

Cuprins

Introducere	2
Contributii	3
1 Elemente de biometrie	4
1.1 Securitatea biometrică	4
1.1.1 Concepte	4
1.1.2 Recunoașterea facială	5
1.1.3 Recunoașterea bazată pe amprentă	8
1.3 Urechea, element biometric	8
1.2.1 Structura	9
1.2.2 Avantaje și dezavantaje	10
1.2.3 Studii anterioare	11
2 Tehnologii utilizate	13
2.1 Python	13
2.2 Open-CV	14
2.3 Scikit-learn	16
3 Implementare	17
3.1 Setul de date	17
3.2 Metode	19
3.2.1 Eigenears	19
3.2.2 HuMoments	24
3.2.3 Local Binary Pattern	27
4 Concluzii	32
5 Bibliografie	33

Introducere

Oamenii și tehnologia avansată au cunoscut un drum comun cu puțin timp în urmă, având în vedere vârsta omenirii. Această explozie cibernetică a luat naștere în secolul XX, iar acum, după zeci de ani, își face simțită prezența din ce în ce mai puternic printre obiceiurile noastre. Vedem cum lucrurile care cândva erau considerate problemele principale ale existenței, sunt rezolvate prin intermediul tehnologiei și par din ce în ce mai triviale pentru consumatori.

Ne-am propus să avem în vedere două aspecte de care oamenii mereu au fost îngrijorați, și anume, siguranța și securitatea personală. Ne dorim să ducem o viață cât mai lipsită de griji, iar controlul cât se poate de ridicat asupra bunurilor noastre, poate fi un factor ce ne influențează calitatea traiului. Fiind într-o perioadă de tranziție a secolului vitezei , putem observa din ce în ce mai puternic dorința populației de a ține pasul cu lumea virtuală. Aceasta impune o mare migrare a atenției dinspre bunurile fizice către cele electronice și anume, datele personale. După cum oamenii au început să conștientizeze , datele personale sunt unele dintre cele mai de valoare resurse ale societății curente. Dorința noastră de a ne simplifica problemele și de a susține o stare de comoditate, ne poate pune în pericol integritatea și din păcate, efectul este unidirecțional. Datele, odată accesate, pot fi utilizate în folosul altora sau mai rău, împotriva noastră.

Elementele biometrice bine cunoscute precum fizionomia feței și amprenta sunt deja utilizate de oameni dar, există în continuare cazuri în care eficiența lor poate scădea drastic. Ne-am propus să aducem în atenția studiilor, posibila capacitate pe care o poate avea urechea în vederea

recunoașterii de persoane. Implementarea sa poate aduce beneficii practice în arii precum securitatea publică și securitatea personală. Identificarea persoanelor din profil, unde datele celorlalte elemente biometrice nu pot fi exploatate, poate aduce contribuții majore în detectarea oamenilor ce nu respectă legi sau sunt urmăriți de către autorități. De asemenea, urechea s-a dovedit a fi diferită la persoane gemene identice fizionomic. O aplicație bazată pe acest element biometric ar putea fi capabilă să diferențieze aceste persoane prin imagini sau clipuri video.

Lucrarea prezintă câteva elemente de biometrie generală pentru a putea înțelege cum funcționează un asemenea sistem biometric împreună cu o serie de detalii despre potențialul urechii în această direcție. Mai departe, am prezentat tehnologiile utilizate în aplicație, alături de implementarea metodelor și modalitatea de organizare. În final, am sesizat o serie de concluzii ca rezultat al observațiilor din urma implementării.

Contribuții

Studiile în direcția recunoașterii bazate pe biometria urechii sunt încă proaspete, astfel încât, oferă o arie foarte largă de explorare a capacității sale. Am pornit de la ideea ce susțin că urechea, ca element biometric, are potențial care tinde să se apropie din ce în ce mai mult către celelalte elemente deja utilizate în acest context.

Principala strategie ce ne marchează contribuția este implementarea metodelor de recunoaștere cunoscute pentru succesul lor în alte tipuri de recunoașteri, de data aceasta având ca sursă de informație - urechea. Bineînțeles, acest lucru a trebuit urmat de o adaptare a parametrilor generali și a acestor metode întocmai pentru a reuși să obținem rezultate în funcție de acest nou element.

1 Elemente de biometrie

1.1 Securitatea biometrică

1.1.1 Concepte

Biometria este o metodă automată de a recunoaște persoane, bazată pe caracteristici fizice sau comportamentale. În trecut, indivizii erau distinși în principal, prin caracteristici evidente precum părți ale corpului, înălțime, culoarea ochilor. Epoca modernă, în schimb, se axează pe unicitatea trăsăturilor și are în vedere tehnici precum recunoașterea în funcție de față, amprentă, voce, iris, cea mai recentă componentă fiind urechea care poate influența într-o direcție pozitivă procesele.

Un sistem biometric poate fi separat în două tipuri:

- sistem de identificare: trăsăturile biometrice vor fi folosite pentru a determina identitatea unui individ dintr-o mulțime de oameni chiar și fără aprobarea acestuia
- sistem de verificare: trăsăturile biometrice sunt utilizate pentru a garanta acces individului (ex. ATM-uri bancare, clădiri securizate)

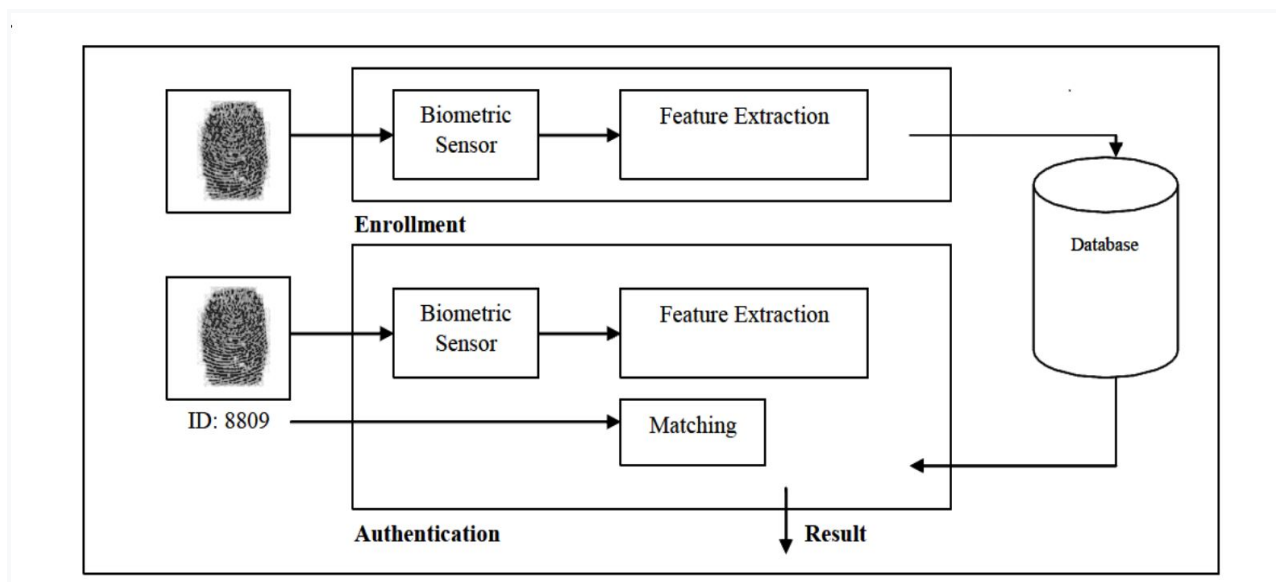


Fig. 1 Schema de bază a sistemului biometric având ca trăsătură amprenta

1.1.2 Recunoașterea facială

Unul dintre cele mai vechi și utilizate exemple de elemente, folosit pentru recunoașterea de oameni este chipul. Încă de la începutul civilizației, oamenii au folosit fețe pentru a identifica indivizi familiari și necunoscuți. Această sarcină simplă a devenit din ce în ce mai dificilă pe măsură ce sporul natural a crescut și pe măsură ce posibilitățile de schimbare a înfățișării au devenit mult mai comune.

Recunoașterea facială a luat amploare în partea a 2-a a secolului anterior, unde diverși oameni de știință au început investigarea computerizării ei. Rezultatele inițiale au existat, însă nu au cunoscut popularitate din cauza investitorilor care nu doreau să le facă cunoscute. Problema a fost în continuare studiată, având totuși parte de mici publicații ce au stârnit interesul societății asupra acestui subiect.

