

MINISTERUL EDUCAȚIEI NAȚIONALE



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

---

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

**SISTEM DE RECUNOAȘTERE FACIALĂ PENTRU ACCESUL  
ANGAJAȚILOR ÎN SEDIU , IMPLEMENTAT PE RASPBERRY PI 3**

**LUCRARE DE LICENȚĂ**

Absolvent: **Dumitru Casap**  
Conducător științific: **S.l.dr.Ing Mihai Negru**

**2018**

MINISTERUL EDUCAȚIEI NAȚIONALE



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

DECAN,  
**Prof. dr. ing. Liviu MICLEA**

DIRECTOR DEPARTAMENT,  
**Prof. dr. ing. Rodica POTOLEA**

Absolvent: **Dumitru Casap**

**SISTEM DE RECUNOAȘTERE FACIALĂ PENTRU ACCESUL  
ANGAJAȚILOR ÎN SEDIU , IMPLEMENTAT PE RASPBERRY PI 3**

1. **Enunțul temei:** Este un sistem care detectează persoanele din fața camerei și execută procesul de recunoaștere facială. Scopul acestei aplicații este să decidă pe baza răspunsului de recunoaștere, dacă persoana poate intra în sediul companiei sau nu.
2. **Conținutul lucrării:** Introducere - Contextul Proiectului, Obiectivile Proiectului, Studiu Bibliografic, Analiză și Fundamentare Teoretică, Proiectare de Detaliu și Implementare, Testare și Validare, Manual de Instalare și Utilizare, Concluzii, Bibliografie
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți:** Ing. Vlad Miclea
5. **Data emiterii temei:** 7 Iulie 2018
6. **Data predării:** 9 iulie 2018

Absolvent: Dumitru Casap

Coordonator științific: S.l.dr.Ing Mihai Negru

MINISTERUL EDUCAȚIEI NAȚIONALE



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

**Declarație pe proprie răspundere privind  
autenticitatea lucrării de licență**

Subsemnatul \_\_\_\_\_,  
legitimat cu \_\_\_\_\_ seria \_\_\_\_\_ nr. \_\_\_\_\_  
CNP \_\_\_\_\_, autorul lucrării \_\_\_\_\_  
\_\_\_\_\_

elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea \_\_\_\_\_  
din cadrul Universității Tehnice din Cluj-Napoca, sesiunea \_\_\_\_\_ a anului universitar \_\_\_\_\_, declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Data

Nume, Prenume

\_\_\_\_\_

\_\_\_\_\_

Semnătura

# Cuprins

<b>Capitolul 1 Introducere - Contextul proiectului</b>	<b>1</b>
<b>Capitolul 2 Obiectivele Proiectului</b>	<b>4</b>
2.1 Scopul și obiectivele proiectului . . . . .	4
2.2 Utilizarea sistemului . . . . .	7
2.2.1 Parțile interesante și utilizatorii . . . . .	7
2.2.2 Fluxul de bază . . . . .	8
2.2.3 Fluxuri alternative . . . . .	9
2.2.4 Precondiții . . . . .	10
2.2.5 Postcondiții . . . . .	10
<b>Capitolul 3 Studiu Bibliografic</b>	<b>11</b>
3.1 Stadiul actual al recunoașterii faciale . . . . .	11
3.2 Dezvoltarea sistemului de recunoaștere facială . . . . .	12
3.2.1 Algoritmi . . . . .	13
3.2.2 Dispozitive . . . . .	20
<b>Capitolul 4 Analiză și Fundamentare Teoretică</b>	<b>25</b>
4.1 Alegerea algoritmilor . . . . .	25
4.1.1 Haar Cascades . . . . .	25
4.1.2 Histogram Oriented Gradient . . . . .	27
4.1.3 Local Binary Pattern . . . . .	30
4.2 Compararea Algoritmilor . . . . .	34
4.3 Fuziunea algoritmilor . . . . .	38
4.4 Alegerea dispozitivelor și tehnologiilor . . . . .	40
<b>Capitolul 5 Proiectare de Detaliu și Implementare</b>	<b>42</b>
5.1 Fluxul de recunoaștere a persoanelor . . . . .	42
5.1.1 Capturarea imaginilor și detectia feței . . . . .	43
5.1.2 Extragerea Caracteristicilor . . . . .	44
5.1.3 Clasificator . . . . .	47
5.1.4 Afisarea mesajului pentru utilizator . . . . .	48

5.2	Fluxul de stocare a imaginilor noi . . . . .	49
5.3	Eficentizarea sistemului . . . . .	49
5.4	Rezultate finale . . . . .	51
<b>Capitolul 6</b>	<b>Testare și Validare</b>	<b>54</b>
6.1	Testarea versiunii inițiale și celei refactorizate . . . . .	54
6.2	Testarea versiunii paralelizate . . . . .	58
6.3	Testarea sistemului final . . . . .	58
<b>Capitolul 7</b>	<b>Manual de Instalare și Utilizare</b>	<b>62</b>
<b>Capitolul 8</b>	<b>Concluzii</b>	<b>64</b>
<b>Bibliografie</b>		<b>66</b>

# Capitolul 1

## Introducere - Contextul proiectului

Universul este tot ce în mod fizic există, care este în permanentă mișcare și transformare. El a trăit o evoluție enormă, conform la toate legile fizicii, materiilor, formelor de radiație și energie. Conform teoriei, omul a apărut mult mai târziu, dar acțiunile sale au avut, au și vor avea o influență majoră la evoluție. Omul a devenit un factor care influențează mediul în care trăiește, modificându-l după necesitățile sale. Au fost construite o mulțime de aşezări, obiecte de o varietate largă, inventate și dezvoltate diferite tehnologii și alte lucruri care au schimbat radical stilul de viață. Pe lângă invențiile concepute și dezvoltări enorme realizate, oamenii au avut și foarte multe acțiuni negative. Printre aceste se enumeră distrugerea anumitor specii de organisme, poluarea mediului, defrișarea care au schimbat fața planetei. Există foarte multe consecințe negative asupra întregii planete care nu sunt ca scop a acțiunilor.

- Suprapopularea
- Încălzirea Globală
- Schimbarea climei
- Modificație genetică
- Acidificarea Oceanului
- Ploaie acidă
- Epuizarea stratului de ozon

Thomas Sowell spunea: „Este greu de imaginat o modalitate mai proastă sau mai periculoasă de a lua decizii decât de a pune acele decizii în mâinile unor oameni care nu plătesc nici un preț pentru că sunt răi.” Este imposibil de știut ce idei, dorințe are fiecare om și faptul că ele pot să distrugă ceva. Un lucru important este creșterea securității a anumitor locuri, verificarea oamenilor, analiza lor pentru a fi împiedicați să facă ceva

rău unei anumite proprietăți. În afara de prevenirea acțiunilor destructive, oamenii trebuie motivați, încurajați la o dezvoltarea personală de nivel cât mai înalt. Phil McGraw spunea: „*Conștientizarea fără acțiune nu are valoare*”. Ar fi posibile mai multe acțiuni, spre exemplu folosirea oamenilor pentru a ridica securitatea, dar ei pot fi amăgiți sau ideele lor pot fi schimbată. O modalitatea eficientă ar fi un dispozitiv tehnic de recunoaștere a persoanelor care poate fi făcut prin mai multe metode:

- Recunoașterea irisului

Este o metodă foarte utilizată în prezent. Peste 1000 de bancomate ale instituțiilor financiare din Chicago și Montreal utilizează acum recunoașterea irisului în locul cărților de debit. Irisul este un organ a cărui structură rămâne stabilă pe tot parcursul vieții. Astfel, servește foarte bine pentru stabilirea identității unui individ. Sistemele de recunoaștere a irisului iau imagini de înaltă rezoluție ale irisului ochiului unei persoane și apoi utilizează recunoașterea modelelor pentru citirea și potrivirea modelelor de iris cu modelele stocate în baza de date biometrică. Primul algoritm pentru recunoașterea irisului a fost creat de John Daugman. Avantajele folosirii acestui sistem sunt:

- Irisurile sunt diferite chiar pentru gemenii identici.
- Un iris are mai mult de 266 de grade de libertate (adică numărul de variabile de structură care poate varia în același timp pentru a face irisul unic între oricare doi indivizi).
- Probabilitatea de deteriorare sau zgârieturi există în cazul altor trăsături biometrice, cum ar fi amprente digitale. Dar irisul este bine protejat în spatele pleoapei, corneeii și a umorului apos, ceea ce îl face foarte puțin susceptibil de a se deteriora.
- Irisul nu degenerăza odată cu îmbătrânirea.
- Utilizarea ochelarilor sau a lentilelor de contact nu are nici un efect asupra citirii automate a structurilor de iris.

- Recunoașterea amprentelor digitale

Persoana poate să fie recunoscută prin apropierea degetului de cititorul de amprentă. Măsurătorile bazate pe amprentă reprezintă una din numeroasele noi posibilități de realizare a autentificării biometrice. Este o tehnologie experimentală care este în curs de dezvoltare la un număr de universități și instituții de cercetare. Recunoașterea amprentelor digitale se referă la metoda automată de identificare sau confirmare a identității unui individ pe baza comparației a două amprente digitale. Recunoașterea amprentelor digitale este una dintre cele mai cunoscute elemente biometrice și este de departe cea mai utilizată soluție biometrică pentru autentificarea pe sisteme computerizate. Motivele de ce recunoașterea amprentelor digitale este atât de populară

sunt că au o ușurință de achiziție, o utilizarea stabilă. Este una dintre cele mai ieftine soluții biometrice și deja cunoaște multe utilizări:

- Controlul accesului logic, există numeroase dispozitive pentru cititoare de amprentă și software pentru controlul accesului la computerele personale
- Controlul accesului fizic, încuietori cu cititor de amprentă
- Sisteme de prezență a amprentelor digitale pentru managementul timpului și prezenței
- Alternativă biometrică la sistemele de carduri de loialitate

#### • Recunoașterea facială

Viața ar fi teribil de confuză dacă oamenii nu ar putea recunoaște fețele familiare. Este o abilitate pe care majoritatea persoanelor o acceptă și nu o văd ca un lucru special și ceva important. O consideră ca o obișnuință și ceva simplu la fel ca alte lucruri pe care oamenii le pot face. Lobul temporal al creierului este responsabil de recunoașterea fețelor. Neuronii acestui lob răspund anumitor trăsături ale feței și le stochează în cele din urmă conducând la identificarea feței. Este o magie cognitivă implicată și fiecare persoană are abilități de memorare și recunoaștere a fețelor de nivel diferit. Dacă recunoașterea facială este o abilitate care se dezvoltă individual de la nașterea persoanelor, aceasta este o temă la care se lucrează foarte intens folosind tehnologiile software. Sistemul de recunoaștere a feței este una dintre minunile tehnologice în care o mașină poate acționa aproape la fel de intelligent ca un om. Un sistem de recunoaștere facială este o aplicație informatică capabilă să identifice sau să verifice o persoană dintr-o imagine digitală. În sistemul de învățare a mașinilor, ele sunt adesea hrănite cu o mulțime imensă de imagini pe care sistemul le absoarbe și le stochează. Atunci când o față trebuie să fie potrivită, ea este comparată cu imaginile stocate folosind un anumit algoritm. Sistemul de recunoaștere facială este folosit pe mai multe ramuri în ziua de azi:

- Prinderea de criminali
- Verificarea persoanelor care intră în țară
- Verificarea persoanei care vrea să intre într-o clădire securizată
- Găsirea persoanelor cunoscute
- Evitarea minorilor ca să cumpere alcool sau țigări

În continuare vor fi analizate problemele existente și care ar fi soluția cea mai potrivită. Se vor descrie cele mai folosite aplicații din acest domeniu și care este rolul lor. Fiind un proiect tehnic, se vor analiza mai mulți algoritmi care există și se va face o alegere pentru construirea unui sistem securizat cu un nivel înalt de recunoaștere, viteza mare de răspuns și ușor de utilizat.

