Documentatie proiect:

**Capitolul 4. Prezentarea aplicației și rezultate experimentale**

**Componente interfață utilizator**

Partea practică a proiectului este o aplicație de recunoaștere automată a persoanelor construită integral în Python. Limbajul de programare Python este unul dintre cele mai utilizate limbaje de programare deoarece pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare.

Structura aplicației poate fi împărțită în două categorii: client și administrator. Fiecare persoană care utilizează una dintre cele două categorii are anumite funcționalităti ce le poate folosi.

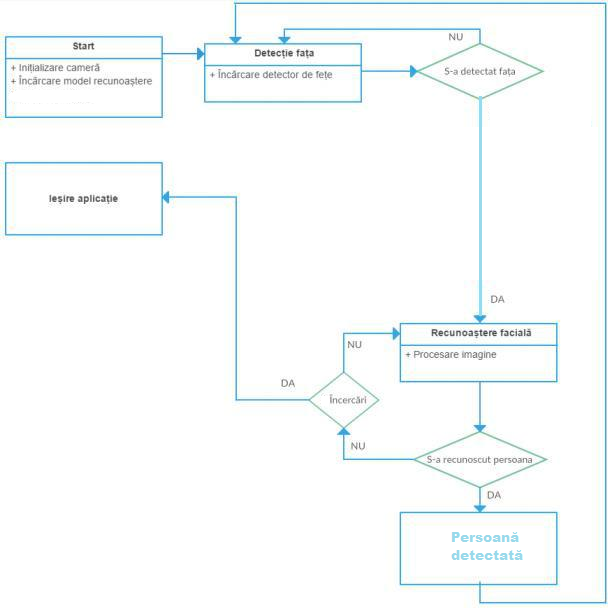


Fig.25 Organigramă sistem

Imediat ce un utilizator a pornit sistemul, acesta este întampinat de un Login Panel, fapt realizat pentru a face diferența dintre funcționalitățile unui student si ale unui administrator. Fiecare student ce utilizează această aplicație îsi poate creea un cont iar datele acestora fiind salvate într-o bază de date.

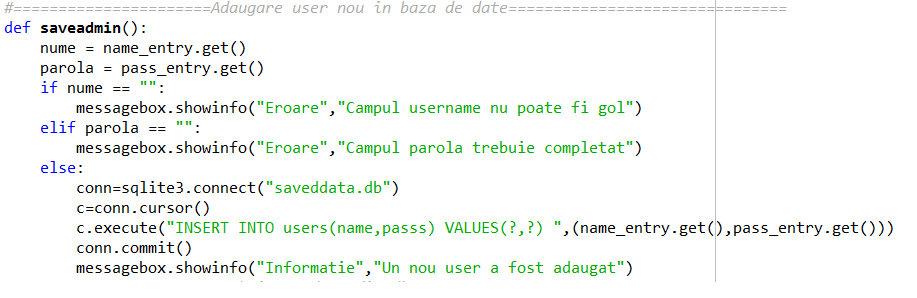


Fig.26 Adăugare user nou in baza de date

Încă de la începutul aplicatiei când utilizatorul porneste programul și este întâmpinat de Login Panel, se face separarea dintre student si administrator, fiecare dintre acestia având buton separat pentru logare.

Imediat ce un utilizator s-a logat în sistemul de management acesta are anumite funcționalități ce le poate folosi precum :

* **Înscrierea studentului** – unde studentul are de completat anumite câmpuri specifice și anume: nume, departament (câmpul departament este de tipul Combobox astfel studentul are de ales între 4 departamente si anume: Informatică, Chimie, Fizică, Matematică) si număr de contact iar datele introduse de către student sunt salvate într-o bază de date SQLite.

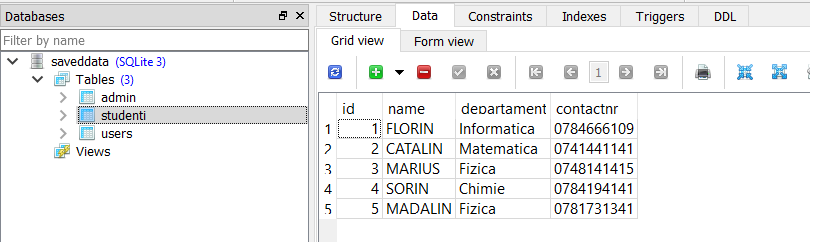


Fig.27 Bază de date proprie



Fig.28 Prelucrarea informaților si adăugarea lor in baza de date

Prima dată se verifica dacă fișierul unde se dorește să fie salvate pozele si informațile există iar dacă acesta nu există, se va creea automat. Datele studentului care s-a înregistrat sunt automat salvate în baza de date. Vom folosi clasa cv :: CascadeClassifier pentru detectarea anumitor obiecte într-un flux video. În special, ne vom folosi de funcțiile acestei clase pentru a detecta persoane într-un flux video.