Tehnica principală prin care un sistem de recunoaștere este realizat, înglobează doi pași: extragerea trăsăturilor și clasificarea obiectelor. Pe parcursul anilor, diverse tehnici de diferite complexități au fost introduse în această zonă:

- Tradițional: algoritmi ce identifică trăsăturile faciale, extrăgând repere sau caracteristici dintr-o imagine a feței subiectului; exemplu care poate fi un algoritm ce poate analiza poziția, dimensiunea și/sau forma relativă a ochilor, nasului, etc; rezultatele sunt ulterior utilizate pentru a căuta alte imagini cu caracteristici similare;
- 3-Dimensional: aceasta utilizează senzori 3D pentru a capta informații despre forma feței; informația este utilizată pentru a identifica caracteristici distinctive pe suprafața feței precum conturul globilor oculari, al nasului, al bărbiei. Etc;
- Analiza structurii pielii: o tehnică ce folosește detaliile vizuale ale pielii, acestea fiind capturate digital; aceasta transformă liniile, modelele și petele unice din pielea unei persoane într-un spațiu matematic;
- Combinarea a două sau mai multe tehnici: fiecare metodă poate avea avantaje sau dezavantaje dar, din fericire, acestea pot fi combinate pentru a crea sisteme de recunoaștere cu rate de succes mult mai mari;

Analiza biometriei feței cunoaște de asemenea avantaje, dar și dezavantaje. Este o tehnică de bază, dar slăbiciunile și problemele pot fi cu ușurință evidențiate.

Principalul factor benefic este recunoașterea în masa pe care o poate realiza, indiferent de mărimea mulțimii indivizilor. Instalarea sistemelor de recunoaștere facială în locurile publice poate identifica în diferite scopuri trecătorii de la distanțe semnificative, fără ca aceștia să realizeze.

Dezavantajele apar masiv atunci când recunoașterea depinde preponderent de unghiul din care se realizează. Studiile au arătat că rezultatele pot varia de la excelente către slabe în momentul în care imaginea este rotită în lateral cu mai mult de 20 de grade. De asemenea, camerele cu rezoluție mică dau un randament foarte slab, fiind necesară utilizarea celor cu o calitate ridicată de preluare a imaginilor.

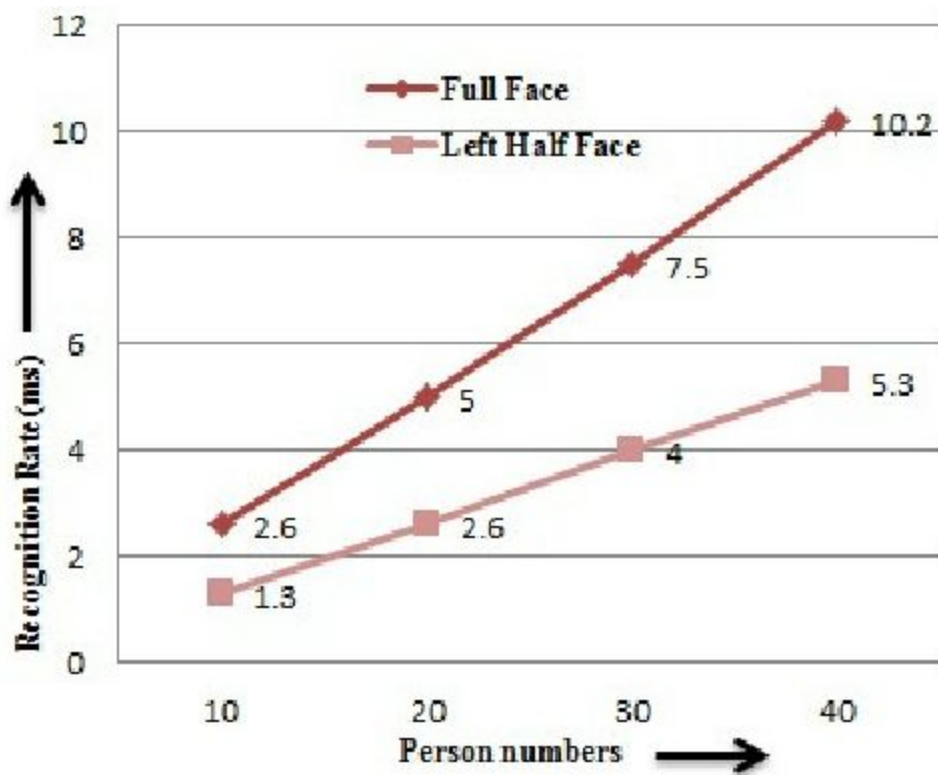


Fig. 2 Graficul a 40 de persoane identificate având în vedere poziționarea feței individului

1.1.3 Recunoașterea bazată pe amprentă

Istoria recunoașterii în funcție de amprentă este un subiect destul de vechi care a fost dezbătut, evident într-un mod mult mai rudimentar, încă din secolul al XIX-lea din dorința liderilor mondiali de a ține evidența oamenilor, nu doar printr-un nume, ci printr-un atribut fizic. Astfel, au găsit soluții și pentru diverse acțiuni considerate severe din punct de vedere social, fiind nevoie de înregistrarea vinovaților în scopul evitării incidentelor asemănătoare. În momentul de față, amprenta este utilizată ca material biometric în diferite domenii care necesită un nivel sporit de securitate: sectorul guvernamental, sectorul companiilor, sectorul bancar.

În viața de zi cu zi ne bazăm din ce în ce mai mult pe parole sau chei care sunt foarte vulnerabile la a fi pierdute sau furate. Având în vedere faptul că amprentele sunt unice pentru fiecare individ și nu pot fi pierdute din motive evidente, face această tehnică să devină una de înaltă acuratețe și încredere. Pe lângă avantajul rezultatelor optime, costul integrării este redus, fapt ce a ușurat accesul întregii populații.

În ciuda succesului său, tehnica poate fi ușor bruiată de diferiți factori externi ce pot reduce cu mult ratele de succes. Printre acestea se numără: umiditatea, răni în zona respectivă, neîncadrarea exactă a modului de testare.

1.2 Urechea, element biometric

Imaginile urechilor în sistemele biometrice automate pot fi colectate cu ușurință prin fotografierea din profil a persoanelor sau capturate dintr-un conținut video. Acest lucru se poate realiza fără contact direct și de asemenea, nu depinde de nivelul de cooperare al persoanei în proces de identificare.

1.2.1 Structura

Urechea umană își începe formarea încă din perioada fetală (8-10 săptămâni) și este deja completă în momentul nașterii. Pe lângă funcția sa de organ auditiv, are o structură comună care este împărțită de majoritatea populației. Acest lucru devine interesant în momentul în care ne îndreptăm întreaga atenție către partea exterioară, vizibilă a acestui organ. Observăm o serie de forme ce definesc trăsăturile unice: tragus, antitragus, helix, antihelix, lobul, incizura și altele pe care le putem observa în figura 3.

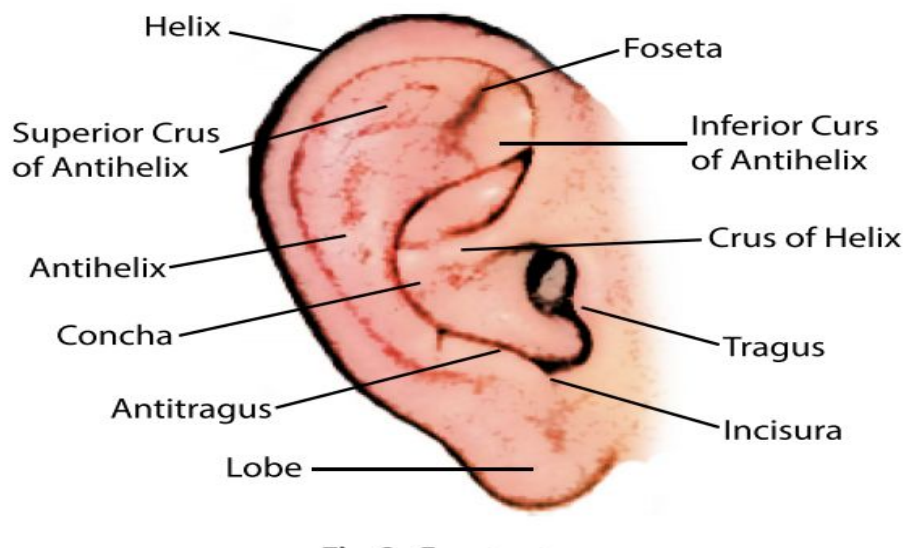


Fig. 3 Principalele forme definitorii ale urechii externe

Formele, din punct de vedere anatomic, pot varia de la om la om în funcție de factori precum mărime, poziție și textura (prin acest termen ne referim spre exemplu la tipuri de piele), având ca posibilitate, exploatarea lor pentru recunoașterea identității. Trebuie ținut cont că fiecare persoană deține două urechi. Cea din partea dreapta este foarte asemănătoare celei din stânga, lucru ce facilitează tehnicile automate de recunoaștere și oferă posibilitatea creșterii șanselor de recunoaștere. Nu poate fi trecută cu vederea însă că această tehnică de validare încrucișată între cele două urechi, nu are rata de succes de 100% dar experimente au arătat că în 9 din 10 cazuri simetria există, iar procesul are loc cu succes.