# Capitolul 2

## Obiectivele Proiectului

### 2.1 Scopul și obiectivele proiectului

Un sistem de recunoaștere poate fi o soluție la multe probleme descrise mai sus. Recunoașterea persoanelor ar duce la creșterea securității pentru anumite locuri importante și de asemenea ar putea fi un ajutor de dezvoltarea personală. Scopul major al acestui proiect este creșterea securității unei companii și oferirea unei comodități înalte angajaților.

În Figura 2.1 se poate vedea un cititor de cartele, metoda folosită de majoritatea companiilor la intrarea în clădire pentru a permite accesul angajaților. Folosirea ei are ca scop interzicerea persoanelor străine să intre în sediu fără accesul și supravegherea de o persoană din companie. Toți angajații dețin o cartelă personală și pe care trebuie să o aibă pentru a putea intra. Nu este foarte comod pentru persoane, deoarece pot pierde mai mult timp pentru căutarea cartelei la intrare sau poate fi uitată acasă și trebuie să aștepte ajutorul unui coleg. Fiind angajat la o companie care folosește acest sistem de acces în sediu am simțit de multe ori un discomfort din cauza:

- Situații la intrare când pierdeam 30 - 45 de secunde pentru căutarea cartelei în rucsac
- Mă porneam să ies și îmi dădeam seama că am uitat cartela și mă întorceam după ea
- Am uitat-o de multe ori acasă și a trebuit să aștept până vine un coleg ca să pot intra



Figura 2.1: Cititor de cartele

Pe lângă o comoditate mică pentru angajați există și riscuri foarte mari de securitate pentru companie. Fiind una mare și foarte dezvoltată deține multă informație privată și are nevoie de un nivel ridicat de securitate. Aplicațiile interne și bazele de date cu multă informație utilă sunt foarte securizate și sănsele ca să fie accesate de hackeri din exterior sunt foarte mici. Cel mai mare risc este pătrunderea unor persoane externe în sediu. Maniera folosită actual pentru detectarea angajaților și permiterea accesului în firmă este riscantă. Cartela poate fi pierdută, furată și utilizată de persoane care pot provoca un lucru rău. Fiind o companie mare și mulți angajați nu se cunosc între ei, nu ar fi foarte vizibil că o persoană străină se află în sediu. O soluție pentru creșterea comodității angajaților și securității firmei ar fi folosirea unui sistem de recunoaștere a persoanelor la intrare. Persoana recunoscută ca angajat va primi accesul în sediu, iar celelalte vor primi un refuz. Acest produs ar îmbunătăți enorm securitatea companiei și managerii firmei pot fi siguri că în firmă sunt doar angajații sau persoanele invitate. Chiar dacă o persoană străină intră imediat după un angajat, ea va trebui să fie instant verificată și rezultatul negativ va porni o alarmă pentru ca cineva să intervină. De asemenea, numărul de angajați fiind enorm se poate analiza foarte bine timpul pe care o persoană l-a petrecut în sediu.

Din metodele existente de recunoaștere a persoanelor descrise în primul capitol, acest proiect va fi axat pe recunoașterea facială. Chiar dacă recunoașterea irisului sau a amprentelor digitale pot oferi un procentaj de recunoaștere mai mare nu este aşa de potrivit pentru recunoașterea unui angajat la intrarea în sediu. Acest sistem ar trebui să poată detecta și efectua recunoașterea a mai multor persoane, chiar dacă ușa a fost deschisă de primul angajat. Metoda axată pe amprente ar lua mai mult timp angajaților dacă vor să intre mai mulți în același moment sau ar crea un risc de securitate pentru firmă dacă ar intra toți doar conform recunoașterei unei persoane. Recunoașterea irisului de asemenea, va fi mai dificilă pentru situațiile când mai multe persoane vreau să intre în clădire. Scopul creării acestui sistem de recunoaștere facială este axat pe:

- Securitate majoră a companiei - accesul neplanificat interzis pentru persoane străine în firmă. Pot fi concurenți ai firmei, a clienților care pot cauza un risc imens având acces la informația secretă a firmei
- Comoditate pentru angajați - chiar dacă compania este foarte securizată, angajații trebuie să fie satisfăcuți. Este necesar ca sistemul de recunoaștere să aibă o acuratețe înaltă, astfel încât un angajat să fie mereu recunoscut pozitiv și să aibă loc într-un timp foarte rapid.

Pentru atingerea scopurilor de a oferi o securitate majoră companiei și o comoditate mare pentru angajați trebuie atinse următoarele obiective ale acestui proiect:

- În prima luna a proiectului să fie analizați algoritmii de recunoaștere facială și să fie aleși cei mai potrivați pentru acest sistem
- Înainte să se înceapă implementarea proiectului să fie ales un limbaj de programare și un calculator care ar putea oferi o viteză rapidă de recunoaștere

- Înainte de testarea sistemului să se pună accent pe cumpărarea unei camere cât mai profesioniste, astfel încât să ofere o precizie mare de recunoaștere
- Folosind aceeași cameră să fie făcute și stocate multe imagini pentru fiecare angajat cu cât mai multe tipuri de emoții, stil și orientare a feței
- După ce acest sistem va fi pus în funcțiune, locul să fie menținut aproximativ în aceeași luminozitate, căci altfel poate fi afectată acuratețea de recunoaștere
- Sistemul să recunoască corect mai mult de 99% din persoane
- Sistemul să ofere răspunsul de recunoaștere în mai puțin de 2 secunde
- Sistemul să fie ușor de utilizat, mesajele să fie foarte clare și de asemenea să existe o metodă de a intra în sediu pentru persoanele străine invitate
- Sistemul implementat și mențenanța lui să nu fie afectată de schimbarea stilului angajaților și de hainele folosite în dependență de anotimpuri
- Să existe o metodă de rezervă în caz de apariția unor probleme ale sistemului
- Să fie foarte flexibil, să poată fi schimbat calculatorul, sistemul de operare și locul de utilizare.

În Figura 2.2 se pot observa scopurile de bază care trebuie atinse de sistemul de recunoaștere. Acestea sunt securitatea informației, recunoașterea facială, viteza de oferire a unui răspuns și comoditatea pentru utilizatori care trebuie să fie la un nivel cât mai înalt. Un alt obiectiv foarte important ar fi ca sistemul să fie foarte acomodabil, putând fi modificat pentru o nouă utilizare sau scop. După cum s-a văzut existența multor probleme ar fi bine ca sistemul să fie valabil pentru soluționarea mai multor din ele sau să fie ușor de gestionat și modificat. De asemenea, să fie bine structurat astfel încât să permită o dezvoltare ulterioară.

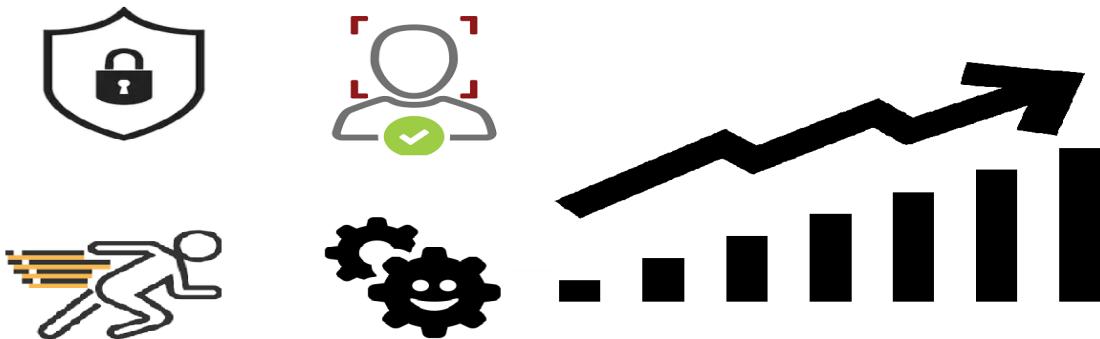


Figura 2.2: Obiectivele sistemului

## 2.2 Utilizarea sistemului

### 2.2.1 Partile interesate și utilizatorii

Părțile interesate ale acestui sistem de recunoaștere facială sunt:

- CEO-ul companiei - este persoana responsabilă de toate deciziile luate în firmă. Este interesat ca compania să fie înalt securizată și să nu aibă probleme cu informația internă
- Clientii companiei - au multe proiecte la care lucrează și sunt menținute de această firmă. Au informații foarte securizate pe care le cunosc doar managerii companiei IT care lucrează la proiectele date. Sunt interesați ca unele idei de dezvoltare și date interne să nu fie furate de concurenți care ar putea afecta dezvoltarea companiei.
- Angajații - sunt responsabili de securitatea proiectelor și informațiilor cu care lucrează. Sunt interesați ca alte persoane să nu aibă acces la acestă informație și la lucrurile lor personale. De asemenea, ar preferă ca intrarea în sediu să fie una comodă și foarte rapidă.

Angajații companiei au rolul de actor principal. Ei sunt persoanele care vor utiliza acest caz al sistemului cel mai des, de mai multe ori pe zi. Scopul lor principal este să primească mereu răspuns pozitiv și să poată intra în sediu indiferent de vestimentația și stilul lor în momentul detecției feței. Pe lângă aceasta, ei sunt interesați să primească răspunsul cât mai rapid. De asemenea, vor fi cazuri când utilizatorii vor fi persoane care nu fac parte din firmă, ele ar trebui să primească un răspuns negativ referitor la accesul în sediu. Cineva responsabil din firmă ar trebui să intervină să clarifice situația.

