

**Universitatea Tehnică a Moldovei**

**DEZVOLTAREA UNEI APLICAȚII de Recunoastere faciala**

**DEVELOPMENT OF AN application for facial recognition**

|  |  |
| --- | --- |
| **Studenții:** | **gr. TI-231,**  **Maximov Dan**  **Duca Nicolae**  **Mereacre Pavel** |
| **Coordonator:** | **Coseru Catalin**  **,asist.univ.** |

**Chişinău, 2024**

**CUPRINS**

[INTRODUCERE 2](#_Toc184312921)

[Analiza domeniului de studiu 4](#_Toc184312922)

[1.1 Importanta temei 4](#_Toc184312923)

[1.2 Sisteme similare cu proiectul realizat 4](#_Toc184312924)

[1.3 Scopul, obiectivele și cerințele sistemului 9](#_Toc184312925)

[MODELAREA ȘI PROIECTAREA SISTEMULUI INFORMATIC 11](#_Toc184312926)

[2.1 Descrierea metodei Fisherface (LDA și PCA) 11](#_Toc184312927)

[2.2 Integrarea Algoritmului Genetic pentru optimizare 12](#_Toc184312928)

[IMPLEMENTAREA SISTEMULUI 13](#_Toc184312929)

[3.1 Uneltele și tehnologiile utilizate 13](#_Toc184312930)

[3.2 Dezvoltarea componentelor sistemului 13](#_Toc184312931)

[3.2.1 Preprocesarea imaginilor 13](#_Toc184312932)

[3.2.2 Implementarea PCA și LDA 14](#_Toc184312933)

[3.2.3 Optimizarea cu Algoritmul Genetic 15](#_Toc184312934)

[Concluzie 15](#_Toc184312935)

[ANEXA A 17](#_Toc184312936)

[ANEXA B 20](#_Toc184312937)

# INTRODUCERE

# Proiectul „Face ID” reprezintă o inițiativă inovatoare ce își propune să revoluționeze domeniul recunoașterii faciale, combinând două dintre cele mai promițătoare ramuri tehnologice: algoritmii genetici și inteligența artificială. Recunoașterea facială, devenită omniprezentă în viața cotidiană, joacă un rol crucial în domenii precum securitatea cibernetică, controlul accesului sau monitorizarea spațiilor publice, iar optimizarea acestui proces este esențială pentru a răspunde cerințelor din ce în ce mai complexe.

# Algoritmii genetici, inspirați din procesele naturale de selecție și evoluție, oferă un cadru flexibil și eficient pentru rezolvarea problemelor complexe, inclusiv în recunoașterea facială. Prin aplicarea acestor metode, sistemul poate analiza și selecta trăsăturile faciale cele mai relevante, eliminând redundanțele și îmbunătățind semnificativ acuratețea procesului de identificare. Această abordare poate contribui la depășirea provocărilor tradiționale, precum variațiile de expresie, iluminare sau unghiuri de captare a imaginii.

# Un element central al proiectului este utilizarea unei baze de date extinse, care conține imagini și caracteristici faciale diverse, esențiale pentru instruirea și testarea modelului propus. Acest set de date nu doar că susține procesul de dezvoltare al algoritmului, dar permite și realizarea unor analize comparative riguroase, contribuind la adaptabilitatea și robustețea soluției finale.

# Prin integrarea algoritmilor genetici într-un sistem de recunoaștere facială, proiectul „Face ID” își propune să stabilească noi standarde în ceea ce privește eficiența, scalabilitatea și precizia. Într-o lume tot mai interconectată, acest demers vine ca un răspuns direct la nevoia de soluții avansate pentru securitate și autentificare, oferind un model care poate fi aplicat într-o gamă largă de contexte, de la utilizarea personală la implementări la scară largă. Această combinație unică de tehnologii avansate reflectă potențialul imens al cercetării interdisciplinare în conturarea viitorului tehnologiei.

# Analiza domeniului de studiu

# Importanta temei

Domeniul recunoașterii faciale reprezintă o ramură interdisciplinară care îmbină elemente din inteligența artificială, viziunea computerizată, biologia computațională și securitatea cibernetică. Acest domeniu a înregistrat o creștere semnificativă în ultimii ani, fiind aplicat într-o gamă largă de industrii, inclusiv în domeniul securității, sănătății, comerțului și în dezvoltarea de tehnologii mobile. Scopul său principal este identificarea și autentificarea indivizilor prin analiza trăsăturilor faciale, iar pentru aceasta este esențială dezvoltarea unor algoritmi și modele care să permită recunoașterea rapidă și precisă.

Una dintre provocările fundamentale ale recunoașterii faciale este variabilitatea naturală a trăsăturilor faciale, influențată de factori precum expresiile faciale, lumina, unghiul de captare a imaginii și modificările de-a lungul timpului. Pentru a rezolva aceste probleme, s-au dezvoltat algoritmi de învățare automată și tehnici de procesare a imaginii care să extragă caracteristicile esențiale, minimizând variabilitatea nerelevantă. Printre aceste metode se numără algoritmii genetici, rețelele neuronale convoluționale (CNN), analiza componentelor principale (PCA) și modelarea trăsăturilor 3D, toate acestea având rolul de a îmbunătăți acuratețea identificării.