Pozele au fost salvate într-un fișier din cadrul proiectului numit “dataset”.

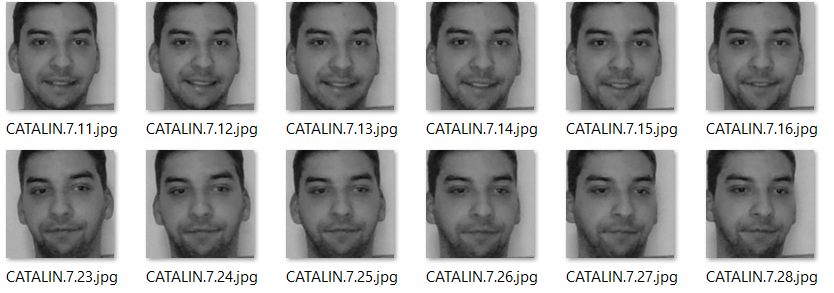


Fig.29 Bază de date proprie

Fiecare poză a fost salvată cu numele studentului, id-ul acestuia si numărului pozei. Imediat ce studentul apasă butonul „Captureaza si salveaza” acestuia i se făc poze. Pentru a nu întampina problema ca studentului să i se facă un număr infinit de poze, am decis ca și condiție de oprire un număr limitat de poze, si anume 50.

Pentru o precizie de detecție mai bună a fost aleasă varianta de a face crop asupra feței astfel încat recunoașterea persoanelor să fie mai rapidă.

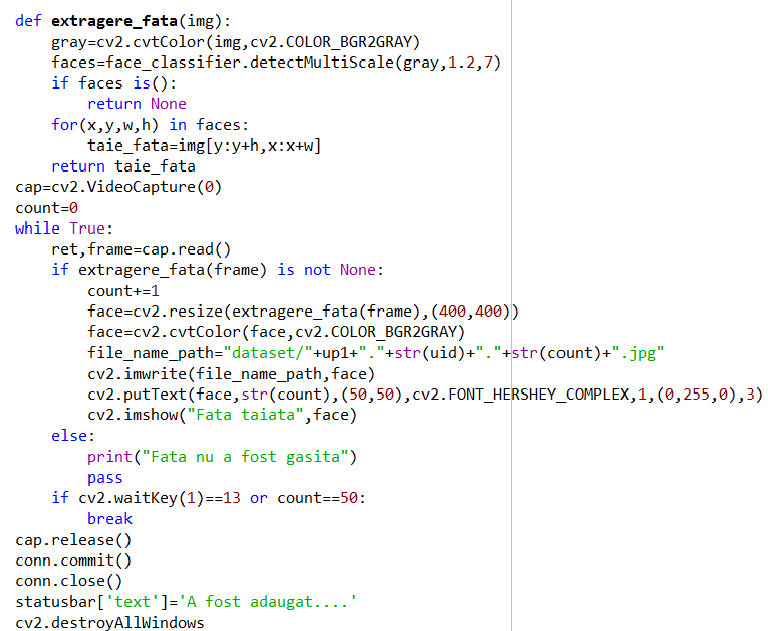


Fig.29 Extragerea feței din imagine

* Imediat ce un student s-a înregistrat, acesta trebuie să **antreneze** modelul pentru ca recunoașterea acestuia să funcționeze corect în sistemul de flux. Recunoașterea persoanelor se face folosind librări OpenCV. Există o mulțime de algoritmi care pot fi folosiți pentru recunoașterea automată de persoane.

