

# DETECTIA ȘI RECUNOAȘTEREA FETELOR

1<sup>st</sup> Martin Maria  
Universitatea Tehnică  
Gheorghe Asachi din Iași  
Iași, România  
maria.martin@student.tuiasi.ro

2<sup>nd</sup> Poclid Ionuț-Andrei  
Universitatea Tehnică  
Gheorghe Asachi din Iași  
Iași, România  
ionut-andrei.poclid@student.tuiasi.ro

## I. INTRODUCERE

Detectarea facială se referă la o tehnologie computerizată care este capabilă să identifice fețele umane în imagini digitale. Primii algoritmi de recunoaștere facială au fost dezvoltati la începutul anilor șaptezeci. De atunci, acuratețea lor s-a îmbunătățit până la punctul pe care îl cunoștem în zilele noastre.

Algoritmii de detectare a feței pun accent pe detectarea fețelor umane în imagini care pot conține și obiecte, alte părți umane, peisaje, animale. Procesul de detectare a feței este un pas esențial în detectarea și localizarea fețelor umane în imagini și videoclipuri. Procesul de captare a feței transformă informații analogice (o față) într-un set de informații digitale (date sau vectori) bazate pe trăsăturile faciale ale persoanei. Procesul de potrivire a feței verifică dacă două fețe aparțin aceleiași persoane.

Sistemele moderne de recunoaștere a feței necesită doar ca utilizatorii să se afle în câmpul vizual al unei camere, cu condiția ca aceștia să se afle la o distanță rezonabilă de cameră. Acest lucru face ca recunoașterea feței să fie cea mai accesibilă modalitate biometrică.

Sistemele de recunoaștere a feței sunt de obicei compuse din:

- Detectare facială: Un detector de fețe găsește poziția fețelor într-o imagine și (dacă există) returnează coordonatele unei casete de delimitare pentru fiecare dintre ele.
- Reprezentarea feței: În etapa de reprezentare a feței, valorile pixelilor unei imagini ale feței sunt transformate într-un vector de caracteristici compact, șablon. Toate fețele aceluiași subiect ar trebui mapate la vectori de caracteristici similare.
- Potrivirea feței: Două șabloane sunt comparate pentru a produce un scor de similaritate care indică probabilitatea ca acestea să aparțină aceluiași subiect.

## II. LOAD ENCODING IMAGE

Un codificator scrie date dintr-o imagine într-un flux. Codificatoarele pot comprima, cripta și modifica pixelii imaginii în mai multe moduri înainte de a le scrie în fluxul de date.

Utilizarea unor codificatoare are ca rezultat compromisuri, de exemplu, JPEG, care schimbă informațiile de culoare pentru o compresie mai bună. Alte codificatoare nu duc la astfel de pierderi, de exemplu, bitmap (BMP).

Deoarece multe codecuri folosesc tehnologie proprie pentru a obține o exactitate mai bună a imaginii, detaliile despre

modul în care o imagine este codificată sunt la latitudinea dezvoltatorului de codec.

O aplicație se poate aștepta să întâlnească următorul set de opțiuni de codificator:

Property Name	VARTYPE	Value	Applicable Codecs
ImageQuality	VT_R4	0-1.0	JPEG, HDPhoto
CompressionQuality	VT_R4	0-1.0	TIFF
Lossless	VT_BOOL	TRUE, FALSE	HDPhoto
BitmapTransform	VT_UI1	<a href="#">WICBitmapTransformOptions</a>	JPEG

ImageQuality de 0.0 înseamnă cea mai scăzută redare a fidelității posibile, iar 1.0 înseamnă cea mai mare fidelitate, ceea ce poate implica, de asemenea, fără pierderi, în funcție de codec.

ImageQuality de 0.0 înseamnă cea mai scăzută redare a fidelității posibile, iar 1.0 înseamnă cea mai mare fidelitate, ceea ce poate implica, de asemenea, fără pierderi, în funcție de codec.

CompressionQuality de 0.0 înseamnă cea mai puțin eficientă schemă de compresie disponibilă, rezultând de obicei o codificare rapidă, dar o ieșire mai mare. O valoare de 1.0 înseamnă cea mai eficientă schemă disponibilă, de obicei durează mai mult timp pentru a codifica, dar produce o ieșire mai mică. În funcție de capacitățile codecului, acest interval poate fi mapat pe un set discret de metode de compresie disponibile.