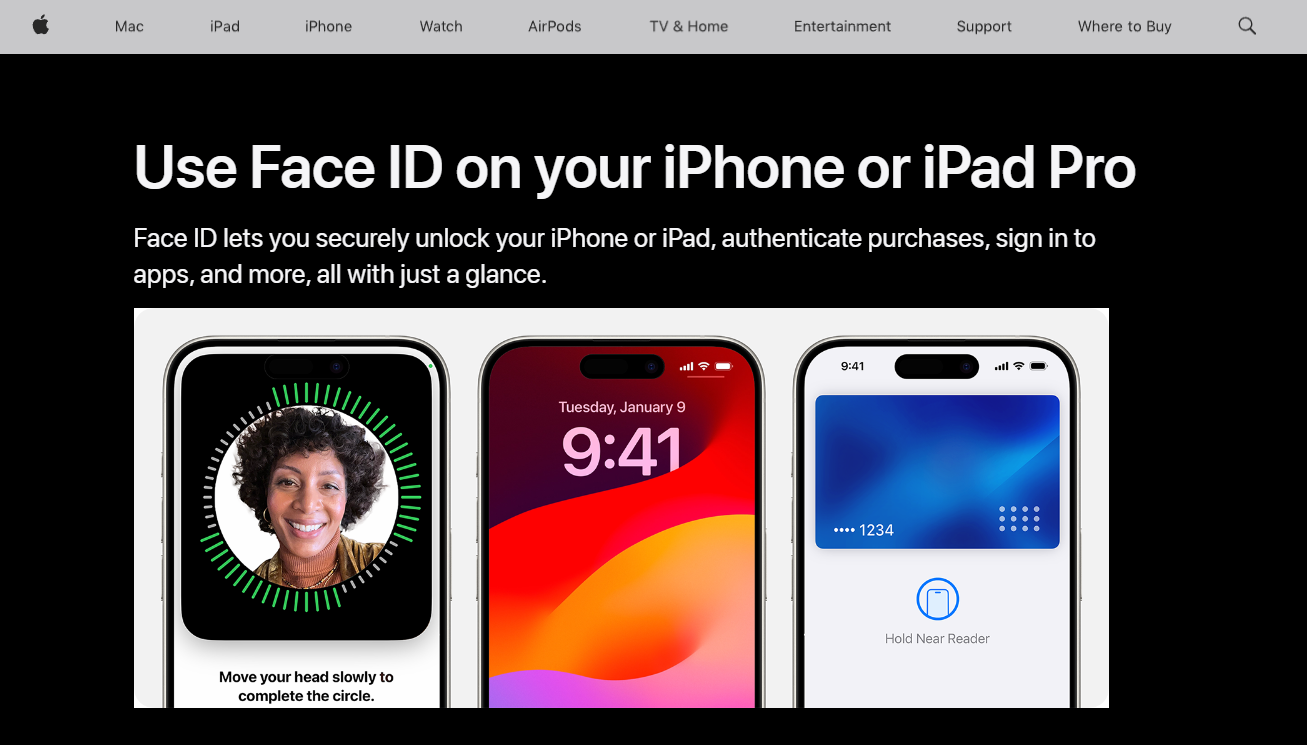
Algoritmii genetici reprezintă o metodă inovatoare în domeniul recunoașterii faciale, fiind inspirați de procesele naturale de selecție și evoluție. Aceștia permit optimizarea parametrilor modelului de recunoaștere prin selecția, încrucișarea și mutația caracteristicilor esențiale, cu scopul de a obține o soluție cât mai performantă. În contextul acestui proiect, algoritmii genetici sunt folosiți pentru a identifica cele mai relevante trăsături faciale care diferențiază indivizii în baza de date.

Un alt element crucial în recunoașterea facială este utilizarea unei baze de date bine structurate și relevante. Aceasta conține imagini faciale și atribute asociate, fiind fundamentul pentru antrenarea și validarea algoritmilor. Alegerea și organizarea corectă a datelor influențează semnificativ performanța modelului, iar accesul la date diverse și de calitate este un factor determinant în succesul aplicațiilor de recunoaștere facială.

# 1.2 Sisteme similare cu proiectul realizat

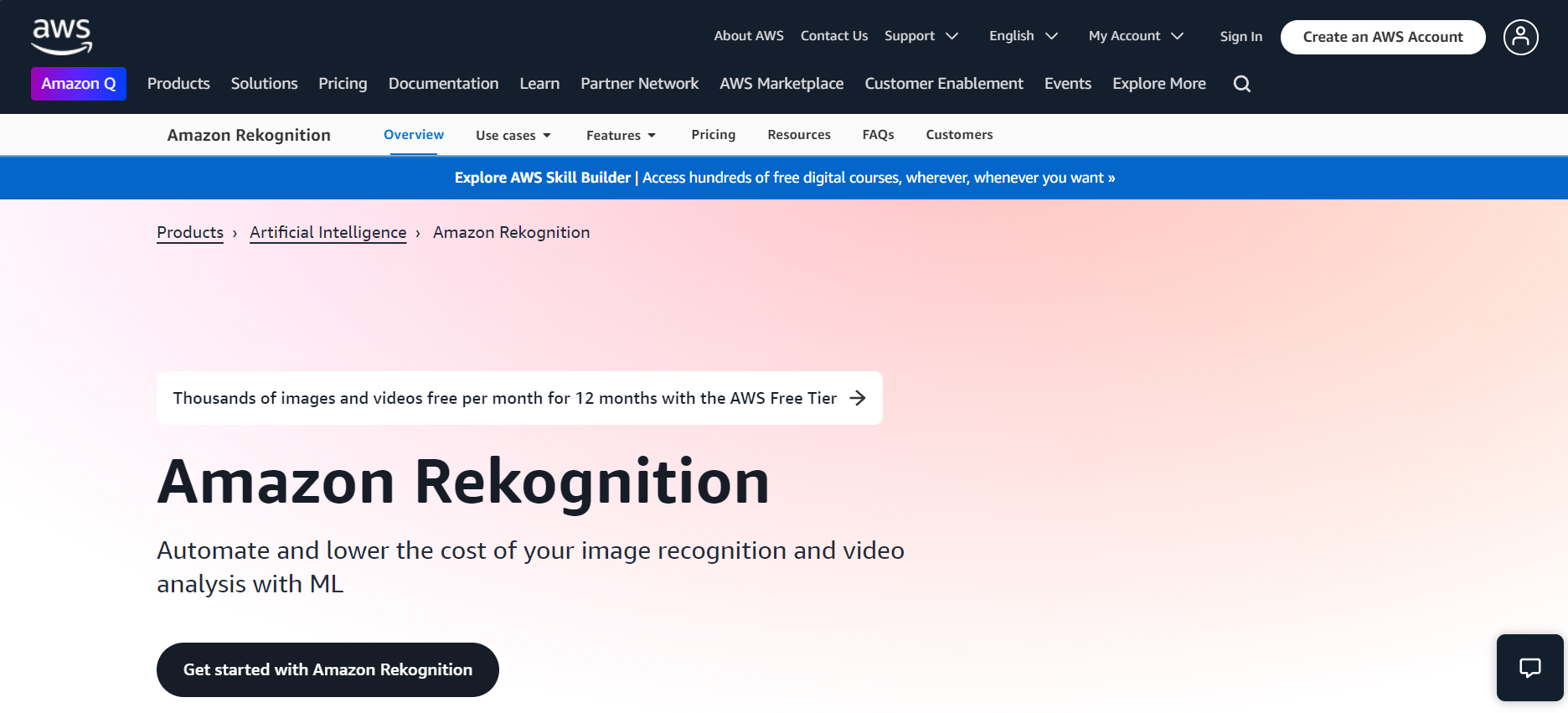
Numeroase sisteme de recunoaștere facială au fost dezvoltate și implementate în ultimii ani, cu scopuri variate, de la securitate și supraveghere la autentificarea utilizatorilor pe dispozitivele mobile. Printre acestea se numără soluțiile oferite de giganții tehnologici precum Apple, Google și Amazon, precum și alte sisteme specializate folosite în sectoare guvernamentale și de securitate. În continuare, vom analiza câteva dintre cele mai cunoscute sisteme similare care aplică principii avansate de recunoaștere facială, unele dintre ele incluzând elemente care se regăsesc și în proiectul „Face ID” bazat pe algoritmi genetici.