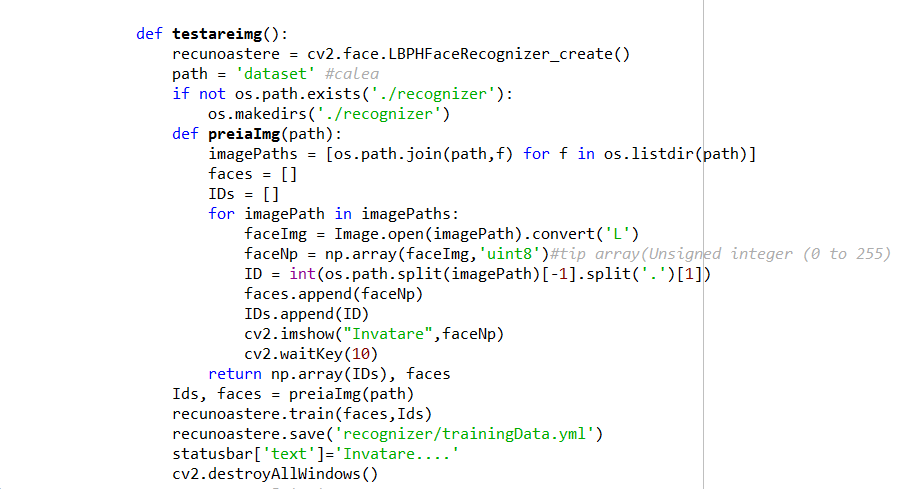


Fig.30 Antrenarea modelului

Datele sunt introduse într-un fișier numit „recognizer” iar salvarea datelor se face conform principiilor LBPH (Local Binary Patterns Histograms).

Metoda funcționează prin compararea feței de recunoscut cu un set de date folosit pentru antrenament unde sunt salvate fețele unor persoan. În setul de instruire, furnizăm chipurile algoritmului și îi spunem cărei persoane îi aparțin.

LBPH analizează fiecare față din setul de antrenament separat și independent. Metoda LBPH este oarecum simplă, în sensul că caracterizează local fiecare imagine din setul de date și atunci când este furnizată o nouă imagine necunoscută, efectuează aceeași analiză asupra acesteia și compară rezultatul cu fiecare dintre imaginile din setul de date.

LBPH oferă anumite facilitați astfel datele au fost salvate într-un fisier yml .

* **Recunoașterea** persoanelor reprezintă procesul cel mai important din cadrul proiectului iar realizarea acestei funcționalități se face cu ajutorul claselor din OpenCv. Folosind datele salvate în xml, recunoașterea persoanelor se realizează într-un mod extrem de simplu.

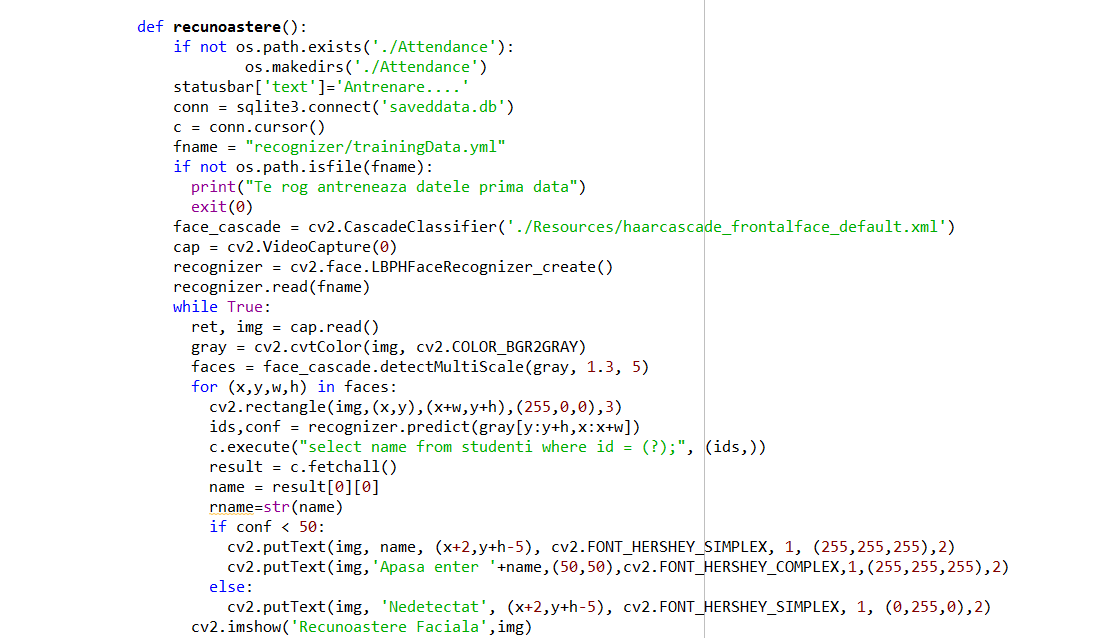


Fig.31 Algoritm pentru recunoașterea persoanelor

Cu ajutorul claselor din OpenCV recunoașterea persoanelor se realizează într-un mod rapid. Pentru recunoașterea persoanelor, importul datelor se face din baza de date unde au fost salvate datele.