## III. OPENCV LIBRARY

OpenCV (Open Source Computer Vision) este o bibliotecă open-source care include câteva sute de algoritmi de computer vision.

OpenCV are o structură modulară, ceea ce înseamnă că pachetul include mai multe biblioteci partajate sau statice.

Ea cuprinde următoarele module:

1. Funcționalitate de bază (core) - un modul compact care definește structurile de date de bază, inclusiv matricea densă multidimensională Mat și funcțiile de bază utilizate de toate celelalte module.
2. Procesarea imaginii (imgproc) - un modul de procesare a

imaginii care include filtrarea liniară și neliniară a imaginii, transformări geometrice ale imaginii, conversia spațiului de culoare.

3. Analiză video (video) - un modul de analiză video care include estimarea mișcării, scăderea fundalului și algoritmi de urmărire a obiectelor.

4. Calibrarea camerei și reconstrucția 3D (calib3d) - algoritmi de bază de geometrie cu vizualizări multiple, calibrarea camerei unice și stereo, estimarea poziției obiectului, algoritmi de corespondență stereo și elemente de reconstrucție 3D.

5. Cadrul de caracteristici 2D (features2d) - detectoare de caracteristici importante, descriptori și potriviri de descriptori.

6. Detectarea obiectelor (objdetect) - detectarea obiectelor și a instanțelor claselor predefinite (de exemplu, fețe, ochi, câni, oameni, mașini).

7. Interfață de nivel înalt (highgui) - o interfață ușor de utilizat pentru capabilități simple de UI.

8. Video I/O (videoio) - o interfață ușor de utilizat pentru captarea video.

#### IV. PYQT5 LIBRARY

PyQt este o bibliotecă Python pentru crearea de interfețe grafice folosind setul de instrumente Qt. Creat de Riverbank Computing, PyQt este un software gratuit (cu licență GPL) și este în dezvoltare din 1999. PyQt5 a fost lansat în 2016 și ultima actualizare în octombrie 2021.

PyQt implementează aproximativ 440 de clase și peste 6.000 de funcții și metode.

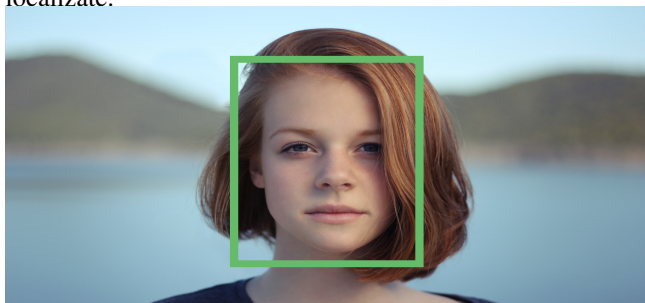
#### V. FACE RECOGNITION LIBRARY

Biblioteca Python face\_recognition a fost creată folosind recunoașterea facială de ultimă generație de la dlib, construită cu deep learning. Modelul are o precizie de 99,38% pe benchmark-ul Labeled Faces in the Wild. Aceasta oferă, de asemenea, un instrument simplu de linie de comandă face\_recognition care vă permite să faceți recunoașterea feței pe un folder de imagini din linia de comandă!

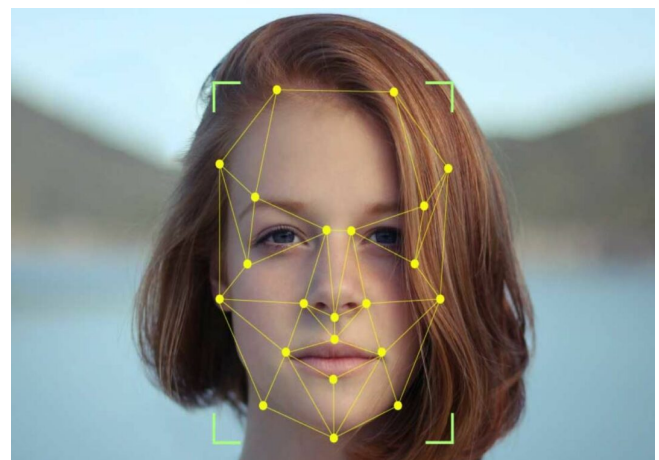
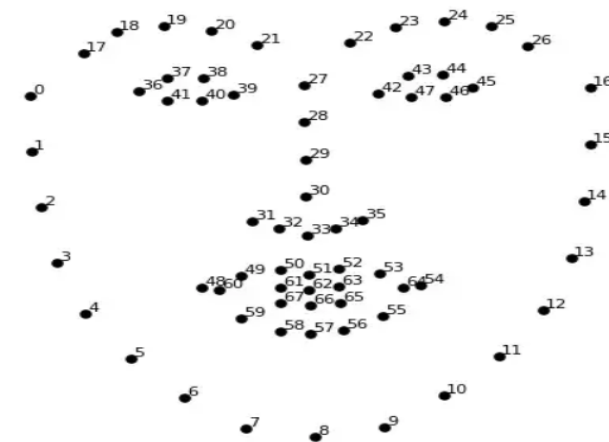
Detectarea feței a devenit populară la începutul anilor 2000, când Paul Viola și Michael Jones au inventat o modalitate de a detecta fețele suficient de rapidă pentru a rula pe camere ieftine. Cu toate acestea, acum există soluții mult mai fiabile.