**Apple Face ID**   
Apple a introdus Face ID pe dispozitivele iPhone ca o metodă biometrică de autentificare. Folosind senzori 3D și o rețea neuronală, Face ID identifică și recunoaște trăsăturile faciale ale utilizatorului chiar și în condiții de iluminare slabă sau din unghiuri diferite. Algoritmii învață și se adaptează la schimbările feței în timp, cum ar fi purtarea de ochelari sau modificările de expresie, păstrând un nivel ridicat de acuratețe și siguranță. În timp ce Apple folosește în principal rețele neuronale pentru procesarea imaginilor faciale, sistemul „Face ID” din acest proiect ar încerca o abordare alternativă, utilizând algoritmi genetici pentru a optimiza parametrii de recunoaștere.



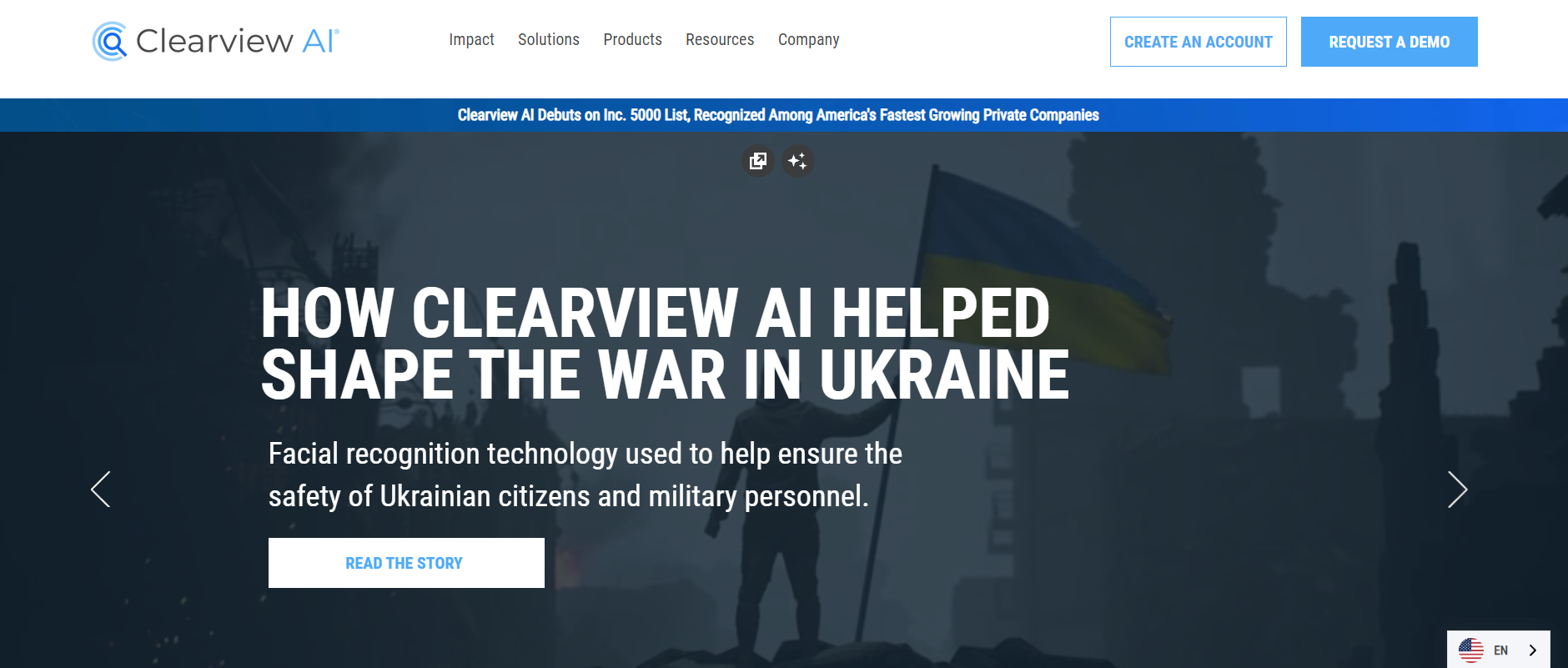
**Figura 1.1 – Interfata Web Apple Face ID**

**Amazon Rekognition**  
Amazon Rekognition este o platformă de recunoaștere facială și analiză video bazată pe inteligență artificială, utilizată în diverse domenii, inclusiv în securitate și supraveghere. Serviciul oferă recunoaștere facială, detectare a emoțiilor, recunoaștere a obiectelor și a textelor. Amazon Rekognition utilizează modele de învățare profundă antrenate pe mari cantități de date pentru a asigura o identificare precisă. Algoritmii genetici nu sunt folosiți direct în acest sistem



