Universitatea Națională de Știință și Tehnologie POLITEHNICA București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Detecția și identificarea Gesturilor mâinilor

Proiect de diplomă

prezentat ca cerință parțială pentru obținerea titlului de

Inginer în domeniul *Electronică şi Telecomunicaţii*

programul de studii de licență *Tehnologii şi Sisteme de Telecomunicaţii*

Conducător(i) științific(i)
Conf. dr. ing. Ionuț Pîrnog

Absolvent
Nelepcu Gabriel

2025

Declarație de onestitate academică

Prin prezență declar că lucrarea cu titlul " *Detecţia şi identificarea gesturilor mâinilor* ", prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității Naționale de Știință și Tehnologie POLITEHNICA București ca cerință parțială pentru obținerea titlului de *Inginer* în domeniul *Electronică şi Telecomunicaţii,* programul de studii *Tehnologii şi Sisteme de Telecomunicaţii* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurătorilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, *data*

Absolvent *Nelepcu Gabriel*

_____

(semnătura în original)

Copyright © *anul* , *nume student* / *nume companie*

Toate drepturile rezervate

Autorul acordă UPB dreptul de a reproduce și de a distribui public copii pe hîrtie sau electronice ale acestei lucrări, în formă integrală sau parțială.

Cuprins

Lista abrevieri
HCI:Human-Computer-Interactions
CNN:Convolutional Neural Network
ML:Machine Learning
AR/VR: Augmented Reality, Virtual Reality

Introducere

Motivația alegerii temei

Tema lucrării a fost aleasă din dorința de a explora o aplicație practică a inteligenței artificiale și viziunii computerizate, cu impact direct în interacțiunea om-calculator. Recunoașterea gesturilor mâinilor oferă un mod intuitiv și natural de control al dispozitivelor, fiind utilizată tot mai frecvent în domenii precum automatizări, realitate virtuală sau asistență pentru persoane cu dizabilități. Detectarea gesturilor este considerate o aplicație foarte importantă în dezvoltarea tehnologiei, în industria Huma-Computer-Interactions(HCI). Acest lucru le oferă computerelor posibilitatea de a identifica și executa comenzi fară să fie fizic atinse.

Prin acest proiect, se urmărește dezvoltarea unui sistem capabil să detecteze și să recunoască gesturi în timp real, combinând metode moderne de procesare video cu modele de învățare automată.

În plus, dezvoltarea unui astfel de sistem presupune utilizarea unor metode complexe de prelucrare video, detectare și urmărire a mâinilor, precum și clasificarea secvențelor de mișcare folosind rețele neuronale. Se urmărește nu doar implementarea unei aplicații funcționale, ci și aprofundarea cunoștințelor teoretice și practice în domeniul învățării automate și procesării imaginilor, într-un context modern și util.

Gradul de noutate al temei

Recunoașterea gesturilor mâinii este un domeniu în continuă dezvoltare, cu aplicații semnificative în interacțiunea om-calculator, realitate augmentată, sisteme de control fără mâini și accesibilitate pentru persoanele cu dizabilități. Totuși, multe soluții existente necesită echipamente hardware specializate, precum senzori de adâncime sau camere cu infraroșu, limitând astfel accesibilitatea și aplicabilitatea lor în contexte de utilizare generală.

Proiectul propus adresează această problemă prin dezvoltarea unui sistem software care folosește ca și limbaj de programare Python. În cadrul acestui limbaj, avem mai multe librării special create pentru analiza și recunoașterea mâinilor. Acestea sunt MediaPipe, creată de Google. Ea este open-source, Google dorind să ofere posibilitatea ca oricine să poată contribui la ea. O altă bine-cunoscută librărie este OpenCV. Această librărie a fost dezvoltată în cadrul Intel, de către Gray Bradsky. A fost lansată în anul 2002, devenind și aceasta open-source.

Un aspect inovator al lucrării este integrarea unei funcționalități care permite reantrenarea modelului direct din interfața utilizatorului, fără a necesita intervenția unui specialist. Astfel, utilizatorii pot adăuga noi gesturi și le pot asocia cu acțiuni specifice, extinzând în mod constant repertoriul de gesturi recunoscute, ceea ce face sistemul mai flexibil și adaptabil nevoilor fiecărui utilizator.