1.2.2 Avantaje și dezavantaje

O serie de factori pro și contra pot fi aduși în discuție pentru a argumenta eficiența acestui element biometric în contextul recunoașterii de persoane.

Urechea, deși nu este cel mai puternic descriptor al unui individ și nici cel mai compact, în continuare deține o sursă grozavă de informație într-o zonă destul de restrânsă comparativ cu alte trăsături biometrice. Putem aminti cazuri în care tehnici populare precum recunoașterea facială au dat greși din cauza imposibilității diferențieri trăsăturilor. Exemplu aferent îl întâlnim la indivizii gemeni, unde facial, poate deveni imposibil de stabilit identitatea fiecăruia. Aici intervine o calitate foarte importantă a biometriei urechii și anume, caracterul puternic distinctiv al formelor. Diferite studii au arătat că gemenii, analizați în detaliu, prezintă diferențe semnificative ale urechii externe, recunoașterea bazată pe biometria urechii fiind soluția acestui caz de excepție.

Un alt avantaj al utilizării acestei tehnici este puterea de suplimentare a informațiilor biometrice către sistemele automate. Nu este restricționată implementarea ei de una singură, ci poate servi unor combinații de tehnici care vor obține rezultate mult mai precise.

Fiind o tehnică ce se bazează pe captura de imagini, este influențată direct de factorii ce dezavantajează acest proces și anume: iluminare, unghi, calitatea instrumentelor. O altă problemă existentă este imposibilitatea menținerii încrederii în seturile de date. Oamenii evoluează odată cu trecerea timpului iar, urmăriți îndeaproape, s-au observat modificări în această zonă a aparatului auditiv. Urechile au crescut în mărime considerabil la bărbați, la femei existând șansa de a rămâne în parametri constanți. Acest fapt afectează în mod negativ sistemele biometrice, fiind incapabile de a mai potrivi corect persoanele testate. Cercetările în această direcție încă sunt deschise, dar ca soluție rudimentară, aprovizionarea cu seturi de date noi, într-un mod constant, pare a funcționa.

1.2.3 Studii anterioare

Cercetările în această direcție au început cu precădere la începutul anilor '60, însă există lucrări ce menționează acest subiect încă dinaintea Primului Război Mondial. În perioada mai recentă, tehnicile automate de recunoaștere a urechii au fost în general divizate în tehnici care, fie lucrează cu date 2D, fie cu date 3D. Principalele cercetări se bazează pe datele 2D, acestea fiind utilizate și în algoritmii care au servit la obținerea rezultatelor aferente acestei lucrări.

Interpretarea datelor 2D se poate realiza prin mai multe metode printre care putem distinge următoarele:

- Geometrică: urmărește explorarea geometrică a trăsăturilor, precum forma generală a urechii, poziționarea diferitelor părți ale sale și relațiile dintre acestea;
- Locală: tratează zone mult mai restrânse ale imaginii, colectând informația pe segmente, utilizând-o ulterior pentru recunoaștere;
- Holistică: diferită conceptual de cea geometrică și se poziționează la un pol opus; această metodă tratează urechea ca pe un întreg și exploatează fizionomia globală a imaginii;
- Hibrid: implică utilizarea a două sau mai multe tehnici anterioare, având ca obiectiv incrementarea acurateții și a ratei de recunoaștere;

O scurtă recapitulare a câtorva metode folosite în trecut pentru a studia capacitatea și puterea discriminativă a urechii, o putem observa în tabelul următor care ne prezintă detalii, precum metoda abordată, tipul acesteia, autorul, performanța și setul de date folosit.

2007, Nanni and Lumini [50]	Multi-matcher	Hibrid	UND E	114	80 (R1)
2007, Nosrati et al. [36]	Wavelet transformation and PCA	Hibrid	USTB II	77	90.5 (R1)
			CP	17	95.0 (R1)
2007, Rahman et al. [51]	Geometric features	Geometric	Own	100	88 (R1)
2007, Yuan and Mu [52]	Full-space LDA	Holistic	USTB II	77	90 (R1)
2008, Arbab-Zavar et al. [53]	Log-Gabor filters	Local	XM2VTS	63	85.7 (R1)
2008, Choras [54]	Geometry of ear outline	Geometric	Own	188	86.2 (R1)
2008, Dong and Mu [55]	Force field transform and NKDFA	Holistic	USTB IV	29	75.3 (R1)
2008, Guo and Xu [56]	Local binary pattern and cellular NN	Local	USTB II	77	93.3 (R1)

Fig. 4 Fragment din tabelul unor studii comparative ce prezintă performanța algoritmilor

2 Tehnologii utilizate

Pentru aplicarea într-un cadru cât mai optim al algoritmilor de procesare și de antrenament al datelor, am căutat setul de instrumente cu cele mai bune rezultate și care oferă o flexibilitate cât mai mare posibilă.

2.1 Python

Python este un limbaj de programare de nivel înalt și este ușor controlabil, atât de programatorii începători, cât și de cei cu experiență. Este considerat cel mai matur și bine documentat dintre limbajele de programare ce își propun rezolvarea problemelor de învățare automată, iar acesta este motivul principal pentru care dezvoltatorii aleg să îl folosească în procesarea imaginilor.

Putem evidenția câteva proprietăți riguroase ce ne dovedesc eficiența sa:

- Ușor de scris: Python folosește cuvinte cheie foarte simple, ușor de reținut, datorită concentrării pe care o acordă legăturii cu limba engleza; acest lucru oferă posibilitatea dezvoltatorilor să-și intensifice atenția pe complexitate, arhitectură și mai puțin pe denumiri cheie;
- Limbaj dinamic: acest lucru ne eliberează de barierele ridicate de către multe limbaje în zona tipurilor structurilor de date; având această proprietate, structuri precum vectorii, de care ne vom folosi cel mai mult, pot manipula date mixte, ușurând complexitatea globală a codului;

- Prototipizare rapidă: oferă un suport foarte bun de lucru pentru integrarea funcțiilor ce sunt scrise în alte limbaje; exemplu perfect este integrarea librăriei OpenCV bazată pe C/C++, având oportunitatea astfel de a dezvolta o aplicație într-un mediu ușor de utilizat și de a lucra cu funcții foarte bine optimizate din punctul de vedere al performanței și al puterii de calcul.
- Numeroase librării pentru învățare automată: utilizat din plin pe această ramură, o mulțime de dezvoltatori își investesc timpul în crearea librăriilor; în acest mod, lucrând cu procesarea imaginilor, ne este facilitată implementarea proiectelor care ar necesita o perioadă foarte mare de pregătire a mediului.
- Este open source: nu are nevoie de un abonament plătit iar, asemenea limbajului MATLAB, este specializat pe analiza datelor, explorare, studii și vizualizare; de asemenea, poate fi ușor integrat cu diverse framework-uri;
- Popularitate: fiind în fruntea limbajelor software, comunitatea este foarte mare; acest lucru ne oferă acces la forumuri create special pe aria noastră de interes și putem avea parte mereu de ajutor atunci când întâmpinăm probleme;

2.2 OpenCV

Librărie a cărei popularitate este de necontestat- va fi utilă tuturor proceselor pe care le vom utiliza în rezolvarea problemei. OpenCV deține cea mai vastă documentație specializată pe procesarea imaginilor. Reprezintă puntea dintre multitudinea de pixeli pe care o vedem într-o

image ca un întreg și mediul de dezvoltare în care devine descompusă, facilitând astfel posibilitatea analizelor și a observațiilor în detaliu ai acestor pixeli.