**Figura 1.2 – Interfata Web Amazon Rekognition**

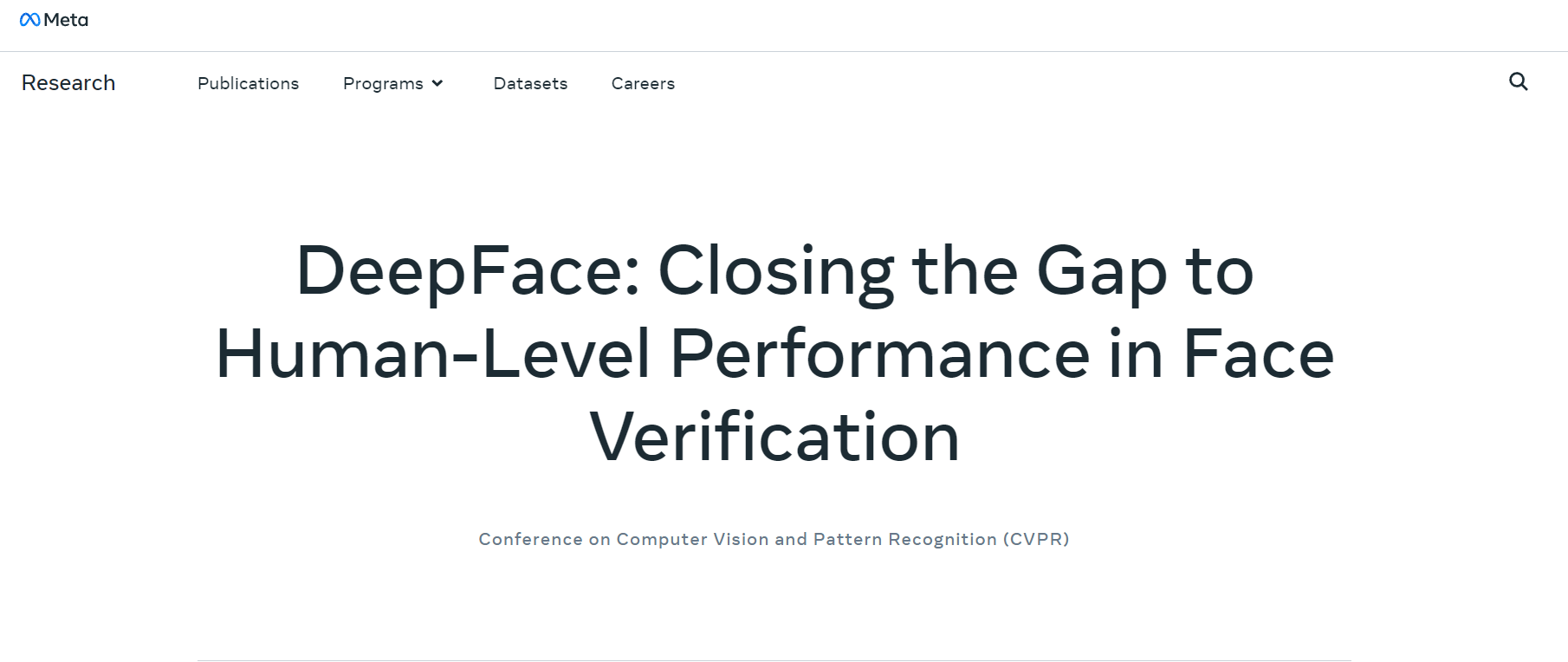
**Clearview AI**  
Clearview AI este o platformă de recunoaștere facială utilizată în mod special de agențiile de aplicare a legii. Aceasta extrage caracteristici faciale din imagini publice și le compară cu baza sa de date extinsă, care conține milioane de imagini colectate din surse online. Sistemul utilizează tehnici de învățare automată avansate pentru a analiza și a compara trăsăturile faciale, fiind capabil să identifice rapid și precis indivizii. Deși utilizarea sa este controversată din punct de vedere etic și al confidențialității, Clearview AI este un exemplu de aplicație avansată a tehnologiei de recunoaștere facială.



**Figura 1.3 – Interfata Web Clearview AI**

**Sistemele de recunoaștere facială în supravegherea publică**  
Unele orașe mari utilizează sisteme de supraveghere bazate pe recunoaștere facială pentru securitate publică și monitorizare a mulțimilor. Acestea folosesc camere și algoritmi de recunoaștere pentru a detecta și identifica indivizi în spațiile publice, contribuind la prevenirea criminalității. Exemplele includ sistemele de recunoaștere facială utilizate în Marea Britanie și China, care sunt implementate la scară largă și sunt conectate la baze de date guvernamentale pentru identificare rapidă. În aceste sisteme, algoritmii folosiți sunt de regulă din categoria învățării automate, dar integrarea algoritmilor genetici ar putea aduce un avantaj în optimizarea parametrilor și îmbunătățirea eficienței de identificare.

**DeepFace**   
DeepFace este sistemul de recunoaștere facială dezvoltat de Facebook, folosit pentru identificarea automată a persoanelor în fotografiile încărcate pe platformă. Bazat pe rețele neuronale convoluționale (CNN), DeepFace este capabil să recunoască utilizatorii cu o acuratețe similară cu a percepției umane. Acest sistem nu folosește algoritmi genetici, însă include tehnici avansate de învățare automată pentru a detecta și clasifica trăsăturile faciale, având un rol similar cu cel al „Face ID” propus în proiectul nostru.



**Figura 1.4 – Interfata Web DeepFace**

Sistemele de recunoaștere facială analizate sunt extrem de diverse și folosesc metode avansate de inteligență artificială și învățare automată. Proiectul „Face ID” propus se distinge prin aplicarea algoritmilor genetici pentru optimizarea procesului de identificare, o abordare ce poate aduce îmbunătățiri semnificative în termeni de acuratețe și eficiență. În timp ce soluțiile existente se bazează în principal pe rețele neuronale și procesare avansată a imaginilor, integrarea unei tehnici evolutive cum sunt algoritmii genetici poate oferi o alternativă inovatoare în acest domeniu în continuă dezvoltare

# 1.3 Scopul, obiectivele și cerințele sistemului

Scopul principal al proiectului „Face ID” este dezvoltarea unui sistem de recunoaștere facială performant, capabil să identifice corect și eficient persoanele dintr-o bază de date existentă. Acest sistem se bazează pe utilizarea algoritmilor genetici pentru a optimiza procesul de recunoaștere facială, maximizând astfel acuratețea și minimizând timpul de procesare. Aplicația va avea o utilitate extinsă în domeniul securității și autentificării, fiind capabilă să răspundă nevoilor unor medii diverse – de la autentificare în aplicații și dispozitive, până la utilizări mai complexe, precum securitatea publică și accesul controlat.