În plus, implementarea unui mecanism prin care gesturile necunoscute sunt stocate automat pentru reantrenare adaugă o componentă de învățare incrementală, permițând sistemului să se îmbunătățească treptat pe măsură ce sunt adăugate noi gesturi. Aceasta este o funcționalitate relativ rar întâlnită în soluțiile existente de recunoaștere a gesturilor.

Obiective Generale

1. Dezvoltarea unui sistem de recunoaștere a gesturilor mâinii în timp rea**l**: Crearea unei aplicații software care să utilizeze o cameră web standard pentru a detecta și identifica gesturile mâinilor. Acest sistem va permite interacțiunea cu dispozitivele sau aplicațiile prin gesturi, fără a necesita echipamente hardware specializate.

2. **I**mplementarea unui model de învățare automată pentru recunoașterea gesturilor statice și dinamice: Antrenarea unui model ML (Machine Learning) capabil să recunoască atât gesturi simple (de exemplu, „thumbs up” sau „peace”), cât și gesturi mai complexe care implică mișcări ale mâinii.

3. Crearea unei funcționalități de reantrenare dinamică a modelului: Permite utilizatorilor să adauge noi gesturi și să le asocieze cu acțiuni specifice, fără a necesita intervenția unui specialist, extinzând astfel constant repertoriul de gesturi recunoscute.

4. Stocarea și reantrenarea gesturilor necunoscute: Implementarea unui sistem care să stocheze automat gesturile necunoscute pentru reantrenare, facilitând astfel învățarea incrementală a modelului pe măsură ce sunt adăugate noi gesturi.

5. Integrarea unui sistem de control al acțiunilor: Permite legarea gesturilor recunoscute cu acțiuni specifice, cum ar fi controlul volumului, deschiderea unui meniu sau derularea unei prezentări, în scopul de a demonstra aplicabilitatea tehnologiei în viața cotidiană.

Metodologia Folosită

1. Colectarea și Preprocesarea Datelor:

-Colectarea unui set de imagini și videoclipuri pentru a antrena și valida modelul de recunoaștere a gesturilor. Aceste date vor proveni din surse publice și vor fi completate cu gesturi noi, colectate de utilizatori prin intermediul aplicației.

-Preprocesarea acestora va include normalizarea dimensiunii, aplicarea unor tehnici de augmentare a datelor pentru a asigura diversitatea setului de date, și conversia imaginilor în formate care pot fi procesate eficient de modele de învățare automată.

2. Dezvoltarea Modelului de Recunoaștere a Gesturilor:

   -Implementarea unui model de învățare automată utilizând tehnici de viziune computerizată (de exemplu, utilizarea unui model CNN (Convolutional Neural Network)pentru extragerea caracteristicilor din imagini).Antrenarea modelului folosind seturi de date etichetate pentru a recunoaște gesturi statice și dinamice. Se vor folosi librării de machine learning precum PyTorch sau TensorFlow.

3. Implementarea Funcționalității de Reantrenare din Interfața Utilizatorului:

   -Dezvoltarea unei interfețe grafice sau terminal-based, prin care utilizatorii pot introduce noi gesturi și le pot asocia cu acțiuni.

   -Crearea unui sistem care stochează gesturile necunoscute și le trimite la model pentru reantrenare automată.

4. Validarea Modelului și Testarea Performanței:

   -Testarea modelului pe seturi de date de validare pentru a evalua performanța acestuia în condiții variate de iluminare și unghiuri de vizualizare.

   -Optimizarea performanței prin ajustarea hiperparametrilor și a algoritmilor de învățare automată.

   -Evaluarea acurateței sistemului prin metrice precum acuratețea, precizia, recall-ul și scorul F1.

5. Integrarea cu Acțiuni Specifice:

   -Implementarea unui sistem de mapping al gesturilor recunoscute către acțiuni de sistem

   -Testarea funcționalității de control al acțiunilor pentru a evalua eficiența și ușurința utilizării.