Prin funcțiile specifice pe care le oferă, putem obține, stoca și transmite mai departe fluxul de informație dintr-o imagine. Structura principală pe care o vom folosi sunt matricile n-dimensionale. Numărul de dimensiuni va depinde de modalitate în care dorim obținerea pixelilor. Astfel, o imagine color, va necesita o matrice tridimensională, cea alb-negru având nevoie doar de una bidimensională.

```
#Taking input from a directory path
img = cv2.imread('C:\Users\USER\Desktop\image.jpg',0)

#Capturing input from a video stream
cap = cv2.VideoCapture(0)
```

Fig. 5 Modalitatea prin care stocăm fluxul de informație dintr-o imagine

Odată ce am obținut informația, librăria ne oferă o multitudine de funcții ce o pot altera în vederea obținerii rezultatelor dorite: redimensionare, rotații și translații, segmentare, filtrare, detectare de margini sau de contururi, extragere de trăsături, etc. După cum am menționat într-un capitol anterior, atuul acestei librării este limbajul în care a fost dezvoltată și anume C++. Vom lucra preponderent cu operații matematice, iar C++ ne va ajuta considerabil în a diminua timpul de execuție.

2.3 Scikit-learn

SK-Learn reprezintă un proiect pornit inițial la Google Summer Of Code în 2007, de către o singură persoană, ajungând în anii următori să se alăture mulți alți dezvoltatori ce au susținut inițiativa. În momentul de față este o librărie destinată pasionaților de învățare automată și oferă nenumărate implementări ale algoritmilor matematici de înaltă complexitate.

O folosim în special pentru partea de antrenament a trăsăturilor pe care le vom extrage din setul nostru de date, completându-ne astfel lipsa unor funcții din OpenCV. Este cea mai flexibilă librărie dintre cele existente când vorbim despre optimizare și despre alternarea parametrilor. Avem parte de o documentație destul de riguroasă ce ne poate ghida prin înțelegerea mai în detaliu a lucrurilor care se întâmplă în spatele cortinei. De asemenea, avem avantajul comunității care ne poate oferi suport în eventualele probleme apărute.

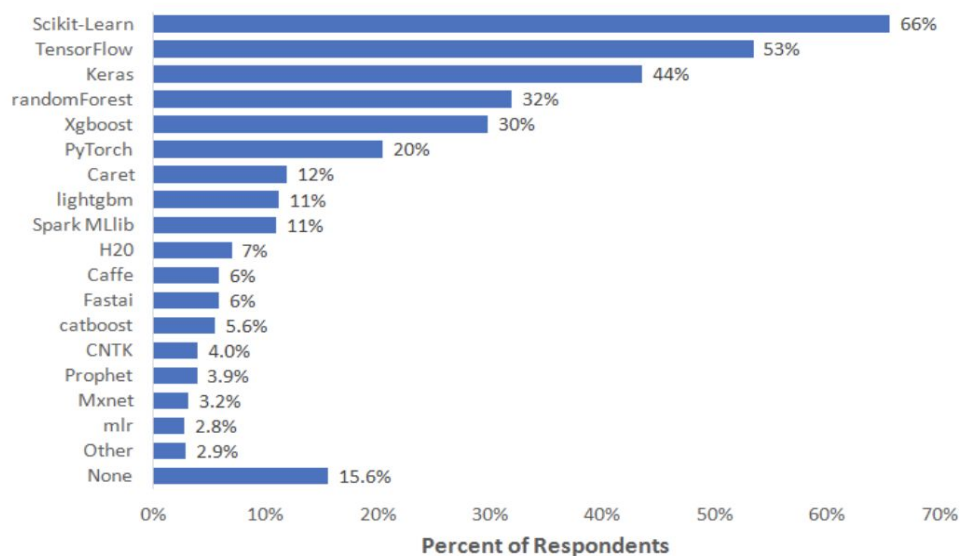


Fig. 6 Graficul popularității librăriei Scikit-learn din 2018

3 Implementare

3.1 Setul de date

Un prim lucru foarte important pe care l-am stabilit înainte de a începe dezvoltarea aplicației, a fost să obținem o bază de date destul de solidă care să ne permită flexibilitatea de a încerca cât mai multe modalități de recunoaștere. De asemenea, am căutat ca aceasta să fie diversificată (genul individului, poziționare, numărul de imagini) și să fie publicată online, special destinată studiilor de acest tip.

DB	Year	#Subjects	#Images	Sides	Pitch	Yaw	Occlusions	Accessories	Gender	Ethnicities
CP [89]	1995	17	102	Left	None	None	None	None	Both	White
USTB I [90]	2002	60	185	Right	Mild	None	None	Yes	Both	Asian
USTB II [90]	2004	77	308	Right	Mild	Mild	None	Yes	Both	Asian
IITD I [9]	2007	125	493	Right	Severe	None	None	Yes	Both	Asian
AMI [91]	NA	100	700	Both	Severe	Mild	mild	None	Both	White
WPUT [8]	2010	501	2071	Both	Severe	Mild	Severe	Yes	Both	White
IITD II [9]	NA	221	793	Right	None	None	None	None	Both	Asian
AWE	2016	100	1000	Both	Severe	Severe	Severe	Yes	Both	Various

Fig. 7 Lista bazelor de date care au fost expuse studiilor până în prezent

După analiza tabelului de mai sus, am ales să utilizăm setul de date AMI(tradus din spaniolă- Analiza Matematică a Imaginilor). A fost creat cu ajutorul persoanelor de la Universitatea Las Palmas și consta în 700 de imagini cu urechi provenite de la 100 de persoane diferite cu vârste cuprinse între 19 și 65 de ani. Toate imaginile au fost preluate sub aceiași factori de luminozitate și cu o aceeași poziționare fixă. Fiecare individ este reprezentat prin 7 imagini, una cu cea stângă, restul fiind cu cea dreaptă, care pot varia în perspectivă și poziționare. Toate urechile setului de date sunt din profil și sunt decupate destul de puțin, urechea acoperind

aproximativ jumătate din totalul pixelilor din imagine. Acest lucru ne va permite să observăm cum vor răspunde ulterior algoritmi la un decupaj mai adânc.



Fig. 8 Cele șapte imagini ale unui individ din primul set de date

Modul în care ne-am organizat setul de date ales a constant în împărțirea sa în două părți mari: primii indivizi indexați de la 1 la 50, fiind urmați de ceilalți indexați de la 50 la 100. Trebuie menționat că aceștia sunt numerotați în baza de date originală de date cu valori de la 1 la 110 din cauza lipsei de poze pentru anumiți indivizi. Ei își păstrează în continuare numărul total de 100.

Astfel am obținut setul de date 1 (1-50) și setul de 2 (50-100). Inițial, am plecat cu o metodă clasică și anume Cross Validation Leave One Out(CVLOO). Pe parcursul lucrării ne vom adresa acestei metode cu acronimul LOO. Pentru a reuși să surprindem mai multe modalități de testare, am urmat și o abordare 2-Fold Cross Validation, obținând astfel un procentaj de antrenament-testare de 70%-30%. (ne vom adresa cu 2-fold în mențiuni ulterioare).

Ambele tehnici urmează același proces: se alege un număr de poze (în cazul nostru 1 sau 2) și sunt păstrate pentru testare. Restul imaginilor vor constitui setul de antrenament sau selecția poate continua, tocmai pentru a descoperi ce factori și ai căror imagini ne oferă cele mai bune rezultate.

Pe parcursul studiilor, setul de date inițial a fost alterat în funcție de nevoi. Fiind un set foarte puțin decupat, rezultatele au putut fi alterate și am încercat să vedem cum putem rezolva această problemă. În acest fel, am creat pentru ambele seturi menționate mai sus, încă două seturi adiționale (denumite `CroppedSet1` și `CroppedSet2`), care surprind aceleași imagini, însă urechea cuprinzând în jur de 70% din totalul de pixeli al imaginii. Toți algoritmi au alternat rezultatele în jurul acestor patru seturi de imagini.

Reprezentarea setului de date în aplicație va fi o matrice, ai căror linii vor fi imaginile. Prin urmare, pe rând, fiecare imagine va fi încărcată prin intermediul funcțiilor din OpenCV, sub forma unui vector de pixeli, urmând a fi adăugat în matricea globală de imagini. Toate operațiile ulterioare vor fi realizate utilizând această matrice.