**Obiectivele sistemului:**

**Identificarea precisă a persoanelor**  
Sistemul trebuie să fie capabil să identifice corect indivizii pe baza trăsăturilor faciale, folosind o bază de date existentă. Algoritmii genetici vor contribui la rafinarea procesului de identificare, îmbunătățind astfel rata de acuratețe și reducând ratele de eroare.

**Optimizarea procesului de recunoaștere prin algoritmi genetici**  
Algoritmii genetici vor fi utilizați pentru a optimiza selectarea și evaluarea trăsăturilor faciale esențiale, prin procese de selecție, încrucișare și mutație. Această abordare va permite sistemului să învețe și să își îmbunătățească performanțele în mod constant, adaptându-se la noi date și condiții.

**Performanță ridicată în termeni de viteză și eficiență**  
Sistemul trebuie să fie capabil să proceseze și să analizeze imaginile în timp real sau cu o întârziere minimă. Acest lucru este esențial pentru a asigura o experiență de utilizare optimă și pentru a permite utilizarea sa în scenarii de securitate și autentificare unde viteza este critică.

**Integrarea și testarea cu o bază de date facială existentă**  
Sistemul va fi testat și optimizat pe o bază de date cu imagini faciale preexistente. Acest obiectiv va asigura o evaluare riguroasă și o validare a performanței sistemului în scenarii reale de utilizare.

**Scalabilitate și flexibilitate în utilizare**  
Sistemul trebuie să fie suficient de flexibil pentru a fi adaptat la diferite dimensiuni ale bazei de date și la diferite aplicații (de exemplu, autentificare pe dispozitive personale, sisteme de securitate pentru clădiri, supraveghere publică).

**Cerințe funcționale:**

**Capacitatea de procesare și stocare a imaginilor faciale**  
Sistemul trebuie să aibă capacitatea de a prelua, procesa și stoca imagini faciale, generând metadate relevante pentru identificarea fiecărei persoane.

**Algoritm de recunoaștere facială bazat pe algoritmi genetici**  
Sistemul va implementa un algoritm de recunoaștere facială care utilizează metode genetice pentru selectarea și rafinarea caracteristicilor faciale esențiale, necesare identificării.

**Actualizare automată a algoritmilor pe baza noilor date**  
Sistemul trebuie să fie capabil să integreze noi date și să adapteze algoritmii pe baza acestor date, prin procese de optimizare continuă.

**Interfață de utilizator pentru interacțiune și autentificare**  
Sistemul va necesita o interfață de utilizator intuitivă, prin care utilizatorii să poată interacționa și să își verifice identitatea, oferind feedback și actualizări privind procesul de identificare.

**Cerințe non-funcționale**

**Acuratețe ridicată a recunoașterii**  
Sistemul trebuie să atingă un nivel ridicat de acuratețe, minimizând ratele de eroare, atât în ceea ce privește identificările false pozitive, cât și cele negative.

**Timp de procesare optimizat**  
Sistemul trebuie să funcționeze în timp real sau cât mai aproape de acest ideal, asigurând astfel o experiență fluidă și eficientă pentru utilizator.

**Securitate și confidențialitate**  
Datele faciale și toate informațiile stocate în baza de date trebuie protejate prin măsuri de securitate solide, asigurând confidențialitatea și integritatea datelor.

**Scalabilitate.** Sistemul trebuie să fie suficient de flexibil pentru a se adapta unui volum crescut de date și pentru a permite extinderea sa fără pierderea performanței.

**Compatibilitate cu diverse tipuri de medii hardware și software**  
Sistemul trebuie să fie compatibil și să funcționeze eficient pe diverse platforme, permițând integrarea sa în medii de utilizare variate.

# MODELAREA ȘI PROIECTAREA SISTEMULUI INFORMATIC

# 2.1 Descrierea metodei Fisherface (LDA și PCA)

Metoda **Fisherface** combină două tehnici fundamentale din analiza dimensionalității: **PCA** (Principal Component Analysis) și **LDA** (Linear Discriminant Analysis).

**PCA**: Este utilizată pentru reducerea dimensionalității setului de date. Aceasta transformă datele originale într-un nou spațiu în care varianța este maximizată. Astfel, imaginile fețelor sunt reprezentate într-un spațiu cu mai puține dimensiuni, păstrând trăsăturile relevante pentru identificare. Matematic, PCA găsește componentele principale prin calcularea vectorilor proprii ai matricei de covarianță a datelor. Aceste componente principale sunt utilizate pentru a proiecta datele într-un spațiu de dimensionalitate redusă.

**LDA**: Este aplicată ulterior pentru a maximiza separarea între clase (adică diferite persoane). LDA utilizează componentele calculate de PCA și maximizează raportul între varianța dintre clase și varianța din interiorul claselor.

LDA se bazează pe optimizarea funcției de discriminare:

A black text on a white background

Description automatically generated

unde:

SB = matricea de dispersie intre clase;

SW = matricea de dispersie in interiorul claselor;

W = matricea proiectiei;

Combinația PCA-LDA permite construirea unui spațiu optim în care diferențele între fețe sunt evidențiate, iar datele redundante sau zgomotoase sunt eliminate.

# 2.2 Integrarea Algoritmului Genetic pentru optimizare

**Algoritmul Genetic (GA)** este o metodă evolutivă care simulează procesul natural de selecție pentru a optimiza funcții complexe. În cadrul proiectului, acesta este utilizat pentru:

**Optimizarea hiperparametrilor PCA și LDA**, precum:

* Numarul de componente principale (PCA);

- Numarul de dimensiuni utilizate pentru LDA;

**Îmbunătățirea clasificatorului final**, prin alegerea unui set optim de funcții de decizie.

**Fluxul algoritmului genetic:**

* **Reprezentare:** Parametrii sunt codificați într-un vector (cromozom), de exemplu:

A black text on a white background

Description automatically generated

* **Inițializare**: Se generează o populație inițială aleatorie de soluții candidate.