1. Context Teoretic

1.1 Recunoașterea Gesturilor Mâinilor

Recunoașterea gesturilor este o topică foarte importantă în știința calculatoarelor, deoarece se dezvoltă tehnologii care fac mai usoară interacțiunea obișnuită a oamenilor cu apartele, fără să fie nevoie de nicio atingere. Întregul proces de recunoaștere,procesare și transformarea lor în comenzi este cunoscută ca și recunoașterea gesturilor [1].[1]

1.2 Librăria MediaPipe

În ziua de astăzi, există multe librării folosite în ML(Machine Learning). Una din ele este MediaPipe. Această librărie a fost concepută pentru a implementa modele de învățare pregătite, care să construiască fluxuri de procesare pentru a realiza inferența unor date senzoriale arbitrare [2]. În MediaPipe, componentele modular provin dintr-un flux de procesare de percepție, împreună cu funcția de inferență, modelul de procesare media și transformările de date [3]. Grafurile de operații sunt utilizate și în alte librării de învățare automată, precum TensorFlow [4], MXNet [5], Pytorch [6] și OpenCv 4.0 [7].

Recunoașterea gesturilor mâinilor a fost cercetată de către Zhang Fahn, folosind o cameră simplă RGB pentru aplicații AR/VR în sisteme în timp real, care puteau prezice mâna scheletului uman [8].

[1] [1]: Z.Xu, et.al, Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors, In Processing og IUI'09, 2009, pp 401-406.

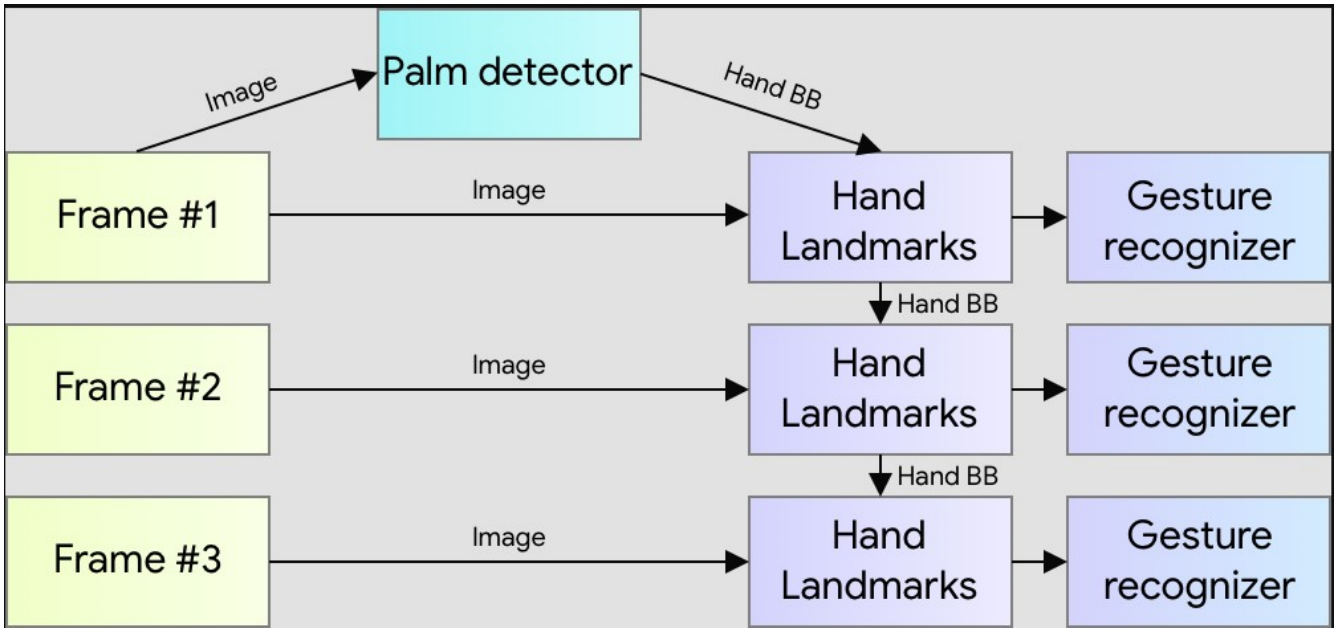

Figura 1.1 Fluxul percepției mâinii

În figura 1.1 este ilustrat fluxul prin care un gest este recunoscut. Acesta este constituit din 3 pași:
-un model de deteție al mâinii care procesează imaginea capturată de cameră și o returnează decupată în jurul mâinii.
-un model de recunoaștere a punctelor de referință ale imaginii decupate și returnează punctele 3D cheie ale mâinii
-un identificator de gesture care clasifică punctele și le configurează într-un set discret de gesture

1.2.1 Modelul de detecție a mâinii

Librăria MediaPipe are integrat un detector inițal al mâinii numit BlazePalm. Detecția mâinii este o sarcină complicate,primul pas fiind antrenarea unui model ce detectează palma utilizatorului, in loc de toată mâna. Următorul pas este folosirea algoritmului de suprimare non-maximă. Aici se folosesc casete delimitatoare pătrate pentru a evita rapoarte de aspect nadorite și de a reduce numărul de ancore cu un factor de 3-5. În continuare, se utilizează o arhitectură encoder-decoder pentru extracția

trăsăturilor,folosită la scară largă,dar care merge și pe obiecte mici. În final se minimizează pierderile focale apărute în timpul antrenării modelului pentru a susține un număr mare de ancore rezultate de la variația mare de scală.