## 2.2.2 Fluxul de bază

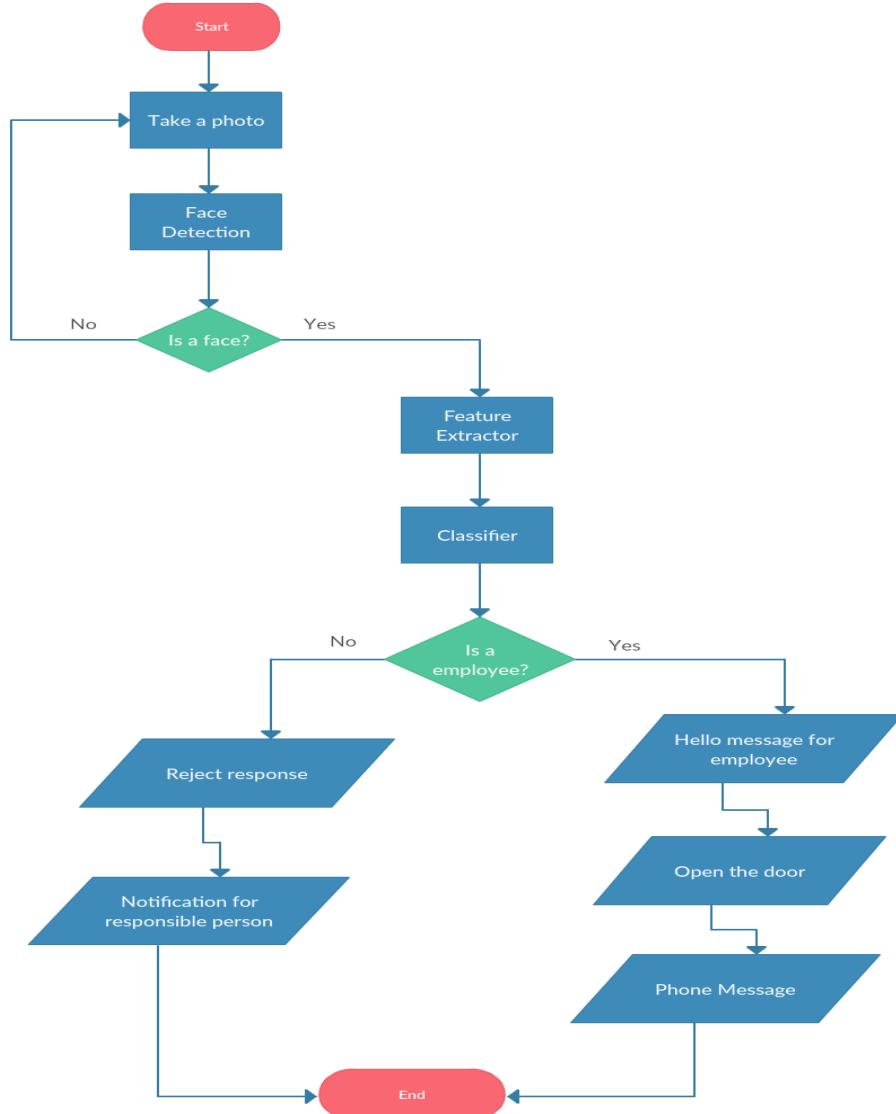


Figura 2.3: Fluxul de evenimente ale aplicației

Cazul de utilizare va rula în continuu cât sediul firmei este deschis. La fiecare secundă se va face detectia feței și restul pașilor se vor executa doar când un utilizator este în fața camerei.

1. Utilizatorul se pune în fața camerei
2. Camera face o poză

3. Sistemul face detecția feței
4. Sistemul verifică dacă a fost detectată o față
  - (a) Dacă nu, se trece la pasul 2 pentru a face o altă poza și a detecta față
  - (b) Dacă da, se decupează imaginea care conține doar față și se trimită mai departe
5. Sistemul aplică algoritmul de recunoaștere facială pentru extragerea caracteristicilor feței detectate
6. Sistemul accesează baza de date cu caracteristicile imaginilor angajaților și face clasificarea persoanei care se află la intrarea în sediu
7. Sistemul verifică răspunsul obținut în urma algoritmului de recunoaștere
  - (a) Dacă nu este recunoscut ca un angajat, primește un mesaj că nu aparține acestei companii și poate doar să contacteze un angajat și să zică de ce are nevoie să aibă acces în sediu. De asemenea, persoana responsabilă de sistem va primi o notificare că cineva nu a fost recunoscut, pentru a putea interveni să știe cauza.
  - (b) Dacă persoana este recunoscută ca un angajat de către sistem, va primi un mesaj de salutare cu numele, ușa clădirii va fi deschisă și va primi un mesaj pe telefon că a intrat în clădire.

Cazul de utilizare pentru această persoană este terminat și va începe altul când un utilizator va fi detectat de cameră.

### 2.2.3 Fluxuri alternative

1. Persoana nu este recunoscută de către sistem, care aparține punctului 7(b) al cazului de utilizare
  - (a) Sistemul face o eroare, utilizatorul este un angajat
    - Angajatul va mai încerca o recunoaștere a feței
    - Dacă primește răspuns negativ, va contacta persoana responsabilă de sistem
    - Angajatul responsabil de acest sistem va permite accesul
    - Va analiza cauza erorii și va încerca să fie eliminată
  - (b) Sistemul dă un răspuns negativ la o persoană străină
    - Persoana responsabilă de sistem va avea poza persoanei care nu a fost recunoscută
    - Va merge să vorbească cu ea și să întelegă necesitatea de accesare a sediului
2. Sistemul face o eroare și recunoaște o persoană străină care vrea să intre în sediu

- Angajatul va primi un mesaj că a primit accesul
  - Având în vedere că el nu a folosit momentan sistemul, este responsabil să dea o știre că o persoană străină a accesat sediul
3. Sistemul nu este funcțional momentan
- Va fi o persoană responsabilă de verificarea persoanelor care vor să intre în sediu
  - Va fi folosită metoda de scanare a cartelei de acces

#### 2.2.4 Precondiții

1. Acțiunea poate să fie făcută când este o zi de lucru și sediul este deschis
2. Sistemul trebuie să fie activ
3. Utilizatorul trebuie să fie instruit cum trebuie să folosească acest sistem pentru a obține accesul în sediu

#### 2.2.5 Postcondiții

1. Sistemul trebuie să rămână consecvent dacă o persoană a primit răspunsul de recunoaștere
2. Sistemul trebuie să afișeze clar răspunsul, astfel încât utilizatorul să știe ce poate face mai departe
3. Dacă utilizatorul primește un răspuns negativ, trebuie să fie informat cum poate contacta pe cineva de la companie. Poate fi un angajat nerecunoscut din cauza unei erori a sistemului sau un client care are o ședință în sediu
4. Persoana responsabilă de sistem trebuie să intervină să vadă cauza și dorința unei persoane care a primit un răspuns negativ

# Capitolul 3

## Studiu Bibliografic

### 3.1 Stadiul actual al recunoașterii faciale

Recunoașterea facială este o aplicație software biometrică, capabilă să identifice sau să verifice persoana, comparând și analizând modelele bazate pe conturul feței. Sistemul este folosit în cea mai mare parte pentru scopuri de securitate, deși există un interes tot mai mare în alte domenii de utilizare. De fapt, tehnologia de recunoaștere a feței a beneficiat de o atenție deosebită, deoarece are potențial pentru o gamă largă de aplicații legate de aplicarea legii, precum și de alte întreprinderi.

În ultimii zece ani, recunoașterea feței a devenit o zonă populară de cercetare în domeniul viziunii calculatorului și una dintre cele mai de succes aplicații de analiză și înțelegere a imaginii. Nu numai cercetătorii din domeniul informaticii sunt interesați de ea, dar și neurologii și psihologii. Opinia generală este că progresele de cercetare în acest domeniu vor oferi informații utile neurologilor și psihologilor despre modul în care funcționează creierul uman și invers. Recunoașterea facială a avut o dezvoltare imensă în ultimii ani. A început să fie folosită de foarte multe companii pe o scară largă.

#### 1. Facebook [1]

Ei au început să folosească sistemul de recunoaștere facială din 2011 și vor să controleze și asigure pozele care sunt încarcate de utilizatori. Facebook are puterea să recunoască 100 de milioane de poze, chiar dacă utilizatorii nu au motiv să aștepte asta. Când o persoană adaugă o poză în care sunt mai multe persoane, Facebook poate recunoaște și eticheta pe toate sau doar să dea această posibilitate utilizatorului. De asemenea, recent a fost implementată și posibilitatea ca utilizatorul să fie notificat că a fost adăugată o poză unde este și el, dar nu este etichetat. Utilizatorii au posibilitatea să gestioneze acest sistem sau chiar să îl închidă.

#### 2. Serviciul de pașapoarte din Australia [2]

Scopul acestui sistem este detectarea fraudelor la pașapoarte și vize. Potrivit rapoartelor, experții care utilizează software-ul automat de recunoaștere a feței în biroul de

pașapoarte australiene sunt cu 20% mai eficienți în comparație cu oamenii care detectează frauda. Acest sistem este tot mai mult folosit de poliție atunci când oamenii aplică pentru documente de identitate ca pașapoarte, vize de emigrare sau permise de conducere.

### 3. ATM-uri și bănci [3]

A început să fie utilizată în China și a asigurat o securitate sporită a cartelei utilizatorului. O nouă mașină de numerar a fost dezvoltată de Universitatea Tsinghua, universitate de cercetare din Beijing. ATM-ul funcționează prin maparea datelor faciale și potrivirea loc cu o bază de identitate. Tehnica utilizată în supraveghere se numește autentificare biometrică și folosește caracteristica facială și recunoașterea irisului. Gu Zikun, peședintele universității care a conceput acest sistem spune că mașina este cu 20% mai exactă la autentificare decât media ATM-urilor folosite global.

### 4. Biroul Federal de Investigații (FBI) [4]

Un instrument din ce în ce mai popular printre agențiile de aplicare a legii. Tehnologia de recunoaștere a feței a contribuit în mod semnificativ în domeniul anchetei și detectării criminalității. Mai multe țări, inclusiv SUA, construiesc baza de date pentru recunoașterea feței pentru a îmbunătăți calitatea investigației. Potrivit unui raport publicat de Centrul pentru confidențialitate și tehnologie la școală de drept din cadrul Universității Georgetown, baza de date privind aplicarea legii din SUA include 117 de milioane de persoane, aproximativ jumate din adulți. Sistemul găsește 15% și cea mai mare probabilitate este la persoanele de culoare.

### 5. Alegerile prezidențiale din Mexic [5]

Are scopul de prevenire a voturilor de fraudă. Detectiona feței a fost folosită în alegerile prezidențiale din 2000 din Mexic pentru a evita votul dublu. Mai multe persoane au încercat să voteze de mai multe ori folosind nume diferite. Voturile duplicate au fost impiedicate în mare măsura, datorită tehnologiei de recunoaștere a feței.

### 6. Școli din Marea Britanie

Au început să folosească recunoașterea facială pentru a detecta prezențele elevilor. Profesorii pot petrece mai mult timp în învățare și mai puțin timp în pronunțarea unor nume și căutarea prin fețe.