Pașii pe care recunoașterea facială îi urmărește:

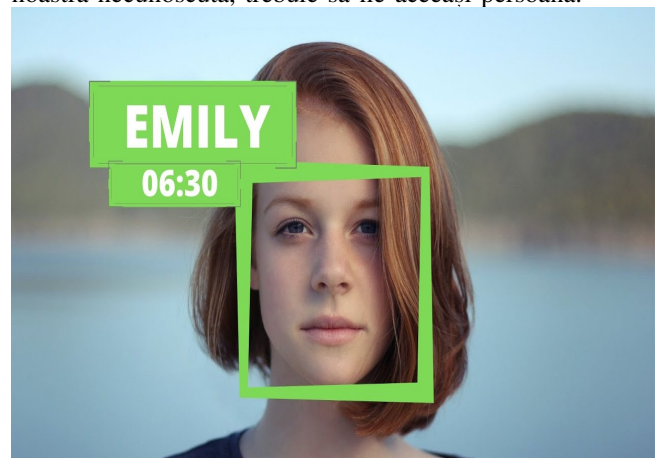
1. Găsirea tuturor fețelor: Primul pas este detectarea feței. Fețele dintr-o fotografie, pentru a le putea deosebi, trebuie localizate.



2. Proiectarea fețelor: Pentru a ține seama de acest lucru, vom încerca să deformăm fiecare imagine, astfel încât ochii și buzele să fie întotdeauna în locul eșantionului din imagine.



3. Codificarea fețelor: Cea mai simplă abordare a recunoașterii feței este să comparăm direct chipul necunoscut pe care l-am găsit la Pasul 2 cu toate imaginile pe care le avem cu persoane care au fost deja etichetate. Când găsim o față etichetată anterior care arată foarte asemănătoare cu fața noastră necunoscută, trebuie să fie aceeași persoană.



## VI. SCOPUL APLICAȚIEI

Scopul aplicației este de a realiza un sistem de recunoaștere facială capabil să detecteze și să verifice identitatea unei persoane dintr-o înregistrare video în timp real. Aceasta va prelua caracteristicile faciale ale persoanei vizate și le va compara cu cele existente în sistemul aplicației. Se va contoriza timpul în care o persoană este prezentă în înregistrare și rezultatul se va trece într-un fișier excel, precum și alte date referitoare la momentul identificării. La finalul proiectului, prin intermediul aplicației realizate vom putea ține o evidență a timpului pe care o persoană (sau mai multe) l-a petrecut într-o locație. Un astfel de sistem poate fi utilizat într-un loc public, de exemplu implementarea aplicației în camerele de luat vederi dintr-o școală pentru a ține evidența prezentei elevilor la ore, sau privat, de exemplu urmărirea programului unor angajați din cadrul unei companii.

## VII. ANALIZĂ DE PIAȚĂ

Potrivit unui articol publicat de Mordor Intelligence, piața recunoașterii faciale a fost evaluată la 3,72 miliarde USD în 2020 și se estimează că va fi evaluată la 11,62 miliarde USD până în 2026, înregistrând o rată anuală de creștere compusă de aproximativ 21,71% în perioada prognozată (2021 – 2026). De asemenea, China este unul dintre principalii consumatori și exportatori de tehnologie de recunoaștere facială deoarece diferite instituții guvernamentale din țară au adoptat tehnologii de supraveghere. În prezent, China are cel mai extins sistem de supraveghere publică. Potrivit Comparitech, în China, numărul camerelor CCTV variază de la 200 milioane la 626 milioane până în 2020. Potrivit Fortune Business Insights, în anul 2019, industria sistemelor de recunoaștere facială era estimată la 4,35 miliarde de dolari și se estimează ca valoarea să se tripleze până în anul 2027, bazat pe cererea ridicată și gradul imens de adopție.