1.2.2 Modelul de identificare a punctelor de referință ale imaginii

Figura 1.2 Reperele mâinii Sursa[9]

Modelul obține o localizare precisă a 21 puncte cheie.Această localizare se realizează în interiorul regiunilor mâinii detectate, printr-un model de regresie care generează direct coordonatele punctelor, reprezentând astfel modelul de identificare a punctelor de referință.
Fiecare articulație a mâinii are coordonate formate din x, y și z, unde x și y sunt normalizate în intervalul [0.0, 1.0] în funcție de lățimea și înălțimea imaginii, iar z reprezintă adâncimea punctului de referință. Adâncimea este raportată față de încheietura mâinii, considerată punctul de referință inițial. Cu cât un punct se află mai aproape de cameră, cu atât valoarea coordonatei z este mai mică.

1.2.3 Detectarea mâinii

Pentru recunoașterea gesturilor mâinii, implementarea unui algoritm simplu constă în calcularea gesturilor cu un unghi cumulat determinat al stării articulației sau condiționarea fiecărui deget, cum ar fi degetul îndoit sau degetul drept.
Pentru un exemplu de recunoaștere avem gesturi precum “salut”,”piatra” în următoarele imagini.

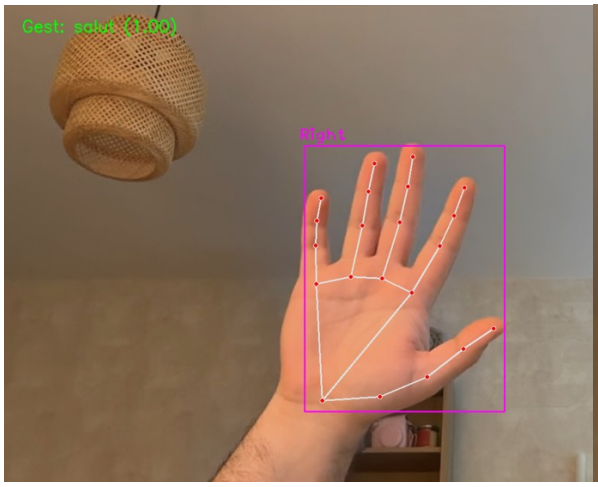

Figura 1.3

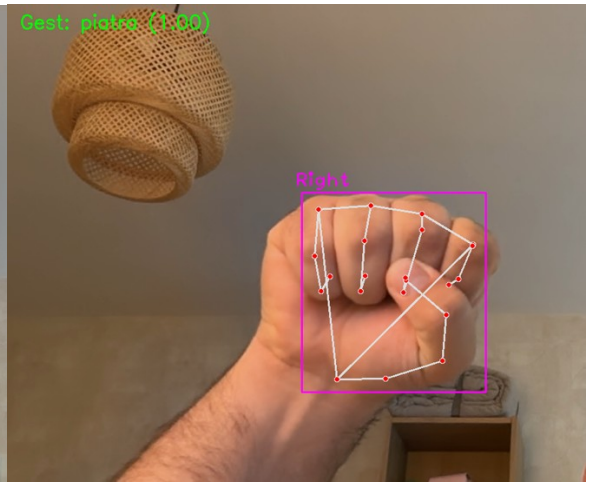

Figura 1.4

2. Proiectarea Programului

2.1 Descriere Generală

Aplicația software are scopul de a detecta și recunoaște diferite semne ale mâinilor, folosind puncte cheie în cadrul librăriei MediaPipe. Apoi aceste gesturi sunt salvate și clasificate cu ajutorul numelui introdus de către utilizator din interfață pentru a putea antrena modelul. În final, după ce modelul a fost antrenat, utilizatorul poate rula programul de detecție pentru ca modelul să recunoască gesturile pe care a fost antrenat. Aplicația este construită pe mai multe module pentru a putea fi extinsă cu noi gesturi și funcționalități.