## 3.2 Dezvoltarea sistemului de recunoaștere facială

Recunoașterea facială este o aplicație biometrică care mapează matematic caracteristicile unei persoane și stochează datele ca imprimare facială. Software-ul folosește

algoritmi de învățare pentru a compara o captură live sau imagine digitală cu chipul imprimat. Sistemul de recunoaștere a feței este una dintre minunile tehnologice în care o mașină poate acționa aproape la fel de inteligent ca un om. În sistemul de învățare a mașinilor, ele sunt adesea hrănite cu o mulțime imensă de imagini pe care sistemul le absoarbe și le stochează. Atunci când o față trebuie să fie potrivită, ea este comparată cu imaginile stocate folosind un anumit algoritm.

Sistemul poate avea foarte multe abordări de implementare. Există aplicații care fac o recunoaștere facială la imaginile stocate local sau accesate de pe un alt server. Cele mai folosite sisteme, au la bază o cameră care va face imagini live și le va trimite la recunoaștere la un anumit timp selectat, la detectarea unei mișcari sau a unei fețe. După cum se vede în Figura 3.1 sunt mai multe metode care pot fi aplicate pentru a avea o recunoaștere facială în dependență de mediul unde este utilizat și scopul major al aplicației. Poate fi detectată doar față din imagine, se poate preprocesa, normaliza, localiza și după se ajunge la faza cea mai importantă, compararea imaginii cu toate persoanele stocate în baza de date. Pot exista mai multe imagini pentru fiecare persoană, preferabil să fie poziții, expresii și stil diferit al feței. Există mai mulți algoritmi care pot fi aplicati pentru compararea imaginilor și găsirea persoanei cu care se potrivește sau oferirea unui răspuns negativ, deoarece această persoană nu este recunoscută.

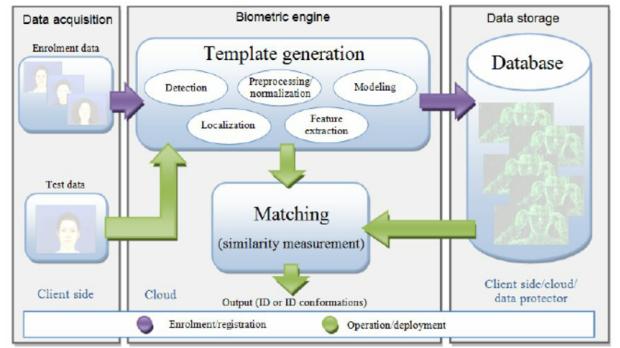


Figura 3.1: Sistem de recunoaștere facială

### 3.2.1 Algoritmi

#### 1. Eigenfaces [6]

Abordarea de folosirea algoritmului eigenfaces a fost dezvoltat de L.Sirovich și M.Kirby în 1987 și a fost folosit de Matthew Turk și Alex Pentland în clasificarea feței. Metoda folosește Principal Component Analysis(PCA)[7], care proiectează liniar spațiul imaginii într-un spațiu cu dimensiuni reduse. Algorimul constă în extragerea eigenvectors din imagine. Dacă imaginea are dimensiunea 256x256, are 65535 pixeli, care reprezintă lungimea vectorului care poate fi format. Imaginele fețelor, fiind similară în configurația generală, nu vor fi distribuite în mod aleatoriu în acest spațiu imens de imagine. În limbajul teoriei informației, dorim extragerea informației relevante a unei imagini. O abordare simplă a extragerii informațiilor conținute într-o imagine este într-un fel captarea variației într-o colecție de imagini de față. Ideea principală din analiza principală a componentelor este de a găsi vectorii care se potrivesc cel mai bine în întregul spațiu imaginii. Acești vectori definesc subspațiul imaginilor, care poate fi numit "spațiu față". Fiecare vector are lungimea  $N^2$ , descrie un

$N \times N$  imagine, și este o combinație liniară a originalului imaginii de față. Deoarece acești vectori sunt vectorii proprii a matricei de covarianță corespunzătoare imaginii originale ale feței și sunt asemănătoare chipului le numim "eigenfaces". Câteva exemple de eigenfaces sunt prezentate în Figura 3.2. Odată ce eigenfaces au fost create, identificare devine o sarcină de recunoaștere a modelului. M vectorii eigenfaces semnificativi sunt aleși cu cele mai mari valori proprii. Cel mai simplu pentru determinarea clasei de fețe care oferă cea mai bună descriere a imaginii unei fețe de intrare este pentru a găsi clasa feței cu distanța euclidiană minimă. În Figura 3.2 este reprezentat un exemplu de aplicare a algoritmului Eigenfaces.



Figura 3.2: Exemplu eigenfaces [8]

Eigenfaces constă din următorii pași:

- Achiziționarea setului de referință pentru față imaginii și calcularea eigenfaces, care definesc spațiul feței
- Când se întâlnește o nouă imagine a feței, se calculează un set de ponderi bazate pe imaginea de intrare și  $M$  eigenii prin proiecția intrării imagine pe fiecare dintre eigenfaces
- Se determină dacă imaginea este o față (de la cunoscut sau necunoscut)
- Dacă este o față, se clasifică modelul de greutate ca o persoană cunoscută sau necunoscută

## 2. Fisherface [9]

Metoda de recunoaștere Fisherface învață o clasă specifică de transformare a matricii. Ea nu capturează iluminarea așa de evident cum o face metoda Eigenface. Este o sporire a algoritmului Eigenface care folosește doar Principal Component Analysis(PCA) [7]. Metoda Fisherface folosește Fisher Linear Discriminant Analysis(FLDA sau LDA) [10] pentru reducerea dimensiunii. Algoritmul Fisherface consideră raportul dintre variația unei persoane și cea a alte persoane. Aceasta maximizează determinantul între clase minimizând factorul determinant din matricea scatter. În Figura 3.3 se vede un exemplu al acestui algoritm.



Figura 3.3: Exemplu fisherfaces [11]

Există mai multe clase, pentru care sunt calculate scatter matrix. În cadrul diferențelor de clase pot fi estimate folosind matricea scatter în interiorul clasei. (3.1)

$$S_w = \sum_{j=1}^C \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T \quad (3.1)$$

$x_{ij}$  este numarul i de probă a clasei j,  $\mu_j$  este media clasei j și  $n_j$  este numărul de eșantioane în clasa j. Diferența între clase se folosește folosind matricea scatter între clase. (3.2)

$$S_b = \sum_{j=1}^C (\mu_j - \mu)(\mu_j - \mu)^T \quad (3.2)$$

$\mu$  este media tuturor claselor. Soluția este dată de generalizarea descompunerei eigenvalue. (3.3)

$$S_b V = S_w V \Lambda \quad (3.3)$$

$V$  este matricea vectorilor proprii și  $\Lambda$  este matricea diagonală corespunzătoare valoilor proprii. Performanța acestei metode de recunoaștere depinde de datele de intrare. Are cele mai bune rezultate atunci când imaginele au o variație largă în iluminare și expresie facială.

### 3. HOG [12]

Hisograma gradientilor orientați (HOG) este un descriptor de caracteristici utilizat în viziunea calculatoarelor și prelucrarea imaginilor în scopul detectării obiectului. Conceptele acestui algoritm au început să fie folosite de Robert K. McConell în 1986, dar folosirea a devenit larg răspândită în 2005, când Navneet Dalal și Bill Triggs, cercetători la Institutul Național Francez au prezentat lucrările lor suplimentare privind descriptorii HOG. Tehnica numără aparițiile de orientare a gradientului în porțiuni localizate ale imaginii. Această metodă este similară cu cea a histogramelor de orientare a marginilor, a descriptorilor de transformare, a caracteristicilor de scală invariante și a contextelor de formă, dar diferă prin faptul că este calculată pe o rețea densă de celule distanțate uniform și folosește normalizarea de contrast local suprapusă pentru o precizie îmbunătățită. Se bazează pe un descriptor de caracteristici care este o reprezentare a unei imagini sau a unui patch de imagine care simplifică imaginea prin extragerea de informații utile și prin aruncarea de informații străine. În mod tipic, un descriptor de caracteristici convertește o imagine cu dimensiunea de lățime  $x$  și înălțime  $y$  cu 3 (canale) la o matrice caracteristică cu lungimea  $n$ . Un exemplu al algoritmului HOG pe o imagine poate fi văzut în Figura 3.4



Figura 3.4: Exemplu HOG [12]

Calcularea Histogramei Gradienților Orientați constă din următorii pași:

- Preprocesarea imaginii
- Calcularea gradienților
- Calcularea histogramei de gradienți, folosind gradienții de direcție și magnitudine obținuți
- Normalizarea histogramei

HOG este folosit în recunoașterea facială prin compararea histogramelor imaginii. Persoana cu care se face asociere este cea care are cea mai mică distanță euclediană dintre vectori imaginilor. Ca descriptor local relativ simplu, caracteristica HOG este utilizat pe scară largă în aplicații precum detectarea pietonilor și de urmărire, dar rareori a fost folosit la recunoașterea facială. Fiind un algoritm relativ simplu, dar nu cu o eficiență foarte bună la recunoașterea feței, caracteristicile HOG sunt cele mai bune alegeri pentru sarcinile de fuziune.

#### 4. LBP [13]

Local Binary Patterns(LBP) este un tip de descriptor vizual utilizat pentru clasificare. Aceasta metodă a fost pentru prima dată introdusă în 1996 de Ojala. Cu LBP este posibil să fie descrisă textura și forma unei imagini digitale de unde vor fi extrase caracteristicile. Aceste caracteristici constau în modele binare care descriu împrejurimile pixelilor din regiuni. Se pot alege distanțe diferite peste stabilirea vecinilor care vor fi luati în calcul și de asemenea numărul de vecini. Operatorul LBP original etichetează pixelii unei imagini cu pragul comparării valorilor pixelor vecini cu valoarea pixelului central și considerând rezultatul ca număr binar. Caracteristicile obținute din mai multe regiuni sunt concatenate într-o singură histogramă caracteristică, care formează reprezentarea imaginii. Imaginile pot fi comparate prin măsurarea similarității (distanță) între histogramele lor. În Figura 3.5 se poate vedea un exemplu LBP.



Figura 3.5: Exemplu LBP [14]

Conform mai multor studii recunoașterea feței folosind metoda LBP are rezultate foarte bune, atât în ceea ce privește viteza cât și recunoaștere. Din cauza modului în care textura și forma de imagini este descrisă, metoda pare a fi destul de robustă împotriva imaginilor cu fețe cu diferite expresii faciale, diferite condiții de lumină, rotația imaginii și îmbătrânirea persoanelor.

### 5. Rețele Neuronale [15]

În mod normal, în învățarea profundă este nevoie de o cantitate mare de date și cu cât sunt mai multe, cu atât rezultatele obținute sunt mai bune. Cu toate acestea, va fi mai convenabil învățarea din doar câteva date, deoarece multe probleme nu pot oferi multe date. La fel este și în recunoașterea facială, sistemul nu va dispune de un număr imens de date pentru a recunoaște o imagine. Rețelele neuronale sunt ca un creier, care nu au nevoie de mii de imagini ale aceluiași obiect pentru a putea recunoaște acest lucru. Ideea principală a acestui algoritm este învățarea unei clase de obiecte doar din câteva date, ca să poată recunoaște persoane din imagini.