## VIII. STATE OF THE ART AND RELATED WORK

Folosind date sintetice, putem reduce dimensiunea seturilor de date din lumea reală cu 75% pentru recunoașterea feței și cu 50% pentru detectarea reperelor faciale, menținând în același timp performanța.

Începând cu anii '70, recunoașterea feței a devenit unul dintre cele mai cercetate subiecte în viziunea computerizată și a biometriei. Metodele tradiționale bazate pe caracteristici realizate manual și tehnici tradiționale de învățare automată au fost înlocuite recent de rețele neuronale profunde antrenate cu seturi de date foarte mari. Metodele populare de recunoaștere a feței includ metode tradiționale (bazate pe geometrie, holistice, bazate pe caracteristici și hibride) și metode de învăț, are profundă.

Metodele de învățare profundă au devenit de ultimă generație pentru rezolvarea sarcinilor precum recunoașterea facială (FR). Deși a prezentat un mare succes, aceste modele de învățare sunt expuse intrărilor adverse, eronate, imagini la care se adaugă zgomot, limitând astfel adoptarea lor în aplicațiile din lumea reală.

### A. *AI-Based Techniques for Real-Time Face Recognition-based Attendance System- A comparative Study*

În acest articol ne sunt prezentate succint cele mai utilizate tehnologii pentru tema aleasă de noi: Cascada Haar și YOLO pentru partea de detecție, iar pentru recunoaștere avem : Modelul Binar Local și Biblioteca de recunoaștere a feței. Cascada Haar: Biblioteca OpenCV ne ajută să identificăm regiunea feței, poziția acesteia ne va ajuta în identificarea persoanei și poate diferenția oamenii folosind matricele feței lor. "YOLO": descompune imaginea care a fost luată ca intrare în celulele grilei, iar preluarea lor ne va ajuta la calcularea coordonatelor pentru locația feței. LBPH(Local Binary Pattern Histogram): se observă că bitul extrage structura imaginii, făcând diferențierea din imagine prin părțile locale. Acesta extrage două modele pentru fiecare parte locală. Biblioteca a fost creată de Adam GitGray și folosește standardul tehnicii Dlib.

### B. *Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms*

În acest articol este prezentat un studiu în care se măsoară acuratețea identificării feței pentru un grup internațional de examinatori faciali criminalistici profesioniști care lucrează în circumstanțe care se aplică în cazurile din lumea reală. Sunt prezentate date care compară tehnologia de recunoaștere a feței de ultimă generație cu cei mai buni identificatori ai feței umane.

Cu toate acestea, se susține ideea ca identificarea optimă a feței a fost obținută numai atunci când oamenii și mașinile au lucrat în colaborare.

Într-o comparație amplă a identificării feței de către oameni și computere, s-a constatat că examinatorii faciali criminalistici au fost mai precisi decât examinatorii de amprente și studenții la un test de identificare a feței.

În cadrul aceluiași test, patru rețele neuronale convoluționale profunde (DCNN), dezvoltate între 2015 și 2017, au identificat fețe în intervalul de precizie umană. Precizia algoritmilor a crescut constant de-a lungul timpului, cel mai recent punctaj DCNN având peste mediana examinatorilor faciali criminalistici.

Folosind metode de crowd-sourcing, au fost unite judecățile mai multor examinatori faciali criminalistici, făcând o medie a judecăților lor de identitate bazate pe rating, rezultând o precizie mai bună pentru judecățile combinate decât pentru persoanele care lucrau singure.

În concluzie, colaborarea între oameni sau între oameni și mașini oferă beneficii concrete pentru a face față preciziei identificării în aplicații importante.

