### UNIVERSITATEA DIN BUCUREȘTI

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

# TITLUL LUCRĂRII DE LICENȚĂ

### Absolvent Numele studentului

**Coordonator științific**

**Titlul și numele profesorului coordonatorului**

**București, iunie 2021**

**Rezumat**

Statisticile miscarilor dintr-un meci de tenis pot oferi un mare avantaj in pregatirea unui jucator. O cunoastere detaliata a unui oponent poate fi benefica in pregatirea unui jucator, oferind un avantaj in intelegerea punctelor forte si a celor slabe ale unui adversar, ceea ce aduce un plus pentru desfasurarea antrenamentelor.

Pentru aceasta vom crea o aplicatie ce va oferi predictii asupra miscarilor unui jucator dintr-un videoclip folosind limbajul python si biblioteca PyTorch pentru a implementa modelele.

**Abstract**

The statistics of the movements in a tennis match can offer a great advantage in the preparation of a player. A detailes knowledge of an opponent can be beneficial in the preparation of a player, offering an advantage in understanding the strengths and weakness of opponent, which brings a plus for training.

For this we will create an application that will provide predictions on the movements of a player from a video using python and the PyTorch library to implement the models.

Contents

[UNIVERSITATEA DIN BUCUREȘTI 1](#_Toc132726652)

[TITLUL LUCRĂRII DE LICENȚĂ 1](#_Toc132726653)

[Absolvent Numele studentului 1](#_Toc132726654)

[1. Introducere 4](#_Toc132726655)

[1.1. Motivatie 4](#_Toc132726656)

[1.2 Structura lucrarii 4](#_Toc132726657)

[1.2.1: Detectia jucatorului 5](#_Toc132726658)

[1.2.2: Predictia actiunii 5](#_Toc132726659)

[2. Concepte teoretice 5](#_Toc132726660)

[2.1: LSTM 5](#_Toc132726661)

[2.2 YOLO 6](#_Toc132726662)

[2.3 YOLO pose 7](#_Toc132726663)

[3. Abordari recente 7](#_Toc132726664)

[4. Tehnologii folosite 8](#_Toc132726665)

[4.1 Python 8](#_Toc132726666)

[4.2 PyTorch 8](#_Toc132726667)

[4.3 Matplotlib 9](#_Toc132726668)

[4.4 OpenCV 9](#_Toc132726669)

[4.5 YOLOv7 9](#_Toc132726670)

[5. Bibliografie 10](#_Toc132726671)

# Introducere

### 1.1. Motivatie

Analiza video a sporturilor este o solutie pentru a extrage diferite informatii care pot fi folosite de profesionisti cat si de telespectatori pentru a avea o privire de ansamblu analitica a ceea ce se intampla pe parcursul jocului.

O astfel de tehnologie este sistemul „Hawk-Eye Live” care poate semnala daca mingea a fost data afara suprafetei de joc. Insa instalarea acestui sistem este dificila si costisitoare, conform [Espn, 2023] costul pentru instalarea acestui sistem pe un teren este intre $60.000 si $70.000, facand folosirea acesteia imposibila pentru antrenamente sau pentru turneele mici.

Scopul acestei lucrari este de a oferi o solutie accesibila din punct de vedere computational, fiind usor de rulat pe un GPU.

Lucrarea propune o modalitate de a vedea ce miscari precise efectueaza un jucator de tenis in cadrul unei filmari, pentru a putea fi analizata. Prin intermediul acestei abordari se poate analiza atent jocul jucatorului, antrenorii putand imbunatati strategiile, tehnica de joc, precum si pregatirea fizica inaintea meciurilor.

Abordarea aleasa va folosi limbajul pyhon pentru implementare si tehnici inovatoare pentru a detecta jucatorul si coordonatele pozitiilor membrelor. Astfel pentru detectia jucatorului vom folosi YOLOv7 care conform [Wang et al., 2022] intrece toti detectorii in privinta vitezei si a acuratetei, oferind o acuratete de 56.8% pe setul de date MS COCO. Pentru obtinerea coordonatelor membrelor vom folosi un detector de pose estimation antrenat pe YOLOv7 avand la baza tehnica prezentata in [Maji et al., 2022] care ofera o acuratete buna, de 68.5 pe setul de date COCO 2017 si are o performanta computationala foarte buna.