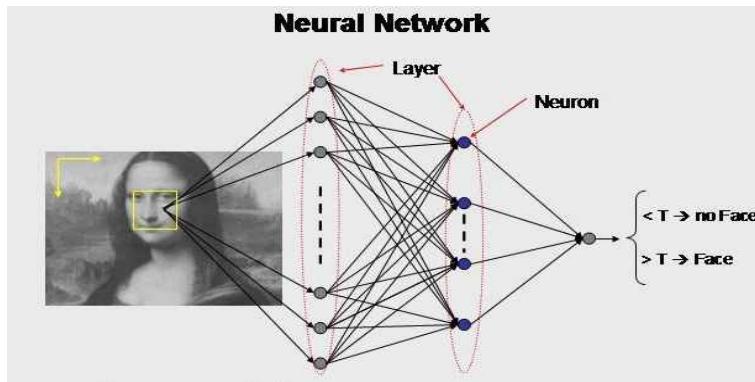


Figura 3.6: Schemă Neural Network [15]

Ideea constă în o funcție de pierdere folosind trei imagini: o imagine de ancorare A, o imagine pozitivă P (aceeași persoană ca ancorea), precum și o imagine negativă N (persoană diferită de ancorea). Așadar, se dorește ca distanța  $d(A, P)$  între codarea ancorei și codificarea exemplului pozitiv să fie mai mică sau egală cu distanța  $d(A, N)$  între codarea ancorei și codarea exemplul negativ. Cu alte cuvinte, trebuie ca imaginile aceleiași persoane să fie apropiate unul de celălalt, iar imaginile diferitelor persoane să fie departe unul de celălalt. Acest lucru poate fi văzut în Figura 3.7.

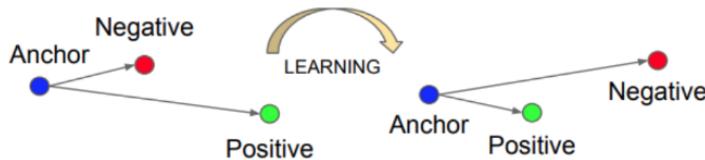


Figura 3.7: Logica algoritmului Neural Network [15]

Algoritmul va începe cu folosirea unui set de instruire care conține mai multe imagini ale aceleiași persoane. Se va face antrenarea modelului pentru a fi capabil să recunoască persoana doar cu o imagine.

### 6. Open Face [16]

Este un model de recunoaștere facială dezvoltat de Brandon Amos, Batrosz Ludwiczuk și Mahadev Satyanarayanan în 2016. În timpe ce Open Face are doar câțiva ani, a fost adoptat pe scară largă, deoarece oferă niveluri de acuratețe asemănătoare modelelor de recunoaștere facială găsite în sistem private de ultima generație, cum ar fi Facebook FaceNet sau Facebooks DeepFace. Sistemul este împărțit în două părți Figura 3.8. Prima este o rețea neuronală - folosită pentru extragerea caracteristicilor. A doua parte necesită mai multe fotografii pe subiect și aici sunt detectate și extrase fețele folosind și algoritmul HOG.

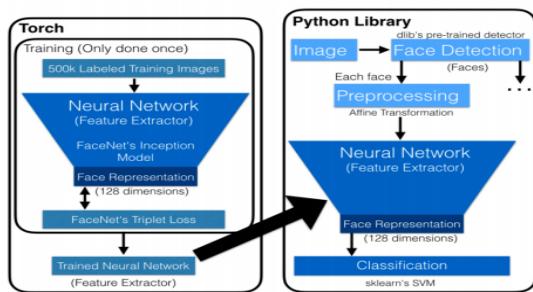


Figura 3.8: Schema Open Face [16]

### 7. Haar Cascades [17]

Comparativ cu algorimii descriși mai sus, nu este folosit la compararea imaginilor și găsirea persoanei cu care se potrivește. Această metodă are rolul de a detecta persoana din imagine și a decupa această porțiune, ceea ce înseamnă că pixelii inutili la recunoaștere vor fi eliberați. Aceasta va crește acuratețea sistemului de recunoaștere. De asemenea, acest algoritm poate detecta diferite obiecte, iar la recunoașterea facială unul foarte important sunt și ochii persoanei. În Figura 3.9 se vede detectarea feței și a ochilor persoanelor.

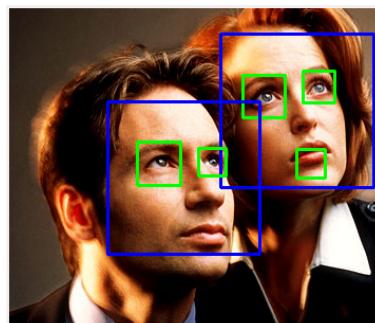


Figura 3.9: Exemplu Haar Cascades [17]

### 3.2.2 Dispozitive

Pentru un sistem de recunoaștere facială, pe lângă alegerea unor algoritmi care să corespundă modului de utilizare, este foarte important și dispozitivul pe care va fi rulat. Acuratețea, viteza de recunoaștere, numărul de persoane care pot fi stocate, detectarea fraudelor sistemului sunt influențate foarte mult de dispozitivul folosit.

#### PC (Personal Computer)

Calculatorul este cel mai folosit dispozitiv pentru implementarea multor algoritmi complexi și chiar în domeniul de procesare de imagini. Un PC are 2 elemente foarte importante CPU(central processing unit) și GPU(graphics processing unit). CPU fiind considerat ca un creier al PC-ului, este responsabil de citirea și execuția majoritatea comenzilor hardware și software. Intel și AMD sunt cele mai populare CPU-uri. Viteza procesoarelor depinde de:

- Viteza ceasului - depinde de numarul de cicluri de ceas pe secundă
- Arhitectura procesorului - specifică situației CPU-ului care determină căt de individuale sunt componentele puse împreună
- Lungimea cuvântului în procesor - este numărul de biți pe care CPU le poate procesa la un moment dat
- Numărul de nuclee din procesor - mai multe nuclee se referă la multe CPU cu un singur cip, unde fiecare thread poate fi executat independent și concurent.

GPU este considerat ca sufletul unui calculator și a fost proiectat cu scopul de procesare grafică foarte rapidă. Din moment ce aceasta este singura preocupare pe care o au, au existat unele optimizări, care permit ca anumite calcule să meargă mult mai repede decât ar fi într-un procesor tradițional. GPU-urile își iau viteza pentru un cost. Un singur GPU funcționează de fapt mult mai lent decât un singur procesor. GPU are mai multe nuclee (până la 16), fiecare operând într-un mod SIMD la 32 de niveluri. Asta aduce 500 de operațiuni efectuate în paralel. CPU-uri comune, cu toate acestea, au până la 4 sau 8 nuclee, și pot funcționa în SIMD la 4 niveluri, care oferă paralelism mult mai mic. Anumite tipuri de algoritmi (procesarea grafică, algebra liniară, codarea video etc.) pot fi ușor paralele pe un număr atât de mare de nuclee. Paralele de rupere se încadrează în această categorie. Mulți algoritmi sunt cu adevărat greu de paralelizat. CPU este mult mai puternic secvențial și folosit la probleme mult mai mari care nu pot fi implementate paralel, iar GPU are foarte multe nuclee și poate rezolva foarte rapid anumiți algoritmi în paralel (Figura 3.10).

A fost făcut un experiment (Figura 3.11) pentru un sistem de detecția feței pe CPU și GPU de Gaowei și Cheming la Universitatea Tianjin din China. Algoritmii au fost testați pe un PC cu Intel(R) Core i7 920 și NVIDIA GeForce GT260 GPU.



Figura 3.10: CPU vs GPU [18]

	<i>Core i7 920</i>	<i>Geforce GTX 260</i>
<i>Number of processor cores</i>	4	24
<i>Processor core clock</i>	2.67GHZ	576MHZ
<i>ALU clock</i>	5.34GHZ	1242MHZ
<i>Memory size</i>	12 G	896MB
<i>Local cache per core</i>	256KBL1+1MBL2+8MBL3	16KB(shared)+8KB(constant)+8KB(texture)
<i>Bandwidth</i>	4.8GT/s	112GB/s
<i>Thermal design power</i>	130W	183W

Figura 3.11: Rezultatele Intel(R) Core i7 920 vs NVIDIA Geforce GT260 [18]

Algoritmul pentru detecția feței care rulează pe CPU se bazează pe Adaboost. Acest algoritm a fost adoptat pentru a fi rulat paralel pe GPU. Algoritmul constă din 3 pași.

- Preprocesarea imaginii
- Calcularea imaginii integrale
- Detecția feței folosind o cascadă de clasificatori

Pentru realizarea experimentului au fost alese 61 imagini frontale (640x480) cu 146 fețe frontale. 134 de fețe au fost detectate corect rata de detecție fiind de 92%. Viteza de detecție este legată de dimensiunea imaginii, numărul clasificatorilor, caracteristici sau scala de redimensionare a imaginii și platforma hardware. Primii factori sunt strâns corelați cu rata de detecție, astfel încât au fost menținuți și neschimbați pentru testarea sistemului accelerat bazat pe hardware-ul CPU+GPU. GPU a adus o îmbunătățire foarte mare a vitezei algoritmului, rezultatele pot fi văzute în Figura 3.12.

	<i>CPU(ms)</i>	<i>GPU(ms)</i>	<i>Speed-up ratio</i>
<i>Image transforming</i>	1.396	0.121	11.537
<i>Image resizing</i>	8.339	2.582	3.229
<i>Integral image calculation</i>	10.936	3.451	3.169
<i>Face detection</i>	360.187	75.132	4.794

Figura 3.12: Creșterea vitezei folosind GPU [18]

### FPGA (Field-Programmable Gate Array)

Este un circuit integrat, conceput pentru a fi configurat de către o persoană după fabricare în general folosind o limbaj de descriere hardware(HDL). FPGA conține porți și blocuri de construcție, ale căror conexiuni sunt reprogramate prin trecerea semnalelor electrice în ele. Avantajul mare este programarea hardware care are foarte multe avantaje față de cea software. Instrucțiunile Verilog, VHDL în programarea FPGA sunt în mare parte executate în paralel. De asemenea, programarea se referă la proiectarea circuitelor logice și digitale pentru a defini comportamentul acestui dispozitiv, care este mult mai rapidă comparativ cu executarea unei secvențe de instrucțiuni secvențiale în programarea software.