### C. *True Key*

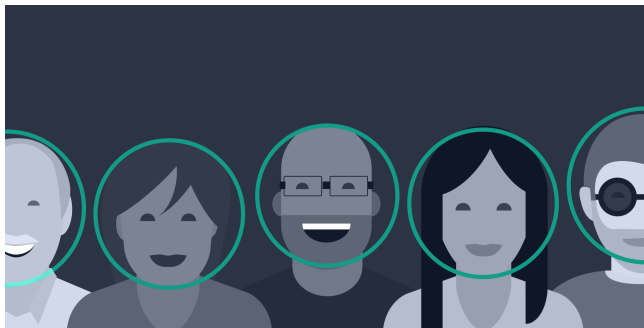
True Key reprezintă o aplicație disponibilă pe sistemul de operare Windows. Rolul acestei aplicații este de a ne conecta la diverse conturi fără să fim nevoiți să introducem parole. Conectarea se va realiza cu datele biometrice (amprenta sau recunoaștere facială). Aplicația True Key pune accent pe un nivel ridicat de securitate, autentificarea în doi factori fiind o

cerință standard, deci va trebui să se parcurgă cel puțin două nivele de autentificare înainte de conectarea dorită.

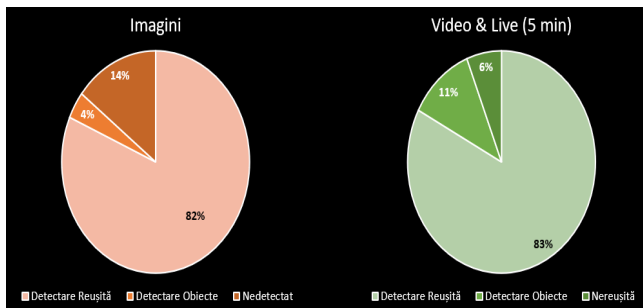
#### D. Tehnologia Face ID

Face ID este o soluție implementată de compania Apple în 2017, ce reprezintă o tehnologie modernă de autorizare biometrică a utilizatorilor. La baza acestei soluții se află recunoașterea facială. În general, este folosită pentru a debloca ecranul dispozitivului. De asemenea, se poate efectua conectarea la diverse servicii sau efectuarea plăților folosind un sistem de plată dedicat. Datorită acesteia, nicio persoană neautorizată nu poate accesa informațiile stocate pe smartphone. Scanarea feței efectuată de Face ID este foarte precisă și are un format 3D, care minimizează posibilitatea de fraudă. Smartphone-ul nu poate fi deblocat prin apropierea unei fotografii a utilizatorului. Această tehnologie este în curs de dezvoltare.

#### IX. EXEMPLU FACE DETECTION:



#### X. REZULTATE INTERMEDIARE OBTINUTE:

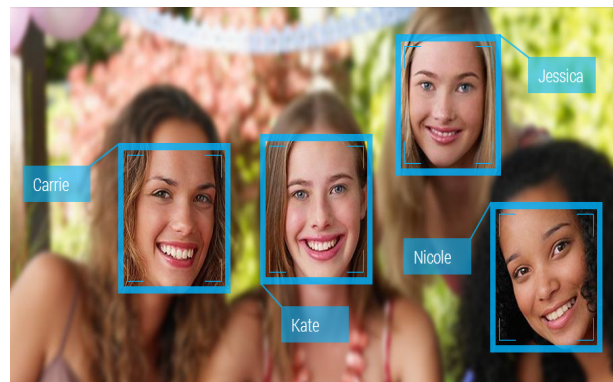


#### XI. CONCLUZII PRELIMINARE:

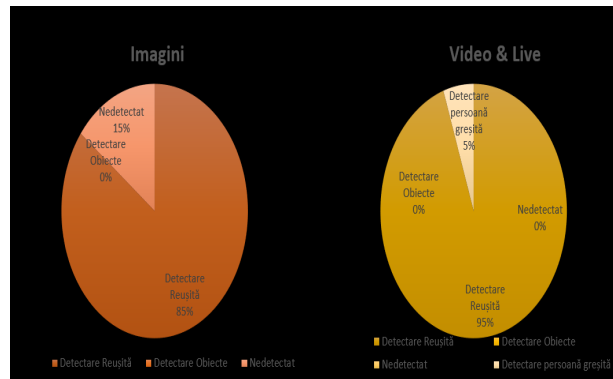
Scopul vizat în această etapă a aplicației îl constituie identificarea și localizarea cât mai precisă a indivizilor din imagini și filmări, înregistrate sau live. Pentru realizarea aplicației, a fost implementată o interfață grafică prin intermediul căreia putem încărca imaginile și filmările dorite, pe rând, și, cu ajutorul funcțiilor din librăria OpenCV, putem efectua detecția facială a indivizilor prezenți în acestea. Pentru testarea aplicației au fost folosite 5 imagini în care sunt prezente un număr diferit de persoane, înregistrarea unui meeting online și o înregistrare al unei conferințe oficiale, pentru detectarea chipurilor în fișiere deja existente, și camera web a laptop-ului, pentru detectarea chipurilor în timp real.