### 1.2 Structura lucrarii

* **Detectia jucatorului**: Selectarea jucatorului care are dimensinile cele mai mari din fiecare cadru, extragerea coordonatelor membrelor si normalizarea datelor pentru a putea fi prelucrate.
* **Predictia actiunii**: Antrenarea, proiectarea si inferenta intr-un videoclip bazandu-se pe coordonatele membrelor pentru a oferi predictia miscarii efectuate, pe un segment de video, avand numarul de date prestabilit ca hiperparametru.

### 1.2.1: Detectia jucatorului

Pentru a detecta jucatorul vom folosi modelul YOLOv7 preantrenat pentru detectia pozitiilor membrelor, pentru a detecta toate persoanele dintr-o imagine. Dupa aceasta detectie vom selecta numai persoana care are dimensiunea cea mai mare din cadru (daca ar fi feed-ul unei camere din transmisia TV, acest model va selecta numai jucatorul din partea de jos). Pentru fiecare membru din „cutia” dominanta ca si dimensiuni, coordonatele acestuia se vor normaliza in intervalul [0, 1]. Aceasta operatiune se va repeta pentru fiecare cadru dintr-o filmare

### 1.2.2: Predictia actiunii

Vom lua datele de la punctul anterior, ne vom folosi de un set de date adnotat cu videoclipuri cu lovituri din tenis pentru a antrena un model care sa poata da o predictie asupra miscarii efectuate de catre un jucator, oferind ca iesire una din 12 miscari, elementare jocului de tenis.

# Concepte teoretice

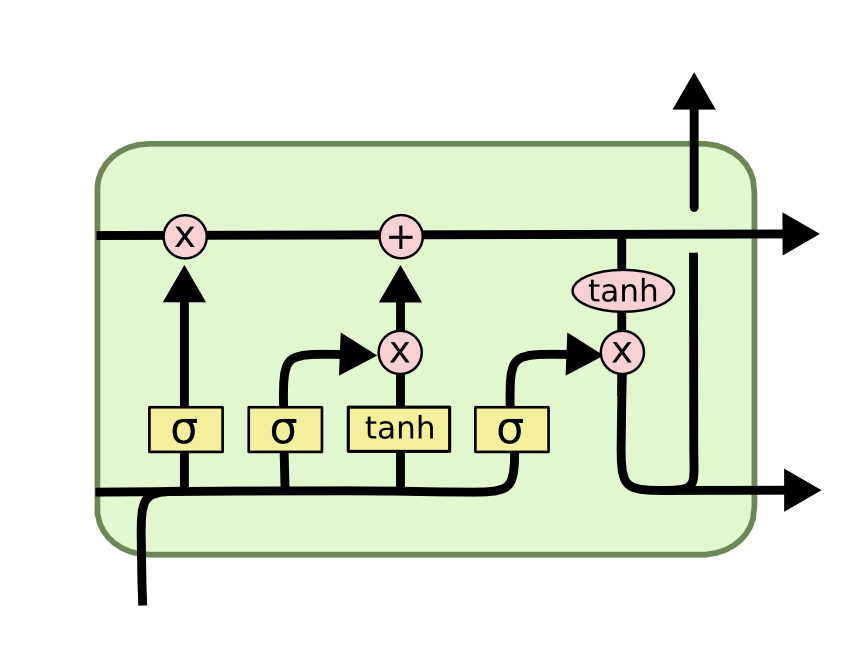
### 2.1: LSTM

Retelele de tip LSTM (Long short term memory) vin ca o evolutie a retelelor RNN (Recurrent neural networks), care conform [Staudemeyer et al., 2019.] nu pot depasi mai mult de 5-10 pasi de timp intrucat gradientii se pot micsora foarte tare (dispar) sau pot avea valori foarte mari (explodeaza). In cazul in care gradientii explodeaza, valoarea „w” oscileaza, iar in cazul in care gradientii dispar, antrenarea dureaza foarte mult timp sau chiar nu functioneaza.

Solutia acestei probleme este folosirea LSTM care poate avea conform [Staudemeyer et al., 2019.] si mai mult de 1000 de pasi de timp. Retelele LSTM exceleaza pe sarcinile care necesita ca un set de date secvential sa fie tinut minte pentru o perioada lunga de timp. Aceasta propietate este atribuita blocurilor de memorie, care pot accesa intrarea si produc o iesire, prevenind astfel ca informatiile irelevante sa intre sau sa iasa din acest bloc. De asemenea aceste blocuri mai au si o poarta de „uitare” astfel incat daca o caracteristica devine irelevanta, aceasta nu mai este luata in calcul [Staudemeyer et al., 2019.].