2.2 Obiective funcționale

Principalele funcționalități sunt:

- Capturarea de cadre prin intermediul camerei video
- Detectarea mâinilor și extragerea punctelor cheie
- Preprocesarea coordonatelor punctelor cheie
- Clasificarea gesturilor
- Salvarea datelor
- Oferirea unei interfețe în CLI (linie de comandă) prin care utilizatorul poate alege din mai multe opțiuni
  -Detectare gest
  -Antrenare Model

2.3Arhitectura generală

Aplicația a fost organizată pe mai multe module, care lucrează împreună pentru a realiza fluxul complet de recunoaștere și învățare. Acestea sunt:

- Modulul de captură video
  Acesta folosește librăria OpenCv pentru a accesa camera dispozitivului și obține cadrele necesare

- Modulul de detecție
  Pentru detecție și extragerea punctelor cheie, este folosit MediaPipe

- Modelul de preprocesare
  Structurează și normalizează datele, pentru o recunoaștere mai bună și precisă

- Modulul de clasificare
  Acesta trimite datele la modelul ce trebuie antrenat și returnează eticheta gestului antrenat

- Modulul de interfață al utilizatorului
  Oferă opțiuni de antrenare sau detectare

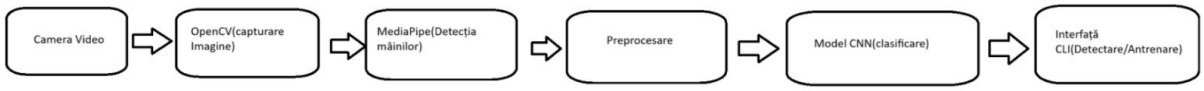

Figura 2.1 Diagrama Arhitecturii

2.4 Justificarea alegerii arhitecturii aplicației

Structura arhitecturală aleasă este una modular,menită să asigure atât claritatea logică a sistemului, cât și posibilitatea extinderii ulterioare. Alegerea componentelor s-a bazat pe cerințele aplicației, urmărind obținerea unei soluții eficiente.

Separarea clară a responsabilităților

Fiecare etapă este tratată de un modul distinct, ceea ce asigură:
-Testabilitate crescută: fiecare modul poate fi verificat separat.

-Extensibilitate: este ușor de înlocuit o componentă fără a afecta restul sistemului (ex: schimbarea MediaPipe cu un alt detector de puncte-cheie).
-Debbuging simplificat: posibilele erori pot fi localizate rapid în cadrul modular.

Alegerea MediaPipe pentru extragerea punctelor-cheie

MediaPipe este un framework optimizat pentru procesarea video în timp real, care oferă o detecție precisă a punctelor mâinii fără a necesita un GPU dedicat. Alegerea oferă un echilibru între:
-Performanță
-Acuratețe (poziționare consistentă a punctelor pe articulații)
-Integrare ușoară cu limbajul Python și biblioteca OpenCV

În comparație cu alte metode MediaPipe este mult mai ușor de configurat și oferă o viteză superioară în contexte embedded sau cu resurse limitate.

Utilizarea unei rețele neuronale convoluționale (CNN)

Modelul CNN este antrenat pe secvențe de puncte-cheie (nu pe imagini brute), ceea ce reduce semnificativ complexitatea și dimensiunea datelor de intrare. Avantajele acestei decizii includ:
-Reducerea dimensiunii de intrare
-Generalizare mai bună: modelul învață variații naturale ale gesturilor în timp

3. Implementarea aplicației

3.1 Organizarea proiectului

Proiectul a fost implementat în Python, versiunea 3.12. Această versiune a fost aleasă datorită faptului că este una dintre cele mai noi versiuni și permite folosirea unor multitudini de biblioteci.
Proiectul a fost structurat pe mai multe fișiere și directoare:
- main.py – fișierul principal unde vor fi apelate funcțiile

-data_collect.py – conține clasa care gestionează fluxul de detecție a gesturilor.

-model.py – include logica de colectare date și antrenare model.

-model/ – director unde sunt salvate modelele CNN antrenate.

-dataset/ – conține datele gesturilor în format .npy sau .json.