După colectarea datelor, conform graficelor prezentate, rezultă că detecția facială are o rată de succes mai mare în cazul imaginilor. Totodată în imaginile în care indivizii poartă accesorii sau nu au chipul complet vizibil, detecția facială este eronată, aplicația nu detectează individul sau detectează obiecte din jur. În cazul filmărilor, chipurile frontale ale indivizilor prezenți sunt detectate cu ușurință iar cele din profil sunt detectate greu sau eronat, asemanător imaginilor statice.

#### XII. EXEMPLE FACE RECOGNITION:



#### XIII. REZULTATE FINALE:



#### XIV. CONCLUZII FINALE:

Scopul aplicației este de a calcula timpul pe care o persoană îl petrece într-un anumit loc.

Pentru realizarea aplicației a fost implementată o interfață grafică cu ajutorul librăriei Qt5, prin intermediul căreia putem încărca imaginile și filmările dorite sau putem porni camera video. Prin pornirea camerei video se va începe și contorizarea timpului prezenței indivizilor.

Prin intermediul librăriei face\_recognition au fost realizate detectarea și recunoașterea facială, iar cu ajutorul bibliotecii OpenCV am realizat câteva prelucrări de imagini (conversii, redimensionări).

Pentru testarea aplicației au fost folosite 10 imagini în care sunt prezente un număr diferit de persoane, 3 fișiere video și un dispozitiv hardware, o cameră web, pentru recunoașterea

facială în timp real.

După colectarea datelor, conform graficelor finale prezentate, rezultă că recunoașterea facială are o rată de succes mai mare în cazul detectării în timp real.

Am putut observa că recunoașterea facială se realizează cu ușurință în cazul în care chipul unei persoane este complet vizibil frontal. Principalii factori care influențează funcționarea cât mai corectă a aplicației sunt lumina, focalizarea, accesoriile.

Pentru ca aplicația să funcționeze cât mai repede și cât mai eficient este nevoie de o aparatură hardware adecvată domeniului și o placă video performantă.

Utilizând o aparatură hardware dedicată se pot face mici modificări la parametrii funcțiilor din biblioteca face\_recognition pentru a obține rezultate mai bune.

## XV. MODALITATE DE LUCRU PROPUȘĂ

### Identificarea și alocarea task-urilor

#### Alocarea task-urilor

**Task 1:** Documentare: librării Python pentru prelucrare imagini și video, librării Python pentru recunoaștere facială – OpenCV: M1,M2

**Task 2:** Implementare interfața grafică: M1

**Task 3:** Implementarea unor algoritmi de detectare facială din poze, înregistrări deja existente sau în timp real: M1

**Task 4:** Implementarea unor algoritmi de recunoaștere facială din poze, înregistrări deja existente sau în timp real: M1,M2

**Task 5:** Construirea sistemului pentru datele de intrare: M1,M2

**Task 6:** Testare și fixare bug-uri: M1,M2

**Task 7:** Creare documentație și manual utilizare: M1,M2

## XVI. REFERINȚE

[1] <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>

[2] <https://docs.opencv.org/4.x/>

[3] <https://ieeexplore.ieee.org/abstract/document/8748389>

[4] <https://paperswithcode.com/paper/detection-of-face-recognition-adversarial>

[5] <https://paperswithcode.com/paper/face-recognition-from-traditional-to-deep>

[6] <https://ieeexplore.ieee.org/document/9297643>

[7] <https://www.pnas.org/doi/full/10.1073/pnas.1721355115>

[8] <https://arxiv.org/pdf/1811.00116v1.pdf>

[9] <https://learn.microsoft.com/en-us/windows/win32/wic/-wic-creating-encoder>

[10] <https://realpython.com/python-pyqt-qthread/>

[11] <https://pypi.org/project/face-recognition/>

[12] <https://openpyxl.readthedocs.io/en/stable/>

[13] <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cfc121d78>

[14] <https://www.pythonguis.com/pyqt5-tutorial/>

## XVII. GIT REPOSITORY:

<https://github.com/VedereArtificiala/prelucrareaimaginilor-proiect-martinpoclid-mss03>