3.2 Metode

3.2.1 Eigenears

Am ales să studiem în primă fază, un algoritm holistic ce cunoaște o foarte mare aplicabilitate în zona procesării de imagini și are la bază PCA (Principal Component Analysis). El este cunoscut sub numele de Eigenfaces fiind utilizat în mare parte în recunoașterea facială. Am dorit, totuși, să observăm rezultatele sale ca răspuns la problema propusă.

Principal component analysis este o metoda de reducere a dimensiunilor ce ne permite să extragem trăsături sub forma unor caracteristici generale. În cazul imaginilor, putem spune că numărul dimensiunilor este de fapt numărul trăsăturilor pe care acestea le dețin. Cert este că nu toate trăsăturile sunt definatorii pentru o imagine. Ceea ce ne

oferă PCA este o observație generală a unor trăsături comune pe care le putem utiliza ulterior pentru a scoate în evidență trăsăturile unice ale fiecărei imagini. Toate aceste trăsături comune vor fi stocate, independent, în vectori cu un număr de $W \times H \times 3$ de elemente (unde W este lățimea imaginii, H înălțimea, iar 3 numărul de canale corespondent culorilor RGB), care sunt denumiți eigenvectori. Procesul de obținere al eigenvectorilor este facilitat de către OpenCV care ne va calcula vectorul medie al tuturor imaginilor din baza de date împreună ca numărul de eigenvectori pe care îl dorim.

Având matricea globală a imaginilor, o setăm ca sursă de informație pentru algoritmul PCA oferit de OpenCV, obținând următoarele:

- imaginea medie a urechilor care în cod va fi transpusă într-un vector

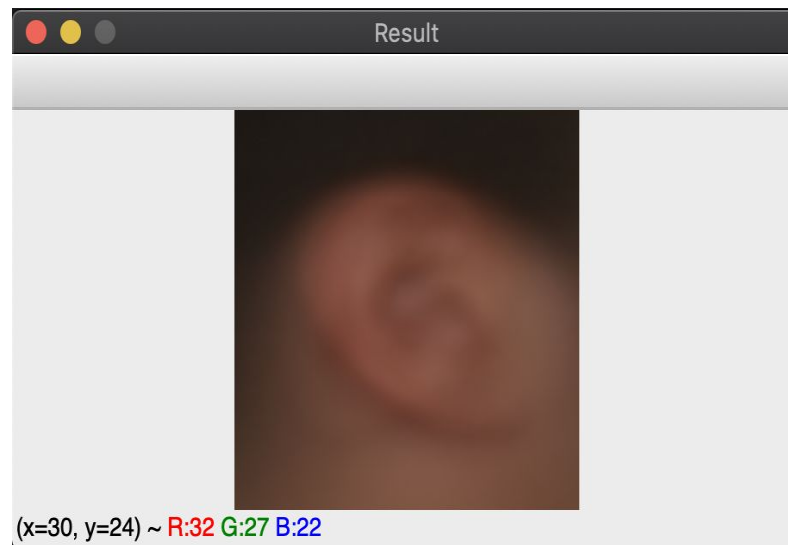


Fig. 9 Reprezentarea vizuală a vectorului mediu

- eigenvectorii aferenți setului de date, în cazul nostru fiind aleși 10

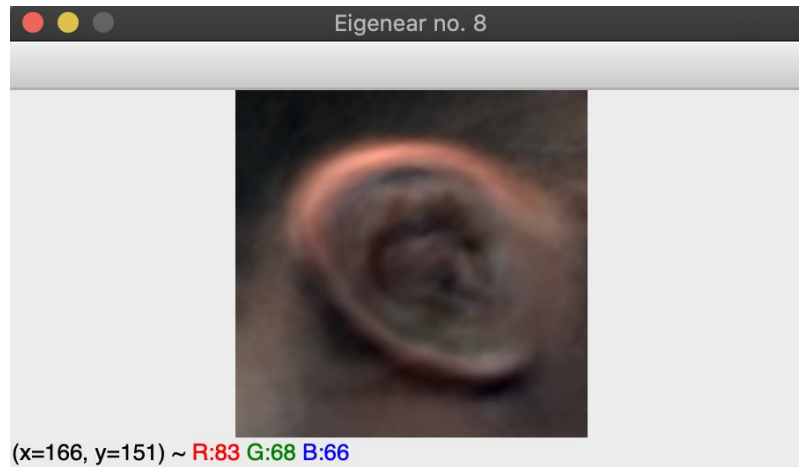


Fig. 10 Reprezentarea vizuală a eigenvectorului cu indexul 8

Următorul pas pe care l-am realizat a fost să eliminăm informația comună din fiecare imagine din setul de antrenament. Astfel, am scăzut din fiecare vector al matricii globale, vectorul mediu obținut în urma PCA. Acum, ținta noastră este să aflăm care este puterea discriminativă a fiecărui vector de trăsături (eigenvectorii) asupra fiecărei imagini. Puterea discriminativă va fi descrisă de un vector denumit vector de weight-uri. În cazul curent, fiecărei imagini din set i se va calcula un vector de acest tip. Alegând să lucrăm cu 10 eigenvectori, un vector de weight-uri va avea exact 10 valori, valoarea cu indexul i reprezentând importanța eigenvectorului cu același index în contextul imaginii studiate. Definind acești pași, obținerea vectorului de weight-uri necesită calcularea produsului scalar dintre vectorul imaginii (din care s-a scăzut vectorul mediu) cu fiecare dintre eigenvectori. Acest lucru se va repeta pentru fiecare imagine, în final obținând o matrice a cărei linii vor fi weight vectorii, în relație de 1 la 1.

Matricea globală de weight-uri va fi cea de care ne vom folosi pentru a recunoaște imaginile rămase în setul nostru de testare. Urmând metoda LOO, fiecare individ a avut o imagine scoasă din setul de antrenament. Fiecărei imagini de test, îi aplicăm pe rând, aceiași pași de mai sus. La

fiecare iterație, după calcularea vectorului de weight-uri al imaginii curente de testare, nu ne rămâne decât să folosim o metrică care să ne permită măsurarea distanțelor dintre vectori.

Din acest punct, am ales să mergem pe doua abordări: fie tratăm fiecare vector de weight ca pe o clasă separată, fie facem o medie între toți vectorii de weight-uri ai aceluiași individ. Cu prima variantă, fiecare individ va fi descris prin 6 vectori diferiți (300 de vectori pentru tot setul), iar în două, doar printr-unul singur (50 de vectori). În ambele cazuri am utilizat o foarte bine cunoscută metodă de măsurare și anume, distanța euclidiană. Calculând distanța euclidiană cu fiecare vector de weight-uri din setul de antrenament vom obține un vector de distanțe, cu un număr de exact N elemente, unde N este numărul de imagini. Valoarea minimă a acestui vector va fi de fapt, distanța dintre imaginea de testare și imaginea cea mai asemănătoare acesteia din set. Pentru a ști care este imaginea corespondentă distanței, salvăm și indexul ei, vectorul nostru de distanțe devenind o înlanțuire de tuple formate din valoarea distanței euclidiene și indexul imaginii corespunzătoare vectorului de weight-uri.

Pentru a clasifica imaginile am utilizat tehnica k-NN, unde k am ales, pe rand, valori din mulțimea [1,3,5]. Aceasta influențează mulțimea valorilor pe care le vom compara din vectorul de distanțe. De exemplu, utilizând 3-NN ca și clasificator, ne vom orienta în funcție de primele 3 valori cele mai mici din vectorul de distanțe. Dacă indexul acestora ne va indica faptul că diferență a făcut realizată între imaginile aceluiași individ, vom considera că algoritmul nostru a clasificat corect imaginea de testare.

Pe setul 1, am reușit să obținem un procentaj destul de mic în ceea ce privește rata de recunoaștere. Observația principală este că această algoritm

poate fi îngreunat de mai mulți factori precum utilizarea imaginilor color în locul celor alb-negru, nepotrivirea mărimii imaginilor (lățime și înălțime), metrică folosită. De asemenea, există posibilitatea incapacității urechii de a oferi o imagine de ansamblu a unei trăsături generale. Astfel, această tehnică ar putea funcționa pe anumite seturi de date cu urechi mult mai bine decupate sau pe urechi ce scot mai bine formele unice în evidență.