# IMPLEMENTAREA SISTEMULUI

Implementarea sistemului pentru recunoașterea facială bazată pe Fisherface (PCA și LDA) și optimizată cu Algoritmul Genetic implică mai multe etape organizate logic. Acestea includ procesarea imaginilor, integrarea algoritmilor de analiză a trăsăturilor și implementarea algoritmului genetic, alături de dezvoltarea unei interfețe grafice intuitive.

# 3.1 Uneltele și tehnologiile utilizate

Pentru dezvoltarea proiectului, au fost selectate următoarele tehnologii și unelte:

**Python**: Limbaj principal de implementare, cunoscut pentru suportul bibliotecilor numerice și de procesare a imaginilor.

**Biblioteci Python**:

* **NumPy**: Operare pe matrici și vectori pentru calcul mathematic;
* **OpenCV**: Procesare a imaginilor și detectarea trăsăturilor faciale;
* **scikit-learn**: Implementarea PCA, LDA și clasificatorului kNN;
* **DEAP**: Framework pentru Algoritmi Genetici.
* **PyQt**: Dezvoltarea interfeței grafice (GUI).
* **Seturi de date**:
* **FERET** sau **LFW** pentru testare și validare.

# 3.2 Dezvoltarea componentelor sistemului

# 3.2.1 Preprocesarea imaginilor

Aceasta etapă pregătește imaginile pentru extragerea trăsăturilor:

Detectarea feței:

* Utilizarea Haar cascades din OpenCV pentru a localiza fața în imagine.
* Tăierea regiunii relevante și ignorarea fundalului.
* Conversia la scară de gri;
* Reducerea dimensionalității imaginii pentru procesare mai rapidă.
* Redimensionarea: Uniformizarea imaginilor la o dimensiune fixă (ex. 100x100 pixeli).
* Normalizarea: Scalarea valorilor pixelilor între 0 și 1 pentru a elimina efectele iluminării.

# Implementarea PCA și LDA

PCA:

* Construirea matricei de date: fiecare imagine este transformată într-un vector.
* Calculul mediei și centrarea datelor:

A black symbol with a white background

Description automatically generated

* Calculul matricei de covarianță:

A black text on a white background

Description automatically generated

* Determinarea vectorilor proprii și păstrarea celor mai relevanți ‘k’;
* Proiecția datelor în spațiul redus:x

A black text on a white background

Description automatically generated

LDA:

* Calcularea matricelor de dispersie între clase (SB​) și în interiorul claselor (SW ​).
* Determinarea vectorilor proprii ai raportului:



* Proiecția datelor în noul spațiu:

A black text on a white background

Description automatically generated

# 3.2.3 Optimizarea cu Algoritmul Genetic

**Pasul 1: Reprezentarea soluției**

Fiecare soluție este reprezentată ca un “cromozom” [kPC A,dLD A], unde:

kPC A : Numărul de componente principale selectate.

dLD A : Dimensiunea spațiului redus prin LDA.

**Pasul 2: Implementarea GA**

**Inițializarea populației:** Se generează un set de soluții aleatoare.

**Evaluarea conformitatii:**

Se calculează acuratețea clasificării pentru fiecare soluție.

Conformitatea este:



**Selecția, încrucișarea și mutația:**

Se selectează soluțiile cele mai promițătoare.

Încrucișarea combină parametrii cromozomilor pentru a genera noi soluții.

Mutația modifică aleator un parametru pentru explorare.

**Iterare:** Se continuă până când conformitatea (F) nu mai crește semnificativ.

# Implementarea extinsă, “cizelarea” aplicației pentru utilizare scalabilă

Pentru a asigura funcționarea corectă și sigură a aplicației de recunoaștere facială, este esențial să implementăm algoritmi avansați care să detecteze și să prevină potențialele amenințări. În această secțiune, vom detalia două dintre cele mai importante aspecte: detectarea tentativelor de fraudă prin deepfake și analiza mișcărilor feței pentru autentificare live.

**Detectarea Tentativelor de Fraudă**

**Detectarea de Deepfake**

**Descriere:**

Tehnologia deepfake a avansat rapid, iar imaginile și videoclipurile falsificate pot fi greu de distins de cele reale. Pentru a preveni utilizarea acestora în aplicația noastră, este necesară implementarea unui sistem robust de detecție.

**Metodă:**

**Utilizarea modelelor pre-antrenate CNN:** Modele precum ResNet sau MobileNet, deja antrenate pe seturi de date vaste, pot servi ca bază pentru detectarea deepfake. Aceste modele au arătat performanțe remarcabile în clasificarea imaginilor și pot fi adaptate la sarcina noastră specifică.

**Antrenarea unui model specific:** Pentru a obține o acuratețe mai mare, este recomandat să antrenăm un model specific pe un set de date care conține atât imagini reale, cât și deepfake. Acest lucru va permite modelului să învețe caracteristicile distinctive ale fiecărei categorii și să facă distincții mai fine.

**Analiza detaliilor fine:** Pe lângă caracteristicile globale ale imaginii, este important să analizăm și detalii fine, cum ar fi textura pielii, mișcarea părului, clipiri, precum și inconsistențe în lumină și umbră. Aceste detalii pot oferi indicii valoroase despre autenticitatea unei imagini.

**Analiza Mișcării Feței**

**Descriere:**

Pentru a verifica dacă o persoană este prezentă fizic în timpul procesului de autentificare, este necesară analiza mișcărilor feței în timp real.

**Metodă:**

**Detectarea caracteristicilor faciale:** Folosind biblioteci precum OpenCV, putem detecta și urmări în timp real caracteristici faciale cheie, cum ar fi ochii, gura și sprâncenele.

**Analiza mișcărilor:** Prin compararea poziției și dimensiunii acestor caracteristici în cadre consecutive, putem detecta mișcări precum clipiri, zâmbete și înclinări ale capului.

**Crearea unui sistem de detectare a anomaliilor:** Pe baza unei analize statistice a mișcărilor feței, putem stabili un set de reguli pentru a identifica mișcări anormale sau nefirești, care ar putea indica prezența unui deepfake sau a unei fotografii statice.