-action_handler.py – se ocupă de gestionarea acțiunilor ce trebuie executate

3.2 Etapele implementării

Captura și prelucrarea datelor

Folosind biblioteca OpenCV, aplicația capturează cadre video de la camera integrată. Fiecare cadru este transmis către MediaPipe pentru a obține pozițiile celor 21 de puncte ale mâinii. Datele sunt convertite într-un vector de trăsături care reflectă forma și poziția mâinii.

Formarea secvențelor

Pentru recunoașterea gesturilor dinamice, aplicația colectează secvențe de N cadre (de exemplu, 30), formând un "clip" care este transmis către modelul CNN. Aceste secvențe sunt necesare pentru ca modelul să poată recunoaște gesturi bazate pe mișcare, nu doar pe poziție statică, în cazul semnelor dinamice. Pentru cele statice, modelul poate să fie antrenat doar pe coordonatele punctelor.

Clasificarea cu CNN

Modelul CNN primește o secvență de puncte și returnează o distribuție de probabilitate peste clasele cunoscute de gesturi. Gestul cu probabilitatea cea mai mare este considerat ca fiind cel detectat.

Executarea acțiunilor cu
action_handler.py

După identificarea unui gest, numele atribuit lui este trimis către action_handler.py, care mapează gestul la o acțiune specifică. Acțiunile pot include:

- controlul volumului audio;
- executarea unui script extern;
- afișarea unui mesaj;
- declanșarea unei funcționalități definite de utilizator.

Această separare permite extinderea mai ușoară aplicației fără a modifica logica de detecție.

3.3 Moduri de rulare

Aplicația este concepută pentru a funcționa în două moduri, selectabile din linia de comandă:

- Modul de detecție
- Modul de antrenare

Dacă se alege modul de detceție, aplicația va deschide direct camera și va începe detecția și recunoașterea semnelor realizate de utilizator.

Dacă este ales modul de antrenare, se va deschide un alt meniu, unde utilizatorul poate alege dintre colectare de date noi pentru un gest nou ce se dorește a fi introdus, antrenare
pe datele colectate la prima opțiune sau înapoi pentru a se îmtoarce la meniul initial.

3.4 Salvarea datelor

După colecatre și antrenare modelul este salvat in format .pth folosind Pytorch. La pornirea aplicației, modelul este recunoscut și încărcat automat. Bazele de gesture sunt salvate în directorul dataset, în format .npy.

3.5 Detectarea gesturilor

Modelul analizează gestul efectuat, iar în funcție de probabilitatea sa, va afișa numele acestuia. Pentru o detecție bună, se recomandă antrenarea modelului pe mai multe seturi de date, în diferite condiții de iluminare. În cazul în care directorul unui gest nu are destule date, modelul va analiza și va returna numele gestului celui mai apropiat, cu proabilitatea cea mai mare.

4. Testare și Validare

5. Concluzii

6.Bibliografie

[1]: Z.Xu, et.al, Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors, In Processing og IUI'09, 2009, pp 401-406.

[2]: Lugaresi C, Tang J, Nash H, McClanahan C, et al. MediaPipe: A Framework for Building Perception Pipelines. Google Research. 2019. https://arxiv.org/abs/2006.10214 "accesat la data de 5 aprilie 2025"

[3]:Lugaresi C, Tang J, Nash H et.al, MediaPipe: A Framework for Perceiving and Processing Reality. Google Research. 2019.

[4]: Abadi M, Barham P, Chen J et.al, Tensorflow: A System for Large-Scale Machine Learning, In 12th USENIX Symposium on Operating System Design and Implementation (OSDI), USA, 2016, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi"accesat la data de 5 aprilie 2025"

[5]:Chen T, Li M, Li Y, MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed System, 2015, https://arxiv.org/pdf/1512.01274.pdf "accesat la data de 6 aprilie 2025"

[6]: Pazke A, Gross A, Chintala S, Automatic Differentiation in PyTorch, In 31st Conference on Neural Information Processing System (NIPS), USA, 2017.

[7]: Matveev D, OpenCV Graph API. Intel Corporation. 2018.

[8]: Zhag F, Bazarevsky, Vakunov A et.al, MediaPipe Hands: On – Device Real Time Hand Tracking, Google Research. USA. 2020. https://arxiv.org/pdf/2006.10214.pdf "accesat la data de 6 aprilie 2025"

[9]: https://mediapipe.readthedocs.io/en/latest/solutions/hands.html "accesat la data de 9 aprilie"

7. **Anexe**

7.1 Cod sursă

7.2 Capturi de ecran