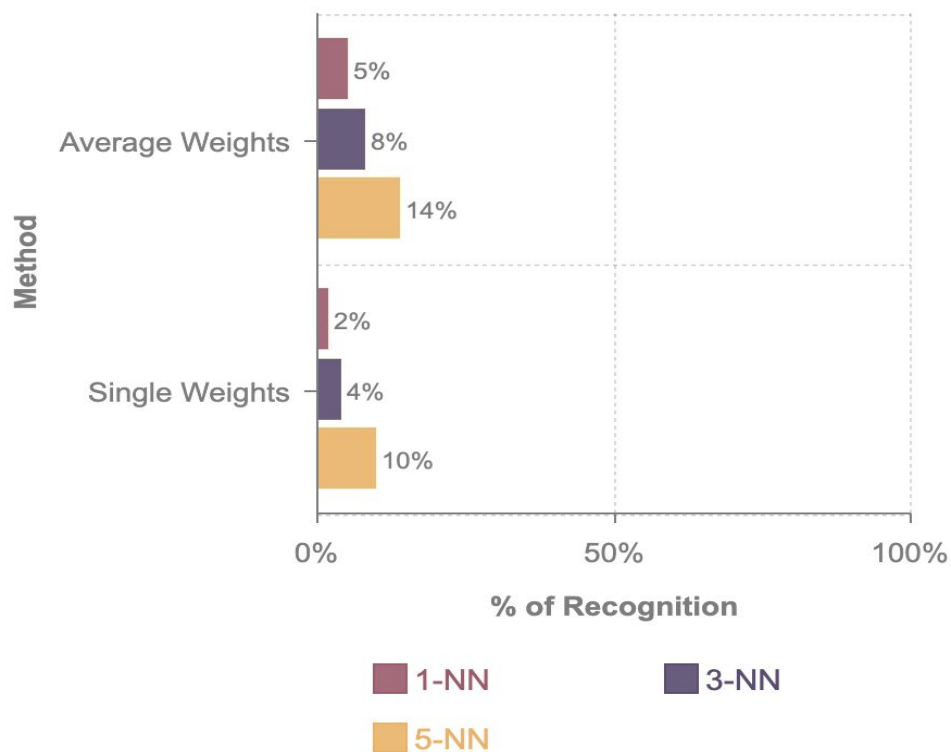


Fig. 11 Graficul performanței utilizând metoda eigenvectorilor împreună cu k-NN

3.2.2 HuMoments

Am încercat să aplicăm o tehnică mai puțin populară în ceea ce privește recunoașterea persoanelor pentru a studia posibilele rezultate în această direcție. HuMoments este o metodă ce se folosește de obicei în studiul recunoașterii formelor de diferite tipuri. (de exemplu, litere de tipar sau de mână). Ceea ce îl face eficient în detectarea formelor, este modul în care acesta tratează pixelii. Am vrut, totuși, să vedem ce fel de rezultate ne poate aduce implementându-l.

Momentele sunt de fapt o medie a intensității pixelilor. Pentru a le calcula, imaginea noastră va trebui binarizată, fiecare pixel fiind scalat la 0 sau 1 în funcție de un anumit prag pe care îl selectăm. După ce binarizarea a fost realizată, obiectivul nostru este de a găsi momentele și ulterior momentele Hu pentru a putea măsura rezultatele. În lucrare, ambele vor fi calculate prin intermediul funcțiilor din OpenCV, însă vom explica cum decurge procesul.

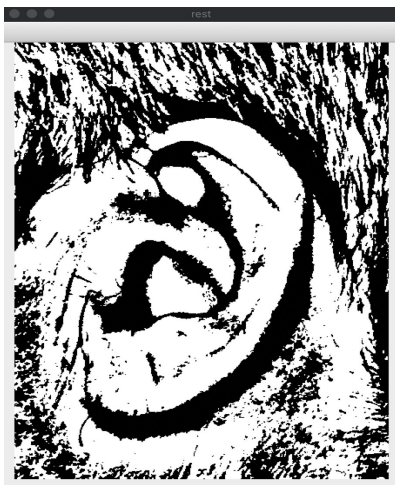


Fig. 12 Imaginea unei urechi binarizate cu funcția `cv2.adaptiveThreshold`

Momentele brute, care sunt de fapt sume ale intensității pixelilor, după cum am menționat anterior, nu ne sunt de ajuns pentru a putea identifica noi modele. Acestea sunt încă vulnerabile diferitelor tipuri de operații precum translatarea, rotația, scalarea, iar ținta noastră este să identificăm urechile indiferent de ce ponderi au acești factori în imaginea testată. Astfel, momentele centrale vor fi cele pe care vom dori să le utilizăm. Valorile acestora vor fi obținute cu ajutorul centroizilor formelor din imaginea binarizată. Normalizarea valorilor ne va obține noi momente care sunt, de fapt, invariante din perspectiva scalării, rotației, translației, etc. În final, putem admite că am aflat momentele H_u ale imaginii respective, într-un număr total de 7, fapt ce ne va permite să detectăm aparițiile aceleiași forme(în cazul nostru formele urechii) în imaginile de test, indiferent de modificările întâlnite asupra ei.

Un foarte important factor ce poate influența capacitatea algoritmului de a recunoaște corect este funcția de thresholding pe care o folosim pentru binarizare. OpenCV ne pune la dispoziție mai multe funcții care pot scoate mai multe sau puține trăsături în evidență în funcție de caz. Pe lângă funcția selectată, trebuie să alegem cu atenție pragul după care dorim binarizarea. Acesta trebuie observat după cum creșterea sau scăderea lui pot influența rezultatele obținute.

Având momentele H_u într-o matrice globală, fiecare poziționându-se pe linia cu același index corespondent imaginii pe care o reprezintă, trebuie să încărcăm imaginile de test și să aplicăm algoritmul descris mai sus. După ce toți pașii de obținere a momentelor au fost realizate, ne rămâne doar să alegem o metrică și un clasificator. Am folosit aceeași abordare ca în

metoda anterioară și anume, distanța euclideană urmată de k-NN pentru clasificare. Rezultatele au fost surprinse într-un grafic în funcție de metoda de thresholding utilizată. De menționat este că pentru seturile de date s-a folosit o împărțire de tip LOO.

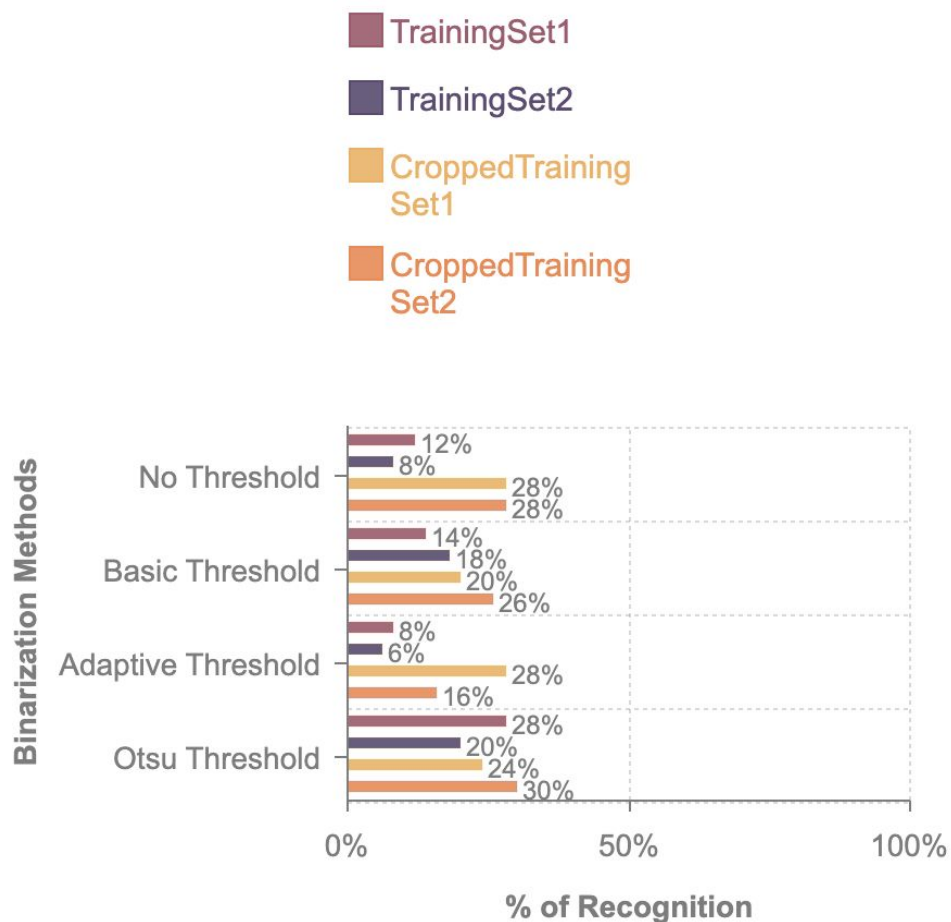


Fig. 13 Ratele de succes ale metodei HuMoments în funcție de threshold

Concluzionând, rezultatele sunt mai bune decât ce ne-a oferit PCA, însă, nu ne putem baza pe această tehnică de una singură. În funcție de situație, aceasta ar putea fi încă utilizată pentru a aduce un plus de informație. Prin urmare, putem considera momentele Hu ca pe o metodă

de optimizare a acurateții în cazul în care o altă metodă întâlnește impedimente în a da randament.