Utilizăm LBPH prin intermediul OpenCV pentru a antrena setul de date care produce fișierul TrainingData.yml și pe care îl vom folosi pentru recunoașterea feței.

Pentru ca recunoașterea să aibă un punct de oprire, am integrat în program ca atunci când utilizatorul dorește să finalizeze recunoașterea facială acesta trebuie să apese tasta “Enter”. Pe ecran va fi afișat numele persoanei recunoscute de către program.

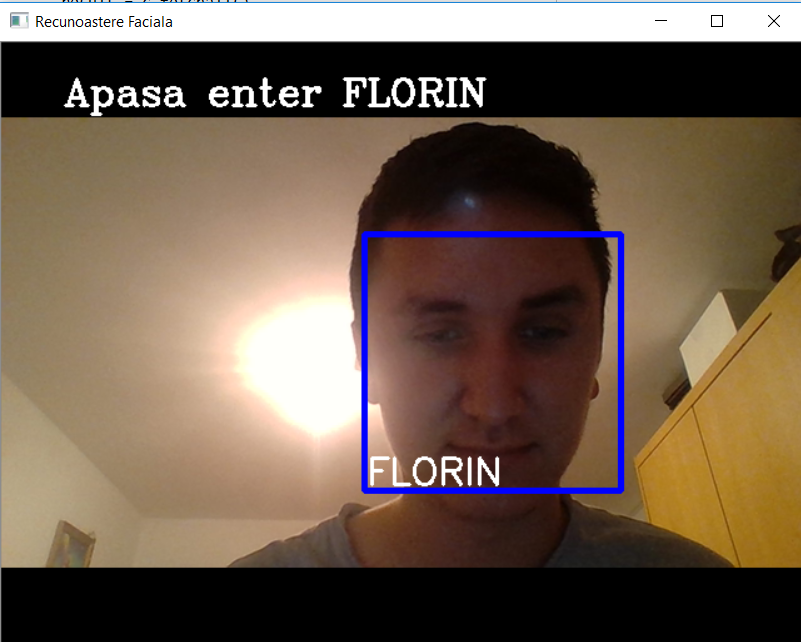


Fig.32 Experiment recunoaștere persoane din baza de date

Recunoașterea persoanelor depinde și de numărul de poze realizate pentru fiecare utilizator astfel încat dacă acestuia îi sunt făcute 150, probabilitatea ca acesta să fie recunoscut mai repede este mai mare iar gradul de acuratețe ale acestuia sa fie înclinat spre partea pozitivă a acestuia .

Pentru o mai bună gestionare a datelor studenților care s-au înregistrat prin intermediul acestei platforme, datele acestora au fost salvate într-un fișier XLSX pentru un control rapid asupra acestora cât si pentru modelarea lor.

Python fiind un limbaj de programare ușor de înțeles inclusiv prin modul de aranjare a liniilor de cod și cu o gamă largă de librări puse la dispoziția utilizatorilor, acesta face posibilă scrierea într-un fișier cu doar câteva linii de cod.

**Componente interfață administrator**

Un administrator este o persoană ce are anumite funcționalități, altele sau asemanătoare cu un utilizator. Principalele funcționalități pe care le poate avea un administrator sunt că acesta poate să aibă un control asupra utilizatorilor lui.

Funcționalități pe care le are un adminstrator în cadrul proiectului :

* Recunoșterea automată de persoane.
* Verificare studenți.
* Căutarea unui student în baza de date.

**Recunoșterea** automată de persoane se realizează la fel ca și în cadrul unui utilizator. Algortimul pentru recunoașterea de persoanelor este aceelași.

**Verificarea** studenților se face prin importul din baza de date a tuturor persoanelor ce au fost înregistrate în sistem. Informațile ce sunt afișate sunt: ID, nume, prenume, departament.

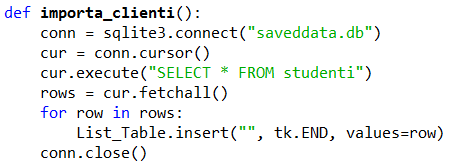


Fig.33 Import studenți din baza de date

Importul studenților din baza de date se face la apăsarea butonului “Vezi date”, astfel se apelează funcția “importa\_clienți”.