Multe aplicații în procesarea imaginilor au o mare legătură cu paralelismul, iar lățimea datelor a multor operațiuni este mai mică decât 16 biți. FPGA poate executa aceste operații în paralel prin configurarea circuitelor dedicate pentru fiecare aplicație. Implementarea executată de J.Matai, A.Irturk și R.Kastner la Universitatea California din San Diego, arată rezultatele care pot fi obținute de folosirea acestui dispozitiv la recunoașterea facială [19]. Sistemul e o soluție completă pentru recunoașterea feței. Primește ca intrare un video de la o cameră, se detectează locațiile feței folosind algoritmul Viola-Jones și ulterior se recunoaște fiecare față folosind algoritmul Eigenface și se afișează rezultatele pe un afișor. S-a folosit un Virtex-5 FPGA cu 45 cadre pe secundă. Pașii acestui sistem sunt:

- Citirea imaginii
- Normalizarea imaginii
- Găsirea vectorului de ponderi pentru trăsături
- Citirea vectorului de ponderi
- Clasificarea și proiecția rezultatului

Algoritmul a fost implementat în două moduri, cu pipeline și fără pipeline și s-a făcut o testare pentru 40, 50 și 60 de imagini. În tabelul de mai jos se văd rezultatele

experimentului: latență în milesecunde, lateță de cicluri și analiza dispozitivului folosit pentru numărul de LUT și blocuri RAM (BRAM). Viteza de recunoaștere este foarte rapidă comparativ cu alte dispozitive utilizate (Figura 3.13).

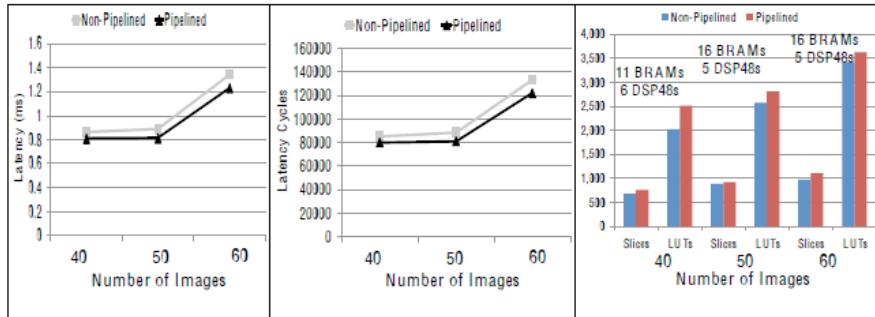


Figura 3.13: Algoritmul implementat cu pipeline și fără pe FPGA [19]

Un dezavantaj al FPGA-ului este complexitate majoră de a crea proiecte foarte mari. Implementarea hardware este foarte diferită de acea software și chiar dacă viteza de execuție este mult mai bună timpul care trebuie depus pentru implementare este mult mai mare. Instrumentele de sinteză pot dura mult timp pentru a sintetiza codul și chiar mai rău, software-ul de sinteză continuă să sintetizeze până când se revarsă calculatoarele RAM și trebuie să fie închis software-ul și să fie reluat.

## Raspberry PI

Raspberry PI este o serie de calculatoare mici cu o singură placă. Scopul acestui dispozitiv era să promoveze învățarea programării școli de persoane indiferent de vârstă. Chiar dacă este mai mult orientat pentru prezentarea bazei domeniului IT la copii, este folosit de foarte multe persoane pentru proiectarea diferitor proiecte. A devenit cel mai vândut calculator din Marea Britanie, în martie 2018 vânzările au atins 19 milioane. Acest dispozitiv este foarte ieftin, accesibil și ușor de utilizat. Există posibilitatea de a fi conectate multe lucruri adiționale pentru a face proiecte cât mai complexe. Acest lucru se poate observa chiar în sistemul de recunoaștere facială creat la Universitatea Savitribai Phule Pune din India. [20] Pentru acest sistem (Figura 3.14) sunt folosite:

- Cameră pentru detectarea fețelor
- Afisor LCD pentru prezentarea rezultatului
- Mouse, Tastatură, Monitor pentru implementare
- SD Card pentru stocarea imaginilor

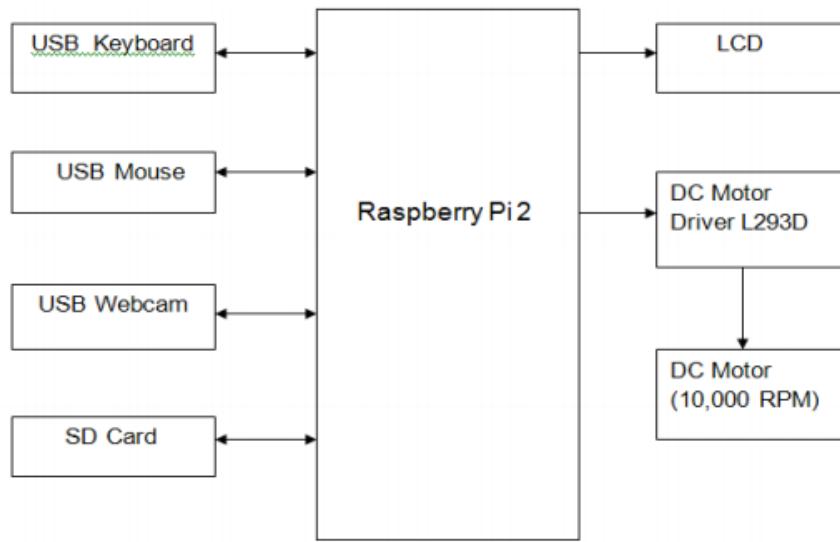


Figura 3.14: Raspberry [20]

Folosind acest dispozitiv, sistemul a fost eficient la preț, ușor de folosit și performanță înaltă. Printre dezavantajele la Raspberry PI sunt viteza algoritmilor implementați și memoria pentru stocarea unui set mare de date. Chiar dacă Pi 2 și Pi 3 conțin procesoare quad-core, ceea ce înseamnă că pot executa simultan patru procese pentru reducerea timpului de execuție, în practică este ceva dificil de a folosi programarea paralelă pe Raspberry PI.

# Capitolul 4

## Analiză și Fundamentare Teoretică

### 4.1 Alegerea algoritmilor

În urma unei analize mai largi a algoritmilor de recunoaștere facială, se va alege cei mai corespunzători acestui sistem. Acest lucru depinde enorm de modul cum va fi folosit sistemul și care sunt factorii care influențează. În ultimii ani se vede că cercetările efectuate în tehniciile de recunoaștere a feței au obținut un impuls semnificativ. Implementarea reală este foarte provocătoare, deoarece trebuie să se țină cont de toate variațiile posibile ale imaginii: iluminare, caracteristici faciale, variații de imagine, zgomot, distanță de vizionare. Multă algoritmă de recunoaștere a feței au fost dezvoltăți și fiecare are punctele sale forte. Sistemul dat având scopul să fie folosit la intrarea angajaților în blocul companiei, va fi folosit în mare parte același mediu și factorii care vor influența recunoașterea este variația de lumini care va apărea în imagini, emoțiile faciale și poziția feței.

#### 4.1.1 Haar Cascades

Primul algoritm ales nu are rolul de recunoaștere a feței, dar este responsabil de foarte mulți factori de funcționare bună a sistemului. Fiind bazat pe cascadă, este o metodă eficientă propusă de Paul Viola și Michael Jones în lucrarea din 2001, "Detectarea rapidă a obiectelor utilizând o cascadă de caracteristici simple". Poate fi folosită pentru detecția diferitor obiecte, iar în acest proiect va fi folosit pentru detecția feței. Are o abordare pe învățarea mașinii în care o funcție cascadă este învățată cu un set de imagini care conțin acel obiect (față în cazul sistemului de recunoaștere facială) și alt set care nu conțin obiectul dorit. Sunt extrase caracteristici din imagini, folosind caracteristicile Haar prezentate în Figura 4.1. Ele sunt folosite la fel ca un nucleu convecțional. Fiecare caracteristică este o singură valoare obținută prin

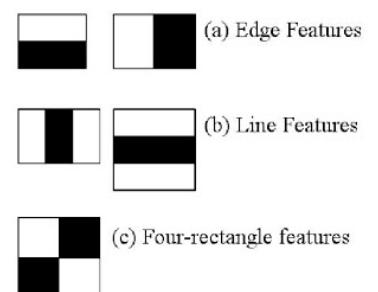


Figura 4.1: Caracteristici Haar [17]

scăderea sumei de pixeli sub dreptunghiul alb din suma pixelilor sub dreptunghiul negru. Se aplică fiecare caracteristică tuturor imaginilor de referință. Pentru fiecare caracteristică, acesta găsește cel mai bun prag care va clasifica fețele pozitive și negative. Evident, vor exista erori sau clasificări greșite. Se selectează caracteristicile cu o rată minimă de eroare, ceea ce înseamnă că acestea sunt caracteristicile care clasifică cel mai bine imaginile de pe față. Fiecare imagine are o pondere egală la început. După fiecare clasificare, greutatea imaginilor clasificate eronate sunt crescute. Apoi, se repetă același proces. De asemenea, sunt calculate rate noi de eroare și procesul este continuat până când se obține precizia sau rata de eroare sau se obține numărul necesar de caracteristici. O imagine poate avea 160000 de caracteristici și asta ar face ca să fie necesar foarte mult timp pentru detectia unui obiect. O soluție bună este împărțirea imaginii în mai multe ferestre, dar de asemenea timpul de comparare cu caracteristicile va fi foarte mare. Pentru aceasta s-a introdus conceptul de cascadă a clasificatorilor. În loc să se aplice toate caracteristicile pe o fereastră, se grupează caracteristicile în diferite etape ale clasificatorilor și se aplică una câte una. În Figura 4.2 se vede parcursarea unei imagini pentru găsirea caracteristicilor și detectia feței.



Figura 4.2: Algoritmul Haar [17]

Algoritmul Haar Cascades ajută la:

- Creșterea acurateței - pentru recunoașterea unei fețe, trebuie să fie comparate doar fețele persoanelor. Pixelii din fundalul imaginii nu merită să fie comparați, deoarece o persoană nu va putea fi recunoscută dacă se află în locuri diferite.
- Creșterea vitezei de recunoaștere - fiecare algoritm necesită un anumit timp de parcursere a tuturor pixelilor din imagine și pentru crearea vectorului de caracteristici.

Detectia feței va reduce semnificativ numărul de pixeli din imagine.

- Apelarea functiilor de recunoastere doar în momentul potrivit - nu se merită ca algoritmul de recunoastere a feței să fie apelat la o anumită perioadă de timp, dar doar atunci când sistemul detectează o față.

### 4.1.2 Histogram Oriented Gradient

Fiind un algoritm care este folosit mai mult la detecția obiectelor, sau la detecția mediului cărui aparține poza, este folosit și la recunoasterea facială și are o acuratețe relativ bună. Ideea de bază a caracteristicilor HOG este că apariția obiectului actual poate fi caracterizată mai degrabă prin distribuirea gradientilor de intensitate locală sau direcții de margine. Metoda este robustă la schimbările de lumină, deoarece histograma dă invarianță translatională. Funcția HOG rezumă distribuția măsurătorilor în regiunile imaginei și este în special utilă pentru recunoasterea obiectelor texturate și cu forme deformabile. Metoda este de asemenea simplă și rapidă. Algoritmul constă în calcularea unui descriptor de caracteristici, care este o reprezentare a unei imagini, care simplifică imaginea prin extragerea informațiilor utile și prin aruncarea celor străine. În mod tipic, un descriptor de caracteristici convertește o imagine într-un vector de caracteristici, care constă din următorii pași:

- Calcularea gradientilor

Primul pas este calcularea gradientilor orizontali și verticali. Acest lucru se obține prin filtrarea imaginii folosind următoarele nuclee (Figura 4.3).