3.2.3 Local Binary Patterns

Local Binary Patterns (LBP) este un algoritm devenit popular în anii 2000 ca și descriptor de texturi. Spre deosebire de metodele holistice, acesta funcționează analizând imaginea pe zone, prin urmare, avem de a face cu o metodă locală de reprezentare a informației.

Procesul prin care LBP examinează pixelii imaginii este bazat pe vecinătățile acestora. Prima fază constă în eliminarea canalelor colore, utilizând un filtru alb-negru. Vecinătatea pixelilor va fi stabilită de noi, aceasta fiind de fapt factorul principal ce ne influențează rezultatele metodei. Conform acestui lucru, va trebui să alegem o rază R , în jurul pixelului nostru curent, care se numește pixelul central. Pe lângă acest prim parametru, vom putea stabili și numărul punctelor vecine, P , pe care vrem să îi luăm în calcul.

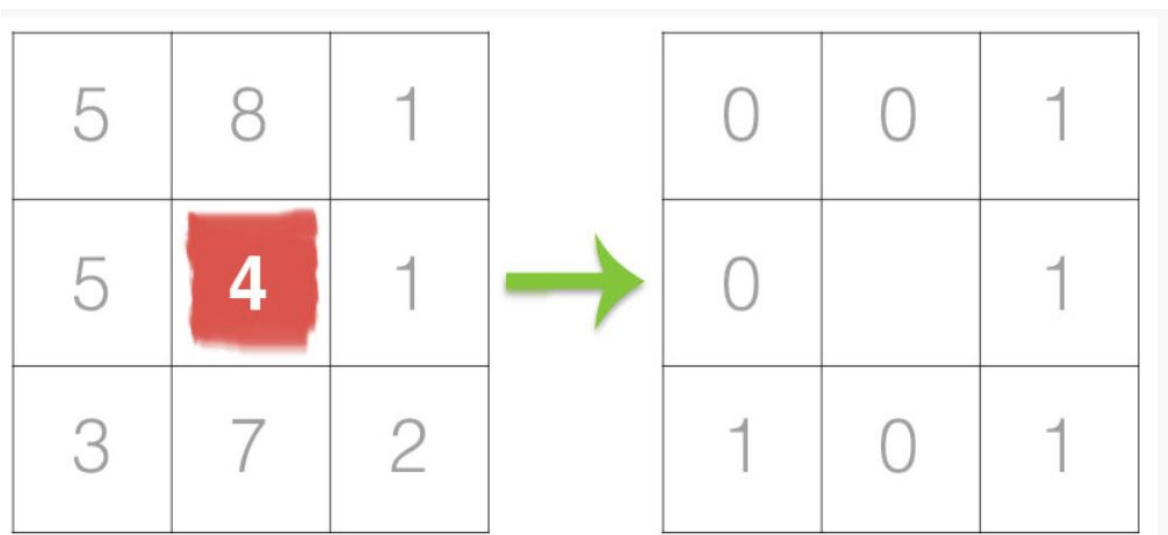


Fig. 14 Modalitatea prin care se calculează pixelul central într-o matrice 3x3 cu $R=1$ și $P=8$

Odată stabiliți vecinii, îi vom compara cu pixelul central. Dacă valoarea este mai mică, vom seta vecinii cu valoarea 1. În caz contrar, vom seta valoarea 0. Cu aceste valori va trebui să formăm un număr în baza 2, alegând mai întâi un colț din care începem formarea numărului. Colțul este irelevant, deoarece informația va fi codată la fel pentru toți pixelii din imagine. Numărul binar obținut va fi convertit în baza 10, urmând să fie plasat în locul pixelului central. În final vom obține o matrice cu aceeași înălțime și lățime ca cea originală. Având această matrice de informație, nu ne rămâne decât să o punem într-o histogramă pentru a construi vectorul de trăsături corespunzător.

Tot acest proces, l-am încadrat într-o singură clasă pentru a facilita utilizarea. Pentru a ne folosi de algoritm, va trebui doar să instanțiem clasa și să îi pasăm cei doi parametri: raza vecinătății și numărul de vecini în funcție de care dorim calcularea pixelilor. Partea de procesare și anume calcularea vecinilor și înlocuirea pixelului central ne este facilitată de o funcție a librăriei scikit-learn.

```
5
6 class LocalBinaryPatterns:
7     def __init__(self, numPoints, radius):
8         self.numPoints = numPoints
9         self.radius = radius
10
11     def describe(self, image):
12         lbp = feature.local_binary_pattern(image, self.numPoints,
13         self.radius)
14
15         (hist, _) = np.histogram(lbp.flatten(),
16         bins = np.arange(0, 256),
17         range=(0, 255))
18         return hist
```

Fig. 15 Clasa definită pentru utilizarea algoritmului LBP

Fiecare imagine va genera un vector de trăsături specific imaginii respective. Aceștia vor fi salvați într-o matrice globală, în paralel salvând în altă matrice și numărul individului studiat, care va fi eticheta asignată trăsăturii respective.

Pentru a putea clasifica imaginile de test, am avut nevoie mai întâi să utilizăm un algoritm care să ne antreneze informațiile privind fiecărui vector de trăsături eticheta aferentă din vectorul de etichete. Pentru această operație am folosit în primă instanță Support Vector Machine (SVM) și în final, Linear Discriminant Analysis (LDA). Cei doi algoritmi sunt implementați în librăria scikit-learn și îi vom utiliza conform documentației existente. Pentru SVM, am putut să setăm numărul de iterații care, în funcție de setul de date, trebuie adaptat pentru a suplimenta capacitatea de antrenament. În continuare, lucrurile sunt pregătite pentru a clasifica imaginile de test. Urmând pașii LBP, obținem histograma imaginii de testare, urmând să o oferim ca parametru clasificatorului. Pe baza informației antrenate, acesta ne va prezice ce etichetă s-ar potrivi imaginii de testare. Rezultatele le salvăm într-o matrice finală pentru a observa câte imaginii au primit etichetele corecte. De asemenea, procentajul final de recunoaștere se calculează în funcție de această matrice.

Am surprins rezultatele oferite de ambele metode în două grafice, împărțite în funcție de metoda de antrenament și clasificare. După cum putem observa în cele două imagini, rezultatele sunt detasat mai bune față de abordările anterioare. Putem spune ca Local Binary Patterns, ca și metodă de extragere a trăsăturilor, a surprins cel mai bine informația, indiferent de metoda de clasificare utilizată.