**Căutarea** unui student în baza de date se realizează prin intermediul unui metode din Python, unde modulul de operare din Python oferă funcții pentru interacțiunea cu sistemul de operare. Sistemul de operare, este inclus în modulele de utilitate standard Python. Acest modul oferă un mod portabil de utilizare a funcționalității, dependent de sistemul de operare.

Comanda “os.system” a fost folosită în acest proiect deoarece reprezintă un mod rapid de a accesa un alt fișier. Ori de câte ori este folosită această metodă, se deschide shell-ul respectiv al sistemului de operare și comanda este executată pe ea.

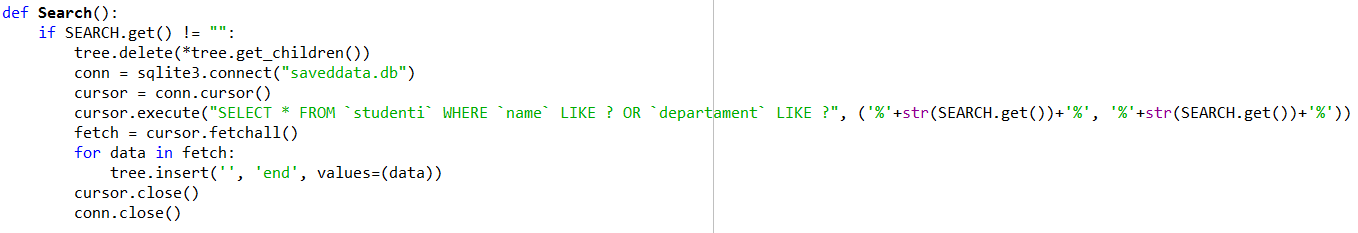


Fig.34 Funcția ce realizează căutarea în baza de date.

Identificarea studenților se realizează după anumite criterii precum nume sau departament. Un administrator poate identifica un student dacă este înregistrat in baza de date sau poate vizualiza toți studenții.

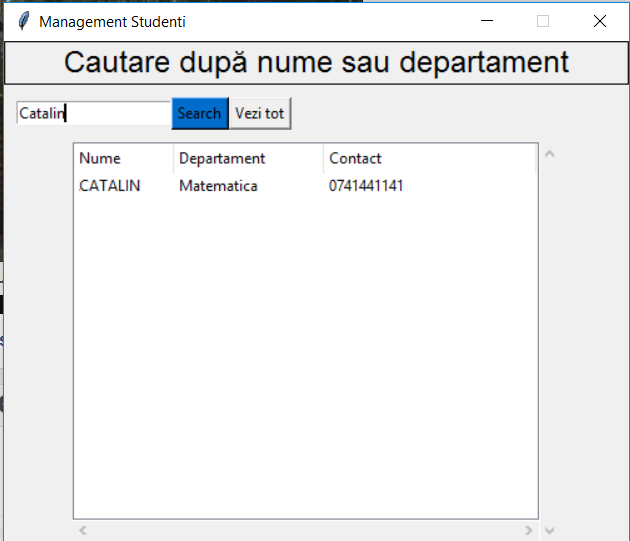


Fig.34 Căutarea unui utilizator în baza de date

**Concluzii**

La momentul actual sistemul este capabil să detecteze fața unei persoane aflată într-o mulțime și să recunoască aceea persoană cu un grad de acuratețe foarte ridicat. Procesarea imaginii şi detectarea este realizată în timp real, însă gradul de recunoaștere facială crește atunci când numărul de poze salvate pentru fiecare persoană este mai mare.

Avantajul major al acestui ansamblu este faptul că poate fi portat pe orice sistem cu resurse și putere de calcul limitată. Pentru realizarea sistemului am avut nevoie de o bază de date și un set de poze pentru recunoașterea facială utilizate pe perioadadesfășurării activității de pregătire a proiectului de licență.

Principala îmbunătățire a fost folosirea proprietății de recunoaștere a persoanelor, astfel dacă avem o sală plină de oameni, sistemul va detecta persoana la un moment dat.

Posibile perspective pe viitor cu privire la acest sistem ar putea fi configurarea pentru logarea persoanelor, atunci când o persoană îsi creează cont, acestuia sa i se facă un număr de poze iar accesul acestuia să se realizeze prin intermediul recunoașterii faciale.