**Alte Considerații**

**Combinarea mai multor metode:** Pentru a îmbunătăți acuratețea și robustețea sistemului, este recomandat să combinăm mai multe metode de detecție, cum ar fi analiza spectrală, analiza texturii și analiza biometrică multi-modală.

**Actualizare continuă a modelelor:** Tehnologia deepfake evoluează rapid, astfel încât modelele noastre trebuie să fie actualizate periodic pentru a rămâne eficiente.

# 3.3.5 Securitatea și Confidențialitatea Datelor

**Protecția Datelor**

**Hashing-ul Fețelor**

**Descriere:**

Pentru a proteja confidențialitatea utilizatorilor, în loc să stocăm direct imaginile faciale, vom converti reprezentările numerice ale acestora (descriptorii faciali) în valori hash. Un hash este o reprezentare unică și ireversibilă a unui set de date. Astfel, chiar dacă baza de date ar fi compromisă, atacatorii nu ar putea reconstrui imaginile originale din valorile hash.

**Metodă:**

**Utilizarea algoritmului SHA-256:** SHA-256 este un algoritm de hashing criptografic robust și larg utilizat, care produce o amprentă digitală de 256 de biți pentru orice set de date de intrare.

**Stocarea descriptorilor în bază de date cu valori hash securizate:** Descriptorii faciali, obținuți prin aplicarea unui model de extragere a caracteristicilor (de exemplu, VGGFace, FaceNet), vor fi transformați în valori hash folosind SHA-256. Aceste valori hash vor fi apoi stocate într-o bază de date securizată, împreună cu alte informații relevante despre utilizator (de exemplu, ID-ul utilizatorului).

**Utilizarea OAuth2**

**Descriere:**

OAuth2 este un protocol de autorizare deschis, care permite aplicațiilor terțe să obțină acces limitat la resursele unui utilizator fără a necesita parola acestuia. În contextul nostru, OAuth2 va fi utilizat pentru a gestiona autentificarea utilizatorilor și pentru a controla accesul la API-urile aplicației noastre.

**Metodă:**

**Integrarea Flask-JWT:** Flask-JWT este o extensie pentru framework-ul Flask care facilitează implementarea autentificării bazate pe JSON Web Tokens (JWT). JWT-urile sunt token-uri auto-conținute și semnate digital, care conțin o încărcătură utilă JSON.

**Generarea și validarea token-urilor JWT:**

Când un utilizator se autentifică cu succes, aplicația noastră va genera un token JWT și îl va trimite înapoi către client.

Token-ul JWT va conține informații despre utilizator (de exemplu, ID-ul utilizatorului, rolul) și va fi semnat cu o cheie secretă.

La fiecare solicitare către API, clientul va trimite token-ul JWT în header-ul autorizării.

Aplicația noastră va valida semnătura token-ului și va extrage informațiile despre utilizator pentru a autoriza accesul la resurse.

**Avantajele acestei abordări**

**Securitate sporită:** Hashing-ul fețelor și utilizarea OAuth2 contribuie la protejarea confidențialității datelor utilizatorilor, reducând riscul de breșe de securitate și de furt de identitate.

**Flexibilitate:** OAuth2 permite o gestionare flexibilă a accesului la resurse, permițând acordarea de diferite niveluri de permisiuni pentru diferiți utilizatori.

**Interoperabilitate:** OAuth2 este un standard deschis, ceea ce facilitează integrarea cu alte sisteme și servicii.

**Considerații suplimentare**

**Atacuri de tip rainbow table:** Pentru a preveni atacurile de tip rainbow table, este recomandat să sărați valorile hash înainte de a le stoca.

**Rotația cheilor:** Cheia secretă utilizată pentru semnarea token-urilor JWT ar trebui să fie rotită periodic pentru a reduce riscul de compromitere.

**Protecția împotriva atacurilor de tip brute-force:** Pentru a proteja împotriva atacurilor de tip brute-force, ar trebui implementate mecanisme de limitare a ratei de autentificare.

**Confidențialitatea datelor biometrice:** Este important să se respecte reglementările privind protecția datelor cu caracter personal atunci când se colectează și se stochează date biometrice.

Prin implementarea acestor măsuri de securitate, putem construi un sistem de recunoaștere facială care protejează confidențialitatea utilizatorilor și respectă cele mai bune practici în domeniul securității cibernetice.

# Concluzie

În concluzie, proiectul „Face ID” reprezintă o inovație semnificativă în domeniul tehnologiilor de recunoaștere facială, oferind un sistem care îmbină inteligența artificială cu puterea algoritmilor genetici pentru a îmbunătăți performanța procesului de identificare. Utilizarea algoritmilor genetici permite selecția și rafinarea caracteristicilor faciale esențiale, asigurând o adaptare continuă a sistemului la schimbările mediului operațional și la complexitatea datelor procesate. Această capacitate de învățare și evoluție face din „Face ID” o soluție robustă și versatilă, capabilă să răspundă provocărilor specifice aplicării în scenarii dinamice și solicitante.

Prin optimizarea preciziei și vitezei de identificare, sistemul oferă o platformă scalabilă, potrivită pentru aplicații variate în domenii precum securitatea publică, controlul accesului, autentificarea electronică sau supravegherea inteligentă. Integrarea unei baze de date faciale extinse și diversificate contribuie la îmbunătățirea continuă a performanței, reducând riscul de erori și asigurând o recunoaștere facială precisă chiar și în condiții dificile.

De asemenea, proiectul „Face ID” adresează în mod eficient preocupările legate de confidențialitatea datelor și respectarea normelor etice. Prin respectarea standardelor înalte de protecție a datelor personale, sistemul poate câștiga încrederea utilizatorilor și a organizațiilor care îl implementează. Acest aspect este crucial în contextul global actual, în care cerințele privind securitatea cibernetică și protecția vieții private sunt din ce în ce mai stricte.