Aceste retele se pot suprapune pentru a forma o retea deep LSTM, astfel iesirea unei astfel de retele, reprezinta intrarea urmatoarei. Aceasta abordare a adus o imbunatatire dramatica fata de o retea cu un singur strat, in domeniul transcrierii mesajelor audio in text [Graves et al., 2013].

Pentru a accelera antrenarea putem folosi „Gradient clipping” care mentine gradientii intr-un interval stabilit ca si hiperparametru. De exemplu daca am setat valoarea de 0.25 ca hiperparametru si gradientul nostru are valoarea de 0.7, acesta va fi setat la valoarea de 0.25. Conform [Zhang et al., 2019] folosind aceasta tehnica, au testat pe o sarcina de prelucrarea limbajului natural, folosind modelul AWD-LSTM, au putut observa o crestere a vitezei cu care modelul este antrenat.



Imagine descriptiva a unui bloc LSTM [colah.github.io, 2023]

### 2.2 YOLO

YOLO (You only look once) este o retea capabila sa detecteze locatiile obiectelor in timp real din imagini. In aceasta lucrare vom folosi YOLOv7 care imbunatateste acuratetea si viteza. Pentru antreneare acesta foloseste setul de date MS COCO, antrenarea facandu-se de la 0, nefolosindu-se alte seturi de date sau retele preantrenate. Comparativ cu alte metode, de exemplu YOLOv4, acest model are cu 75% mai putini parametrii si aduce o imbunatatire a preciziei medii de 1.5%. Comparativ cu modelul state-of-the-art YOLOR-CSP, autorii ne prezinta ca modelul are cu YOLOv7 are cu 43% mai putini parametrii si o precizie medie cu 0.4% mai mare [Wang et al., 2022].

### 2.3 YOLO pose

Sarcina acestui model este de a detecta coordonatele articulatiilor tuturor persoanelor dintr-p imagine. Pentru antrenarea acestui model, autorii au folosit setul de date COCO Keypoint Detection, unde au augumentat datele folosind redimensionare, translatare, inversare, augumentari mozaic si diferite auggumentari ale culorilor. Pe setul de date COCO val2017 precizia medie este de 68.5% iar cel mai bun model este HigherHRNet cu 70.5%. Conform autorilor la rezolutii mai mici, YOLOv5s6-pose are o performanta mult mai buna decat alte modele de complexitate mica, de exemplu EfficientHRNet, pe setul de date COCO val2017 [Maji et al., 2022].

# Abordari recente

O abordare propusa in estimarea actiunii este folosindu-ne de caracteristici calculate de mana precum in lucrarea [Wang et al., 2013] care se foloseste de caracteristici calculate de mana pentru a estima miscarea. Pentru aceasta, autorii calculeaza homografia folosind algoritmul RANSAC pentru a stabiliza miscarea camerei, intre 2 cadre. Dupa aceea se folosesc de un detector uman pentru a detecta oamenii din imagine, folosindu-se de acele detectii pentru a imbunatati estimarea optical flow-ului eliminand oamenii care pot avea o alta miscare fata de cea a camerei, astfel stricand estimarea miscarii camerei. In urmatorul pas sunt calculati descriptorii HOF si MBH optical flow, iar descriptorul HOG sunt aplicati pe imaginea originala. Pentru clasificare autorii se folosesc de un SVM liniar pentru a face predictiile, datele fiind codificate intr-un vector Fisher.

La momentul aparitiei aceasta abordare s-a dovedit a fi cea mai buna pe seturile de dateHMDB51 (57.2% acuratete), UCF50 (91.2% acuratete).

O alta abordare este cea [Varol et al., 2017] care se foloseste de retele convolutionale adanci pentru a estima actiunea. Ca si date de intrare autorii experimenteaza cu diferite tehnici, de exemplu de a oferi imaginile ca RGB sau ca un optical flow. Rezultatele cele mai bune provin din folosirea metodei Brox de a obtine optical flow-ul, aceasta fiind furnizata retelei. Alte experimente au fost efectuate in gasirea numarului optim de cadre care sa fie furnizat retelei.