Figura 4.3: Nuclee pentru calcularea gradientilor [21]

Filtrarea unei imagini folosind un nucleu, constă în parcurgerea imaginii cu matricea caracteristică nucleului. Se iau toți pixelii cuprinși de matrice, și se face o sumă a produselor dintre pixel și valoarea care îi corespunde din nucleu. Filtrarea unei imagini și cum arată calcularea unui pixel pentru rezultatul filtrării este reprezentată în Figura 4.4

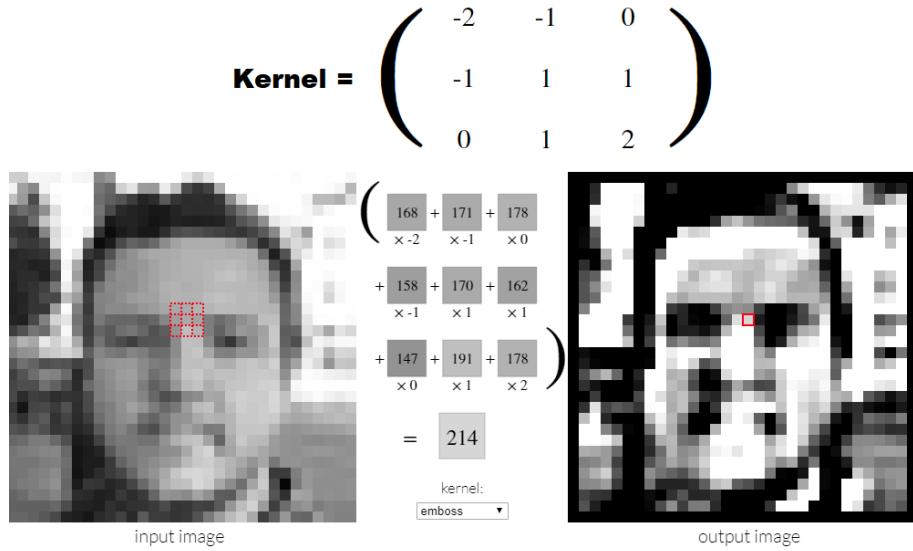


Figura 4.4: Filtrarea imaginii conform algoritmului HOG [21]

Folosind gradienții orizontali și verticali se calculează direcția și magnitudinea gradientului(4.1)(4.2).

$$g = \sqrt{g_x^2 + g_y^2} \quad (4.1)$$

$$\theta = \arctan \frac{g_x^2}{g_y^2} \quad (4.2)$$

$g$  - gradientul de magnitudine,  $\theta$  - gradientul de direcție,  $g_x$  - gradientul orizontal și  $g_y$  - gradientul vertical. Prin calcularea gradientului s-a renunțat în mod deliberat la imaginea care arată direcția de înclinare, deoarece ea nu transmite prea mult. De asemenea, imaginea gradientului a îndepărtat o mulțime de informații neesențiale, dar a subliniat contururi. Aceasta ajută sa fie ușor vizibil că este o persoană în imagine. La fiecare pixel, gradientul are o magnitudine și direcție.

- Calcularea histogramei de gradienți

Histograma este un vector de 9 poziții corespunzătoare unghilor  $0,20,40 \dots 160$ . Este recomandat să fie folosite unghiurile doar între  $0^\circ$  și  $180^\circ$ , în loc de  $0^\circ$  pâna la  $360^\circ$ , deoarece chiar s-a demonstrat că aceștia funcționează mai bine la detectarea pietonilor. Pentru a construi histograma, se lucrează cu matricea de gradienți și de magnitudine(Figura 4.5). Pentru un pixel care are unghiul  $80^\circ$  și magnitudinea 2, se va adăuga 2 pe a 5 poziție din histogramă. Atunci când unghiul nu este egal cu unul corespunzător valorii specifice pentru indexul histogramei, unghiul se va împărți și se va adăuga pe 2 poziții. Pentru gradientul 10 și magnitudinea 4, se va adăuga 2

pe indexul 0 și 1 al histogrammei. Pentru unghiuile mai mari ca 160, gradientul va merge pe primul și ultimul index la histogrammei.

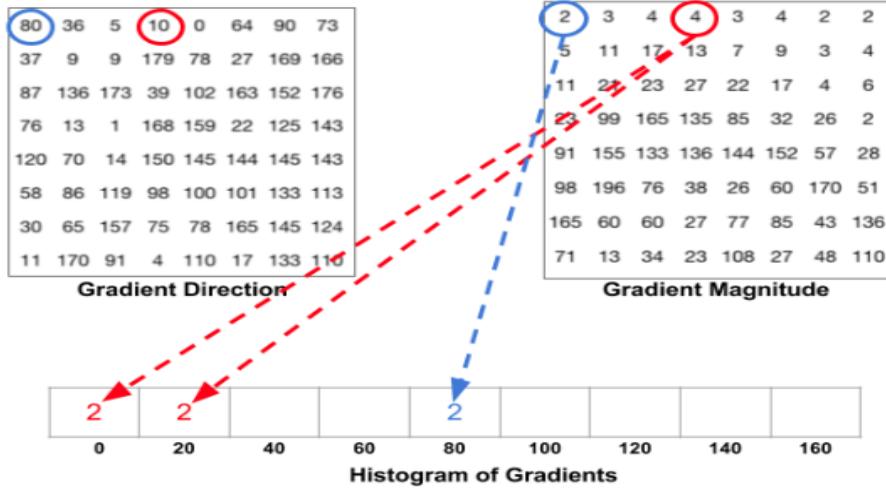


Figura 4.5: Calcularea histogrammei de gradienți [21]

- Normalizarea histogrammei (Figura 4.6)

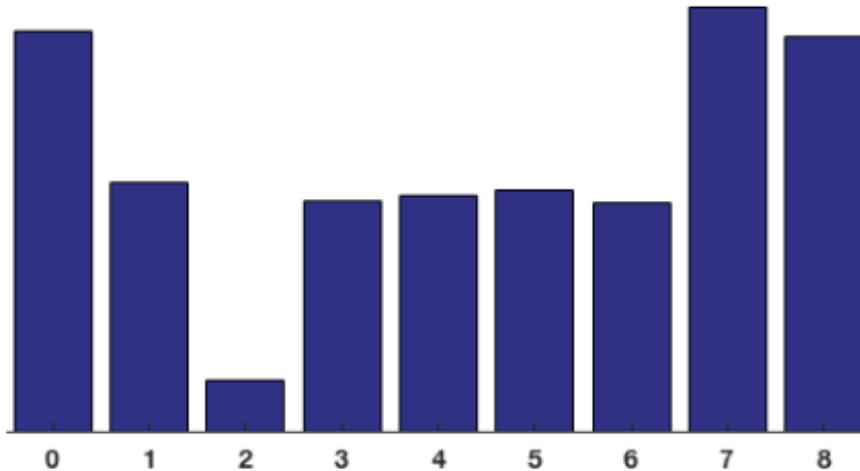


Figura 4.6: Normalizarea histogrammei [21]

Aceasta fiind histogramă obținută, pentru a crește acuratețea de recunoaștere facială este normalizată. Imaginele comparate și folosite de algoritm, având dimensiuni diferite, una cu un număr mare de pixeli și alta cu un număr mic, chiar fiind aceeași față nu va fi recunoscută. Normalizarea imaginii va aduce valorile histogrammei în intervalul 0 și 1, indiferent de dimensiunile imaginilor. Un alt scop al normalizării

este pentru a avea o invarianță mai bună la iluminare și zgromot. Fiecare element din histogramă se va divide cu radicalul din suma pătratelor a celor 9 valori din histogramă (4.3).

$$h_i = \frac{h_i}{\sqrt{h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 + h_7^2 + h_8^2}} \quad (4.3)$$

În Figura 4.7 se prezintă tot fluxul de execuție a algoritmului Histogram Oriented Gradients. Pentru fiecare pixel, se calculează gradientul orizontal și vertical, care sunt folosite pentru calcularea magnitudinii și gradientului de direcție. Aceste două valori au rolul de a adăuga o valoare pentru pixelul curent în descriptorul de caracteristici al imaginii, care este o histogramă cu 9 poziții.

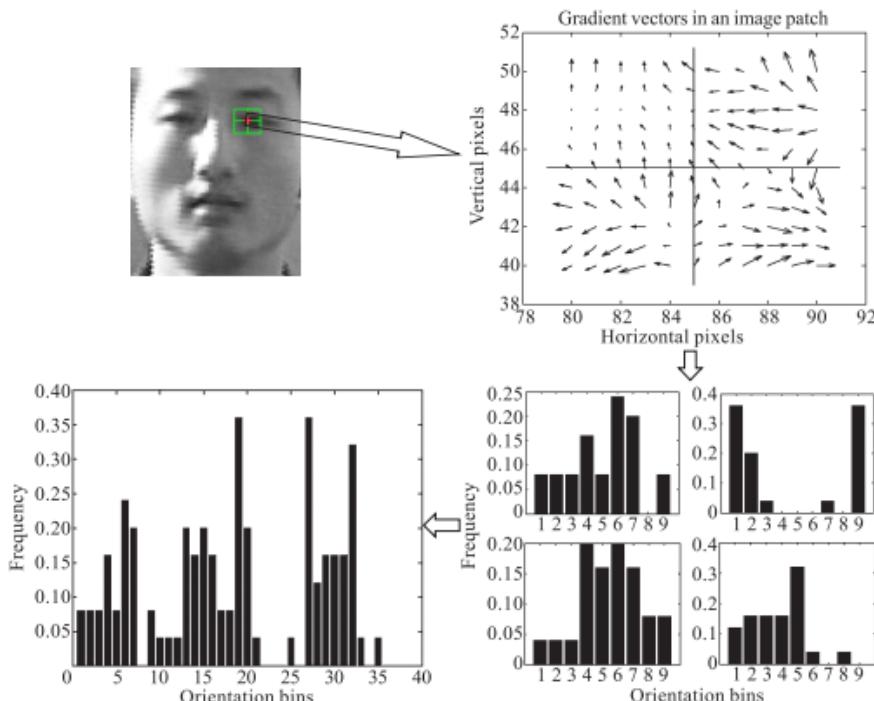


Figura 4.7: Fluxul algoritmului HOG [12]

#### 4.1.3 Local Binary Pattern

În urma testărilor descrise în capitolul 3, s-a văzut că este unul dintre algoritmii cu cea mai bună viteză și acuratețe de recunoaștere. Este al doilea algoritm ales pentru implementarea acestui sistem. Consta în extragerea celor mai importante caracteristici dintr-o imagine pentru a efectua recunoașterea facială. Cu LBP este posibil să se descrie textura și forma unei imagini digitale. Acest lucru este făcut prin împărțirea unei imagini

în mai multe regiuni mici din care fac parte caracteristicile extrase. Aceste caracteristici constau din modele binare care descriu împrejurimile pixelilor din regiuni. Caracteristicile obținute din regiuni sunt concatenate într-o singură histogramă caracteristică, care formează reprezentarea imaginii. Imaginile pot fi apoi comparate prin măsurare similitudinei (distanței) între histogramele lor. Metoda LBP a fost deja utilizată în mai multe aplicații din întreaga lume ca inspecție vizuală, recuperarea imaginilor, analiza imaginii biomedicale, analiza imaginii feței, analiza mișcării, modelarea mediului și analiza scenei în aer liber. Forma actuală a metodei LBP este destul de diferită de versiunea de bază, au fost dezvoltate o serie de extensii în ultimii câțiva ani. Un LBP este un cod binar pentru un pixel al imaginei, care spune ceva despre vecinătatea locală a acestui pixel. Pașii de implementare a acestui algoritm [22] sunt:

- Stabilirea vecinilor pentru un pixel din imagine

Histograma LBP se construiește pe baza comparării fiecărui pixel cu vecinii săi. Există mai multe posibilități de alegere care sunt vecinii unui pixel, utilizând vecinătate de dimensiuni diferite (Figura 4.8). Se face un cerc cu raza R de la pixelul central și P puncte de pe marginea acestui cerc sunt considerate ca vecini.

