



Recunoașterea persoanelor după mers

Bolotă Iustin
Țoșu Cosmina Diana

1. Context & Motivație

- **Context:** Mersul este o caracteristică biometrică importantă, iar identificarea persoanelor după acesta este substanțială în securitate, prevenirea crimelor și supraveghere video.
- **Motivație:** Spre deosebire de alte date biometrice, mersul poate fi înregistrat de la distanță și fără cooperarea subiectului (ca în cazul faceID-ului, ampretei). Soluțiile eficiente pot duce la acuratețe mare, chiar dacă există covariații.
- **Obiectivul proiectului:** Scopul proiectului este să dezvolte o metodă de recunoaștere a persoanelor prin mers, folosind un model deep-learning pe seturi de siluete, capabil să identifice inclusiv în condiții nefavorabile.

Gait Recognition for Identity Verification



Prelucrarea Imaginilor - Proiect, 2025

2. Arhitectura preliminară a soluției

- Schema arhitecturii:



•Descrierea componentelor:

1. **DATASETS** – conțin datele brute, respectiv siluete alb-negru pentru mai multe persoane în diferite ipostaze. Datasetul folosit este CASIA-B.
2. **PRETREATMENT** (*pretreatment.py*) - traversează imaginile brute din dataset și realizează preprocesarea lor. Mai exact: elimină cadre invalide cu prea puțini pixeli albi, taie marginile goale sus/jos pentru a păstra doar corpul, centrează silueta pe axa orizontală, redimensionează la o dimensiune standard 64x64, adaugă padding dacă este nevoie și salvează imaginile pentru antrenare.
3. **DATA LOADER** (*data_loader.py*, *data_set.py*, *sampler.py*) - transformă cadrele procesate în structuri de date utilizabile de modelul de învățare. Fiecare set de imagini este încărcat într-un obiect de tip set (care nu este o secvență ordonată; modelul tratează mersul ca un set de forme, nu o serie temporală), după care are loc alinierea la nivel de frame , reprezentarea numerică (convertire în matrici) și TripletSampler (formează batch-uri de forma [Anchor, Positive, Negative], unde Anchor și Positive sunt imagini cu aceeași persoană, iar Negative cu o alta).

4. **FEATURE EXTRACTION – SETNET** (*network/gaitset.py + basic_blocks.py*) - extrage trăsăturile din seturile de imagini. *SetNet* este o rețea convoluțională care primește seturi de cadre și extrage *embedding*-uri – vectori pentru reprezentarea unică a persoanei. Sunt incluse: *Convolutional Blocks* (filtre CNN care învață formele locale din siluetă), *Pooling global/local* – combină informația din toate cadrele și *HPM* (Hierarchical Pyramid Mapping) (strat care combină informațiile la niveluri diferite de detaliu (global și local) pentru o reprezentare mai robustă). Rezultatul este un vector de trăsături pentru mersul unei persoane.

5. **TRIPLER LOSS** (*network/triplet.py*) - învață să facă diferența între persoane prin compararea distanțelor între *embedding*-uri: (Anchor-Positive) trebuie să fie apropiate, (Anchor-Negative) trebuie să fie depărtate. Folosește *hard mining* pentru a alege cele mai dificile exemple din batch. Se optimizează astfel spațiul vectorial al trăsăturilor.

6. **MODEL TRAINING** (*model.py, initialization.py, train.py*) - optimizează parametrii rețelei pentru a minimiza Triplet Loss. Se inițializează modelul și datele, se rulează procesul de antrenare folosind *Adam optimizer*, se calculează pierderea și se actualizează greutatea CNN-ului. Periodic, modelul este salvat sub formă de checkpoint.

7. **EVALUATION** (*test.py, evaluator.py*) – încarcă modelul antrenat și evaluează performanța acestuia pe setul de test. Se extrag vectorii de trăsături, se calculează distanțele între ei și se determină precizia modelului pentru diferite unghiuri și condiții (mers normal, cu bagaj, haine diferite).

3. Evaluarea Preliminară a Soluției

- **Metodologia de evaluare:** Metodologia de testare a soluției a fost în modul Galerie-Test, unde șabloanele de mers sunt comparate cu probele utilizând distanța euclidiană; performanța este măsurată prin Acuratețea Rank-1 (procentul de identificări corecte obținute pe prima poziție).
- **Setul de date:** Am ales setul de date CASIA-B pentru magnitudinea acestuia și pentru faptul că include date capturate în condiții diverse.
- **Exemple de cazuri de test:** Modelul a fost testat pe subiecți cu mersul normal, subiecți mergând cu o geantă în mână și subiecți cu articole vestimentare proeminente.

4. Rezultate Preliminare

- **Rezultate obținute:** Acuratețe medie: 30.44% identificare corectă la prima încercare
- **Interpretarea rezultatelor:** Rezultatele obținute reprezintă acuratețea identificării persoanelor în diverse condiții (mers normal, ducând o geantă, îmbrăcați cu o haină) și cu unghiuri de filmare într 0 și 180 de grade.

	Unghi Aprox. (Grade)	NM (Normal)	BG (Geantă)	CL (Haină)
	0° (Față)	29.40%	25.10%	11.00%
	18°	42.10%	29.50%	11.50%
	36°	45.30%	33.60%	10.20%
	54°	47.20%	34.24%	8.60%
	72°	56.90%	39.00%	11.60%
	90° (Lateral)	52.40%	38.90%	9.20%
	108°	54.00%	42.30%	12.80%
	126°	55.70%	42.30%	13.80%
	144°	48.70%	32.80%	13.50%
	162°	45.80%	30.91%	12.90%
	180° (Spate)	29.70%	21.60%	11.90%
Media Agregată	—	46.11%	33.66%	11.55%

5. Concluzii Preliminare

- **Rezumatul progresului:** Modelul selectat a fost antrenat și testat cu date din setul CASIA-B, utilizându-se 73 de subiecți pentru antrenare și 51 pentru testare.
- **Limitările soluției actuale:** Modelul utilizat necesită o memorie VRAM dedicată cu o capacitate relativ mare pentru a se reduce durata de timp a antrenării. De asemenea, pentru a se evita un timp excesiv de mare pentru antrenare ar fi recomandată rularea pe un sistem cu multiple GPU-uri.
- **Potențiale îmbunătățiri:** Procesul de antrenare poate parcurge mai multe iterații pentru a crește acuratețea soluției.

6. Direcții Viitoare

- **Pași următori:** Creșterea numărului de iterații pentru antrenarea modelului și compararea performanței cu alte metode pentru o evaluare robustă.
- **Plan de implementare:** Optimizarea parametrilor, ajustarea rețelei pentru o mai bună generalizare.
- **Obiectivele finale:** Obținerea unei acurateți RANK-1 cât mai mare și demonstrarea funcționării complete a fluxului - de la date brute la recunoaștere.