Cand a aparut rezultatele pe seturile de date UCF101 (92.7% acuratete) si pe HMDB1 (67.2% acuratete) au fost cele mai bune.

O alta abordare este facuta in [Luvizon et al., 2018] care pentru o imagine se foloseste de Inception-V4 pentru a extrage caracteristicile vizuale precum si estimarea locatiilor articulatiilor. Aceste 2 rezultate sunt agregate pentru a furniza predictia finala. Autorii folosesc aceste 2 informatii deoarece unele actiuni pot fi dificil de interpretat folosind doar una dintre metode. Ei dau ca exemplu actiunea de a bea apa si actiunea de a da un telefon care poate sa fie dificil de interpretat doar din locatiile articulatiilor. Ei antreneaza reteaua folosind 16 cadre continue, obtinand cele mai bune rezultate pe setul de date Penn Action for 2D action recognition (98.6% acuratete).

# Tehnologii folosite

### Python

Pentru realizarea aplicatiei am folosit limbajul Python care ofera o flexibilitate mare in privinta alegerii unei biblioteci de deep learning, oferind cele mai populare 2 optiuni: TensorFlow si Pytorch. De asemenea ofera biblioteca Open-CV care este de mare ajutor pentru extrage cadrele din videoclipuri.

### PyTorch

Pytorch este o biblioteca de deep learming care se ocupa de operatii cu tensori si cu o diferentiere automata, utilizandu-se de accelerarea placii grafice. Principiile de proiectare sunt de a fi cat mai usor de utilizat pentru un dezvoltator familiarizat cu limbajul Python, sa puna oamenii din cercetare pe primul loc, oferind un mijloc usor de a dezvolta modele, optimizatoare si un mijloc de incarcare al datelor usor de folosit. Complexitatea operatiunilor din model trebuie sa fie ascunsa de dezvoltator, el trebuind sa foloseasca doar API-uri intuitive. De asemenea trebuie sa aiba o performanta pragrmatica, implementarea acceptand putina complexitate in plus pentru a oferi o experienta mai intutiva. Aceasta biblioteca este scrisa in Python cu cod optimizat de C++ [Paszke et al., 2018]. Aceasta biblioteca a fost folosita pentru a construi modelul de LSTM folosit pentru recunoasterea actiunii.

### 4.3 Matplotlib

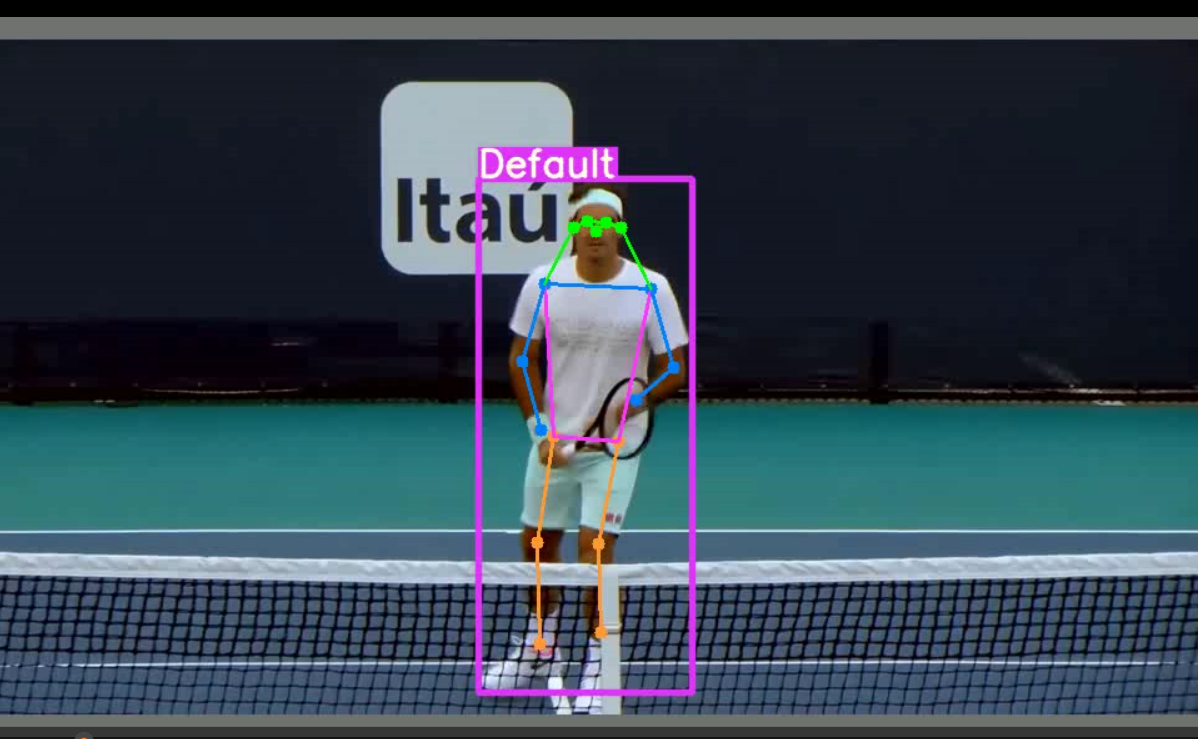
Matplotlib este o biblioteca de Python care are rolul de a crea grafice statice, animate si interactive. In cadrul proiectului aceasta a fost folosita pentru a afisa metricele de performanta, precum acuratetea modelului, si functiile de pierdere.