Fig. 16 Rezultatele obținute de LBP alături de SVM

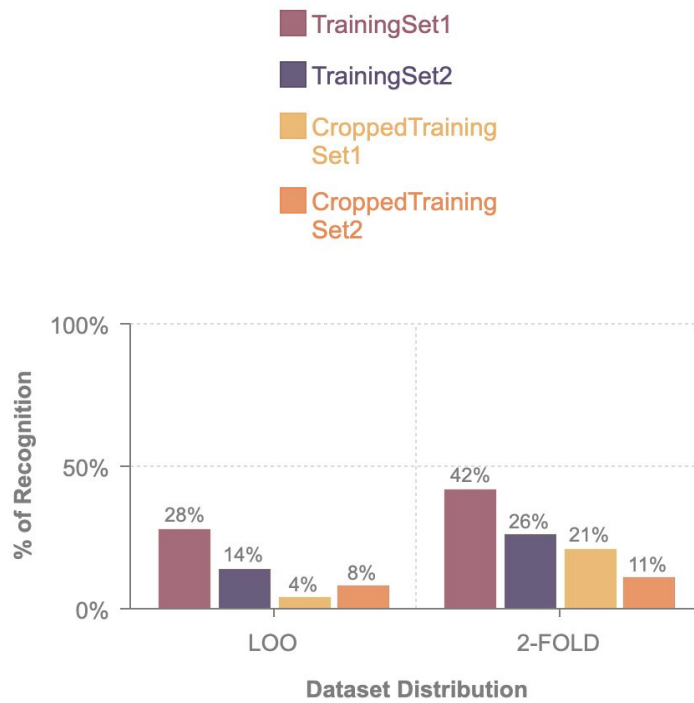


Fig. 17 Rezultatele obținute de LBP alături de LDA

Putem observa procentajul de recunoaștere pe care îl avem împreună cu SVM. Există diferențe între modalitatea de distribuție, astfel că putem observa că rata de recunoaștere a scăzut cu cel puțin 25% în momentul în care am utilizat doar 5 imagini pentru antrenament și am lăsat 2 pentru testare. Putem sesiza că este esențial să avem urechea din toate unghiurile în setul de antrenament pentru a reuși să recunoaștem orice imagine de testare. Avantajul în LOO l-am avut, deoarece în setul de antrenament foloseam o imagine ce avea exact același unghi fotografiat ca cea din testare, astfel încât am văzut capacitatea algoritmului de a oferi 100% rata de succes. Cum au mai fost adăugate fotografiile la testare, rata a scăzut neavând suficientă informație pentru a recunoaște corect.

Mai mult, observăm creșterea ratei de succes cu cel puțin 10% atunci când folosim un set de date decupat. Concluzia pe care o putem trage este că alte elemente, precum părul, obiecte puse pe/în ureche, pot brui calitatea rezultatelor.

În ceea ce privește îmbinarea cu LDA, am observat o capacitate mai mare de recunoaștere atunci când am folosit distribuirea 2-fold, însă nu la fel de bună comparativ cu SVM.

Un rol foarte important în algoritmul nostru l-au jucat parametrii setați pentru LBP: raza și numărul de puncte utilizate în calcularea pixelilor centrali. Euristic, am ajuns la parametrii ce funcționează cel mai bine și se pliază pe modelul și caracteristicile urechii. Astfel, pentru SVM, având acces la un articol legat de studii asupra recunoașterii folosind urechea, am aflat că parametrii optimi în aplicarea LBP, ar putea fi: $R=18$ și $P=10$. Am încercat să obținem rezultate mai bune cu alți parametri, însă nu am reușit, aceștia oferind rezultatele prezentate în grafic. Pentru abordarea cu LDA, aceștia nu

au mai funcționat, prin încercări reușind să găsim cele mai bune rezolvări alături de parametrii: $R=10$ și $P=10$.

4 Concluzii

Problema recunoașterii persoanelor cu ajutorul urechii este o problemă ce deține soluții, dar rezultatele pot varia de la slabe la foarte bune. Observațiile generale pe care le putem desprinde din comportamentul metodelor aplicate sunt că acest element biometric are foarte mult potențial de recunoaștere, având detalii foarte bine conturate însă este nevoie de o tehnică potrivită. Ratele cele mai mari de succes au fost obținute de metodele locale, pe când cea holistică nu a reușit să ne ofere prea multă satisfacție.

Recunoașterea cu ajutorul urechii este ușor influențată de unghiul din care este realizată fotografia sau de informații în plus, precum texturile din jurul elementului studiat (păr, obiecte). Un plus major asupra aplicației consider că l-ar aduce un decupaj automat al urechii, astfel încât să reținem doar informația absolut necesară. Acest lucru este puternic motivat de rezultatele pe care le-am văzut îmbunătățindu-se de fiecare dată când utilizam setul de date cu 20% mai decupat. Problema unghiurilor ar putea fi rezolvată cu ajutorul unui set de date mult mai detaliat pentru fiecare individ, astfel încât, imaginile de testare să poată fi recunoscute indiferent de direcția din care sunt făcute fotografiile.

Cu toate acestea, precum am menționat și într-un capitol anterior, urechea nu trebuie utilizată ca element biometric de unul singur. Ea poate veni în ajutorul altor elemente precum fața unui individ. Această combinație de elemente poate fi ușor implementată în securitatea publică. Putem presupune că nu avem acces la imaginea în mare parte a feței unui

individ ce trebuie identificat într-un sector public, însă avem parte doar de imaginea sa din profil. Având datele faciale alături de cele ale urechii, șansa de recunoaștere va fi mult mai ridicată, fiind posibilă prelucrarea informației provenită din mai multe elemente.

De-a lungul lucrării, am încercat să obținem rezultate cât mai optime, aplicând și diverse metode, populare sau mai puțin populare, tocmai pentru a observa comportamentul urechii în contextul recunoașterii biometrice. Rezultatele au fost prezente, în final, reușind să obținem rate foarte bune de succes, având în minte și îmbunătățirii viitoare prin care putem extinde aria de recunoaștere a acestui element biometric. Toate acestea fiind spuse, scopul inițial, și anume aducerea urechii în punctul în care putem identifica indivizi cu ajutorul informației expuse de aceasta, a fost atins.

5 Bibliografie

1. *Ear recognition: More than a survey*, de Žiga Emeršic, Vitomir Štruc și Peter Peer
2. *A comparative Study on Texture and Surface Descriptors for Ear Biometrics*, de Anika Pflug, Pascal Nicklas Paul și Christoph Busch
3. *Ear recognition system using principal component analysis*, de Avijit Mathur
4. *LBP-based ear recognition*, de Nazmeen Boodoo și S.Baichoo, https://www.researchgate.net/publication/261089724_LBP-based_ear_recognition
5. *Local Binary Patterns*, de Philipp Wagner, https://www.bytefish.de/blog/local_binary_patterns/
6. *Face Recognition with Python*, de Philipp Wagner, https://www.bytefish.de/pdf/facerec_python.pdf

7. *Face Recognition: Understanding LBPH Algorithm*, de Kelvin Salton do Prado,
<https://towardsdatascience.com/face-recognition-how-lbp-h-works-90ec258c3d6b>
8. *Face Recognition using Local Binary Patterns (LBP)*, de Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid și Md. Shafiul Azam,
https://globaljournals.org/GJCST_Volumel3/1-Face-Recognition-using-Local.pdf
9. *21st century technologies: a future of promise*, de Wolfgang Michalski,
<http://oecdobserver.org/news/archivestory.php/aid/48/21st-AOcentury-technologies:-a-future-of-promise.html>
10. *History of Biometrics*, de Stephen Mayhew,
<https://www.biometricupdate.com/201802/history-of-biometrics-2>
11. *History of Fingerprint Pattern Recognition*, de Simon A. Cole,
https://link.springer.com/chapter/10.1007/0-387-21685-5_1
12. *Biometrics and Face Recognition Techniques*, de Renu Bathia,
<https://pdfs.semanticscholar.org/a7cf/ede8225c99f6e1883d4ae14c66fb20191117.pdf>
13. *Importance of Biometric Fingerprinting Technology*, de Danny Thakkar,
<https://www.bayometric.com/importance-of-biometric-fingerprinting-technology/>
14. *Advantages of Using Python for Computer Vision*,
<https://fullscale.io/advantages-using-python-computer-vision/>

15. *Most Popular Machine Learning Frameworks and Products Used by Data Professionals*,
<http://businessoverbroadway.com/2019/01/27/most-popular-machine-learning-frameworks-and-products-used-by-data-professionals/>
16. *Image Processing with OpenCV*,
<https://medium.com/@ariesiitr/image-processing-with-opencv-45c3a5cefd10>
17. *OpenCV Functions to Start your Computer Vision journey*, de Saurabh Pal,
<https://www.analyticsvidhya.com/blog/2019/03/opencv-functions-computer-vision-python/>
18. *A Gentle Introduction to k-fold Cross-Validation*, de Jason Brownlee,
<https://machinelearningmastery.com/k-fold-cross-validation/>
19. *Eigenface using OpenCV (C++/Python)*, de Satya Mallick,
<https://www.learnopencv.com/eigenface-using-opencv-c-python/>
20. *Shape Matching using Hu Moments (C++/Python)*, de Satya Mallick si Krutika Bapat