Astfel, proiectul deschide noi perspective pentru utilizarea tehnologiei de recunoaștere facială într-o manieră sigură, eficientă și etică. Prin capacitatea sa de a învăța, de a evolua și de a se adapta la cerințele tehnologice în continuă schimbare, „Face ID” contribuie semnificativ la dezvoltarea tehnologiilor biometrice. În plus, oferă o bază solidă pentru extinderi viitoare, având potențialul de a deveni un standard în industria recunoașterii faciale. Această combinație între inovație, performanță și respect pentru confidențialitate îl poziționează drept un element esențial pentru viitorul tehnologiilor de autentificare și identificare biometrică.

# Bibliografie

1. Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.
2. Duda, R. O., Hart, P. E., Stork, D. G. *Pattern Classification*. Wiley, 2001.
3. Jain, A. K., Duin, R. P. W., Mao, J. *Statistical Pattern Recognition: A Review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.
4. Turk, M., Pentland, A. *Eigenfaces for Recognition*. Journal of Cognitive Neuroscience, 1991.
5. Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
6. OpenCV Documentation. Disponibil la: https://docs.opencv.org.
7. scikit-learn Documentation. Disponibil la: <https://scikit-learn.org>.
8. Face Recognition Library GitHub Repository. Disponibil la: <https://github.com/ageitgey/face_recognition>.
9. FERET Dataset. Disponibil la: <https://www.nist.gov/itl/iad/image-group/feret-database>.
10. *Deep Learning for Computer Vision with Python*. PyImageSearch, 2017.

# ANEXA A

from flask import Flask, request, jsonify

from flask\_cors import CORS

import mysql.connector

import face\_recognition

import numpy as np

import random

import os

class Database:

def \_\_init\_\_(self):

self.connection = mysql.connector.connect(

host="localhost",

user="root",

password="Movileanu123!",

database="face\_recognition"

)

self.cursor = self.connection.cursor()

def query(self, query, params=None):

self.cursor.execute(query, params)

return self.cursor.fetchall()

def execute(self, query, params=None):

self.cursor.execute(query, params)

self.connection.commit()

class FaceRecognitionService:

def \_\_init\_\_(self, db):

self.db = db

def recognize\_face(self, face\_data):

known\_faces = self.get\_known\_faces()

matches = face\_recognition.compare\_faces(known\_faces, np.array(face\_data))

if True in matches:

match\_index = matches.index(True)

user = self.get\_user\_by\_face(known\_faces[match\_index])

return {"message": f"Welcome back, {user[0]}!"}

else:

optimized\_data = self.optimize\_face\_data(face\_data)

self.add\_new\_user("New User", optimized\_data)

return {"message": "New user created!"}

def get\_known\_faces(self):

results = self.db.query("SELECT face\_data FROM users")

return [np.frombuffer(row[0], dtype='float32') for row in results]

def get\_user\_by\_face(self, face\_data):

return self.db.query("SELECT name FROM users WHERE face\_data = %s", (face\_data.tobytes(),))

def add\_new\_user(self, name, face\_data):

self.db.execute("INSERT INTO users (name, face\_data) VALUES (%s, %s)", (name, np.array(face\_data).tobytes()))

def optimize\_face\_data(self, face\_data):

data = face\_data[0]

optimized\_data = self.optimize\_with\_random(data)

optimized\_data = self.optimize\_with\_gaussian\_noise(optimized\_data)

optimized\_data = self.optimize\_with\_genetic(optimized\_data)

return [optimized\_data]

def optimize\_with\_random(self, data):

# Amestecare aleatorie a datelor

random.shuffle(data)

return data

def optimize\_with\_gaussian\_noise(self, data):

# Adăugare de zgomot Gaussian

noise = np.random.normal(0, 0.01, len(data))

return data + noise

def optimize\_with\_genetic(self, data):

# Algoritm genetic

population = [data + np.random.normal(0, 0.01, len(data)) for \_ in range(10)]

for \_ in range(10):

# Selectarea celor mai buni indivizi

population = sorted(population, key=lambda x: -np.linalg.norm(x - data))

new\_population = population[:2] # Elitism: păstrarea celor mai buni 2 indivizi

for \_ in range(8):

parent1, parent2 = random.sample(new\_population, 2)

# Încrucișare (crossover)

child = 0.5 \* (parent1 + parent2) + np.random.normal(0, 0.01, len(data))

new\_population.append(child)

population = new\_population

# Selectarea celui mai bun individ

best\_match = max(population, key=lambda x: -np.linalg.norm(x - data))

return best\_match

app = Flask(\_\_name\_\_)

CORS(app)

db = Database()

face\_recognition\_service = FaceRecognitionService(db)

@app.route('/recognize', methods=['POST'])

def recognize():

data = request.json['data']

response = face\_recognition\_service.recognize\_face(data)

return jsonify(response)

host = os.environ.get('HOST', 'localhost')

print(f"Host: {host}")

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

# ANEXA B

import React, { useEffect, useRef, useState } from 'react';

import axios from 'axios';

import \* as faceapi from 'face-api.js';

function App() {

const videoRef = useRef(null);

const [response, setResponse] = useState(null);

useEffect(() => {

const loadModels = async () => {

const MODEL\_URL = '/models';

await faceapi.nets.tinyFaceDetector.loadFromUri(MODEL\_URL);

await faceapi.nets.faceLandmark68Net.loadFromUri(MODEL\_URL);

await faceapi.nets.faceRecognitionNet.loadFromUri(MODEL\_URL);

startVideo();

};

const startVideo = () => {

navigator.mediaDevices.getUserMedia(

{ video: {} },

stream => (videoRef.current.srcObject = stream),

err => console.error(err)

);

};

loadModels();

}, []);

const handleVideoPlay = async () => {

const detections = await faceapi.detectAllFaces(

videoRef.current,

new faceapi.TinyFaceDetectorOptions()

).withFaceLandmarks().withFaceDescriptors();

if (detections.length > 0) {

const data = detections.map(d => d.descriptor);

axios.post('http://localhost:5000/recognize', { data })

.then(res => setResponse(res.data))

.catch(err => console.error(err));

}

};

return (

<div className="App">

<h1>Face Recognition App</h1>

<video ref={videoRef} onPlay={handleVideoPlay} autoPlay muted width="720" height="560" />

{response && <div>{response.message}</div>}

</div>

);

}

export default App;