției exercițiilor fizice, fără a necesita echipamente speciale sau intervenția unui expert. Platforma propusă reprezintă o demonstrație practică a modului în care inteligența artificială poate contribui la auto-corectarea și educarea posturală într-un cadru accesibil.

În viitor, aplicația ar putea fi extinsă prin includerea de noi exerciții, introducerea unui sistem de autentificare cu istoric de sesiuni și personalizarea feedback-ului în funcție de progresul utilizatorului. De asemenea, se poate urmări migrarea arhitecturii către o structură completă de tip MVC, integrarea cu baze de date relaționale sau chiar adoptarea unei arhitecturi bazate pe microservicii pentru scalabilitate crescută. Din punct de vedere AI, se pot explora abordări mai sofisticate precum transfer learning, regresie posturală multi-clasă sau rețele de tip transformer adaptate pe date biomecanice.

Realizarea acestei aplicații a reprezentat o oportunitate excelentă de a combina



concepte avansate din învățarea automată cu dezvoltarea de software orientată către utilizator. Proiectul final oferă un punct de plecare solid pentru viitoare cercetări și aplicații în domeniul computer vision pentru fitness și sănătate, fiind totodată un exemplu viabil de integrare practică a inteligenței artificiale în viața de zi cu zi.

# Bibliografie

- [BLG<sup>+</sup>20] Valentin Bazarevsky, Camillo Lugaresi, Matthias Grundmann, Awni Hannun, and Alexey Gorban. Mediapipe hands: On-device real-time hand tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1435–1444. IEEE, 2020.
- [Boy21] Nikhil Kumar Reddy Boyalla. Real-time exercise posture correction using human pose detection technique. Master’s Thesis, SUNY Polytechnic Institute, 2021. Accessed 2024-07-11.
- [Bud10] Emil Budescu. *Biomechanica*. Editura Universitatii “Dunarea de Jos” din Galati, 2010.
- [CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.
- [DAHG<sup>+</sup>15] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015.
- [Hay98] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1998.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [KKDT23] Hitesh Kotte, Miloš Kravčík, and Nghia Duong-Trung. Real-time posture correction in gym exercises: A computer vision-based approach for performance analysis, error classification and feedback. In *18th European Conference on Technology Enhanced Learning (EC-TEL)*. CEUR Workshop Proceedings, 2023.
- [KTB 22] Mislav Kasović, Tanja Trošt Bobić, and Lovro Štefan. The role of gluteus maximus in posture and movement. *Archives of kinesiology*, 2(1), 2022.
- [LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.
- [Pan22] Shaohong Pan. A method of key posture detection and motion recognition in sports based on deep learning. *Mobile Information Systems*, 2022:1–9, 2022.
- [PGR22] Paritosh Parmar, Amol Gharat, and Helge Rhodin. Domain knowledge-informed self-supervised representations for workout form assessment. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 105–123. Springer, 2022.
- [PIE21] A. Pavlova, V. Ivanov, and N. Egorova. Posture analysis and correction technologies: A literature review. *Procedia Computer Science*, 186:462–469, 2021.
- [SARG23] L Sheela, M Arpana, P Reddy, and Sudhagar Govindaswamy. A media-pipe blaze pose model to evaluate yoga posture with immediate feedback section a-research paper 3448 a mediapipe blaze pose model to evaluate yoga posture with immediate feedback. pages 3448–3452, 07 2023.
- [ZWWX21] Gang Zeng, Wen Wang, Qing Wang, and Xiaojie Xue. A survey on human motion analysis from depth data. *Pattern Recognition*, 113:107828, 2021.