### 4.4 OpenCV

Este o biblioteca open-source care are o functionalitate potrivita pentru sarcinile de vedere artificiala in timp real. Este scrisa in principal in C++ si este oferita pentru Python, Java si MATLAB/Octave. Incepand cu anul 2010 aceasta biblioteca ofera accelerare de pe placa grafica. In cadrul acestui proiect aceasta biblioteca a fost folosita pentru a citi videoclipurile cadru cu cadru pentru a putea fi prelucrate de modelul YOLO.

### 4.5 YOLOv7

Pentru a prelua informatiile despre coordonatele persoanelor, respectiv a coordonatelor articulatiilor m-am folosit de implementarea oficiala a [Wang et al., 2022] care se poate regasi la [github.com/yolov7, 2023]. Din implementarile oferite am folosit modelul preantrenat pe gasirea coordonatelor articulatiilor, folosindu-ma de functionaliteatea oferita de biblioteca. De exemplu o functie folosita a fost cea care imi translateaza coordonatele detectiei de la formatul in care se ofera coordonatele detectiei, aceasta fiind oferita in locatia x si y a punctului din mijloc, in lungimea si inaltimea detectiei in pixeli, in cea care reprezinta punctul din stanga sus si cel din dreapta jos.



Exemplu de output al modelului YOLO din cadrul proiectului

# Bibliografie

[Espn, 2023] URL: <https://www.espn.com/tennis/story/_/id/30877297/hawk-eye-live-gains-more-support-australian-open> (accesat la data 17.04.2023)

[Wang et al., 2022] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *arXiv preprint arXiv:2207.02696* (2022).

[Maji et al., 2022] Maji, Debapriya, Soyeb Nagori, Manu Mathew, and Deepak Poddar. "YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2637-2646. 2022.

[Staudemeyer et al., 2019.] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks." *arXiv preprint arXiv:1909.09586* (2019).

[Graves et al., 2013] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645-6649. Ieee, 2013.

[Zhang et al., 2019] Zhang, Jingzhao, Tianxing He, Suvrit Sra, and Ali Jadbabaie. "Why gradient clipping accelerates training: A theoretical justification for adaptivity." *arXiv preprint arXiv:1905.11881* (2019).

[colah.github.io, 2023] URL: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accesat la data 17.04.2023)

[Wang et al., 2013] Wang, Heng, and Cordelia Schmid. "Action recognition with improved trajectories." In *Proceedings of the IEEE international conference on computer vision*, pp. 3551-3558. 2013.

[Varol et al., 2017] Varol, G., Laptev, I. and Schmid, C., 2017. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, *40*(6), pp.1510-1517.

[Luvizon et al., 2018] Luvizon, Diogo C., David Picard, and Hedi Tabia. "2d/3d pose estimation and action recognition using multitask deep learning." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5137-5146. 2018.

[Paszke et al., 2018] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen et al. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems* 32 (2019).

[wikipedia.com, 2023] URL: <https://en.wikipedia.org/wiki/OpenCV> (accesat la data 18.04.2023)

[github.com/yolov7, 2023] URL: <https://github.com/WongKinYiu/yolov7> (accesat la data 18.04.2023)