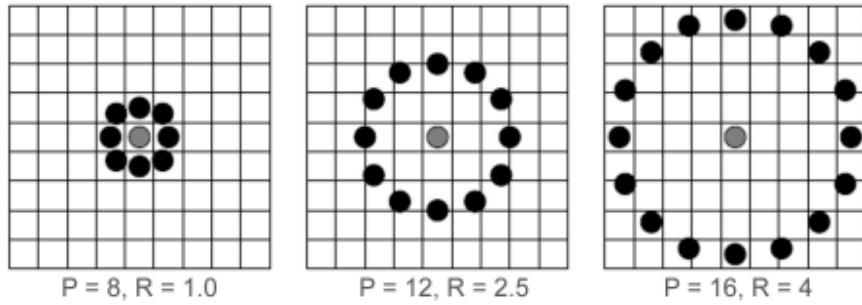


Figura 4.8: Tipuri de vecinătăți folosite de LBP [13]

Dacă coordonatele pixelului central sunt  $(x_c, y_c)$ , atunci coordonatele celor P vecini  $(x_p, y_p)$  pe marginea cercului pot fi calculați cu sinus și cosinus (4.4).

$$x_p = x_c + R \cdot \cos(2\pi/P) \quad y_p = y_c + R \cdot \sin(2\pi/P) \quad (4.4)$$

- Compararea pixelului central cu vecinii

Presupunem că  $P = 8$  și  $R = 1$ , fiecare pixel central va avea 8 vecini. Valoarea gri a pixelului central este comparată cu valoarea gri a pixelilor din vecinătatea sa. Se va construi un vector de biți, cu lungimea P (Figura 4.9).

Dacă valoarea pixelului din centru este  $g_c$ , și valorile vecinilor sunt  $g_p$ , cu  $p = 0, \dots, P-1$ , atunci textura T din vecinătatea locală a pixelilor  $x_c, y_c$  pot fi făzute ca în relația (4.5).

$$T = t(g_c, g_0, g_1, \dots, g_{P-1}) \quad (4.5)$$

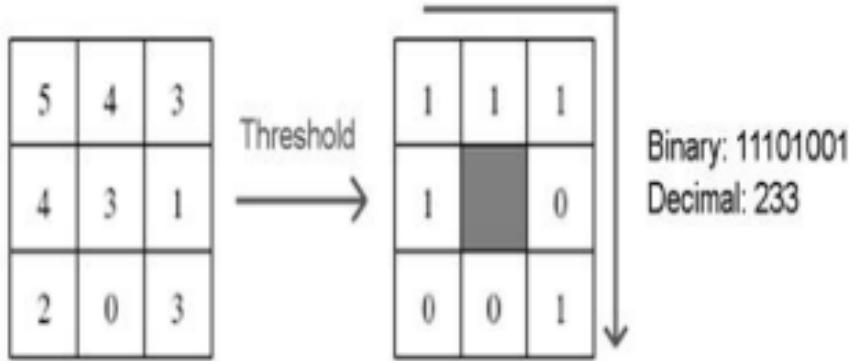


Figura 4.9: Construirea unui vector de biți [13]

Se vor parcurge toți vecinii pixelului curent și pentru a obține invariантă în ceea ce privește orice transformare monotonă din grila gri, sunt luate în considerare doar semnele diferențelor. Asta înseamnă că în cazul în care un punct din cerc are o valoare mai mare decât cea a pixelului din centru(sau aceeași valoare), este atribuit 1 la acest punct, altfel 0. (4.6)

$$T = (s(g_0 - g_c), (g_1 - g_c), \dots, (g_{P-1} - g_c)) \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4.6)$$

Pentru a produce LBP pentru pixelul  $(x_c, y_c)$ , o greutate binomică  $2^p$  este atribuită fiecărui  $s(g_p - g_g)$  și aceste ponderi sunt însumate. (4.7)

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (4.7)$$

- Alegerea dintre LBP uniform sau neuniform

Un model binar local este numit uniform dacă conține cel mult două tranziții de biți de la 0 la 1 sau invers. De fapt, aceasta înseamnă că modelul uniform nu are tranziții sau două tranziții. Doar o singură tranzație nu este posibilă, deoarece sirul binar este considerat circular. Cele două modele cu zero tranziții pentru 8 biți sunt: 00000000 și 11111111. Pentru modele cu două tranziții sunt  $P(P-1)$  combinații posibile (Figura 4.10). Folosirea LBP-ului uniform are două beneficii importante față de cel neuniform. Primul este că se salvează multă memorie, deoarece folosind metoda neuniformă sunt  $P^2$  combinații posibile, iar cu uniformă doar  $P(P-1) + 2$ . Un alt beneficiu este că LBP uniform detectează doar texturile locale importante, cum ar fi pete, linii, muchii, colțuri.

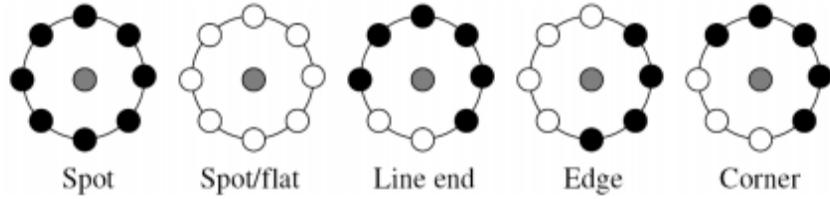


Figura 4.10: Tipuri de tranzacții a biților [13]

- Construirea histogramei

Histograma va avea lungimea de  $P(P-1) + 2$ , care reprezintă numărul de combinații posibile pentru LBP uniform și +1, care reprezintă caracteristicile excluse căci vectorul binar are mai mult de 2 tranzații de biți de la 1 la 0 sau invers. Pentru fiecare pixel, după ce se calculează valoarea zecimală din vectorul obținut pe baza comparațiilor pixelului curent cu vecini, se va incrementa poziția histogramei cărui corespunde combinația binară. Pentru o acuratețe mai mare, este posibilitatea să se împartă imaginea în mai multe imagini și fiecare va avea histograma sa. La final prin concatenarea tuturor histogramelor, se va obține histograma întregii imagini (Figura 4.11).

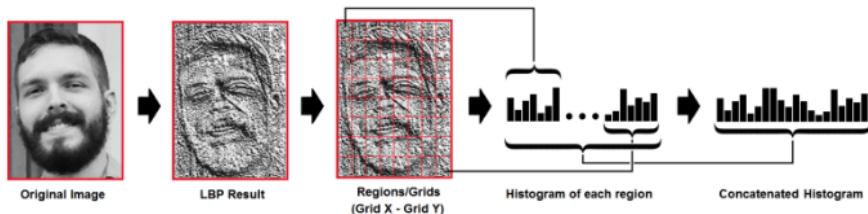


Figura 4.11: Algoritmul LBP prin împărțirea imaginii în mai multe regiuni [13]

- Compararea histogramelor

Ultimul pas al acestui algoritm pentru recunoașterea facială, este compararea histogramei obținute cu toate histogramele existente. Histograma imaginii de referință stocată care va fi cel mai aproape de trăsăturile histogramei curente de test va fi aleasă ca răspusul sistemului de recunoaștere. Sunt mai multe metode de găsirea histogramei cu care se aseamană cel mai mult.

1. Intersecția histogramei

$$D(S, M) = \sum_{i=1}^N \min(S_i, M_i) \quad (4.8)$$

2. Statistica likelihood

$$L(S, M) = \sum_{i=1}^N S_i \log \cdot M_i \quad (4.9)$$

3. Statistica Chi pătrată

$$X^2(S, M) = \sum_{i=1}^N \frac{(S_i - M_i)^2}{S_i + M_i} \quad (4.10)$$

4. Distanța Euclidiană

$$D(S, M) = \sqrt{\sum_{i=1}^N (S_i - M_i)^2} \quad (4.11)$$

## 4.2 Compararea Algoritmilor

Recunoașterea facială a devenit un domeniu foarte important în mai multe ramuri. Aceasta a influențat și apariția a mai multor algoritmi și moduri de recunoaștere a feței. Algoritmii descriși mai sus au o precizie mare de recunoaștere. Fiecare având tehnica sa de recunoaștere, precizia depinde mult de locul unde este folosit sistemul și factorii existenți (iluminare, umbră, claritatea imaginii, emoția facială, poziția feței). Pentru a vedea factorii care influențează acuratețea imaginii și care algoritm se descurcă mai bine în diferite momente vom analiza comparația algoritmilor de recunoaștere facială, făcută de Nicolas Delbiaggio în 2017 [23]. Algoritmii testați sunt: Eigenfaces, Fisherfaces, Local Binary Pattern, OpenFaces. Toate fotografile din datele de referință au fost luate cu o cameră Nikon D3100. Ele au fost pre-procesate în același mod. Fețele au fost detectate cu HOG. Testul va fi rulat cu două tipuri de date. În primul, datele de referință ale unei persoane vor fi în același mediu cu datele de testare. Înseamnă că fotografile sunt luate în aceeași cameră și cu aceeași lumină. În cel de-al doilea, fotografile de la referință și datele de testare nu vor proveni din același mediu. Pozele ar putea fi luate într-o altă locație și unde luminozitatea este variabilă. În a doua fază se va vedea reacția algoritmilor cu imagini din alt loc. Robustetea algoritmilor vor fi testate în acest fel.

- Testare folosind același mediu de antrenare și testare (Figura 4.12)

La prima testare, mai multe poze de referință fac acuratețea mai mare. Pentru 10 persoane, toți patru algoritmi au acuratețea de 100%, dacă sunt folosite 40 de imagini de referință. Acuratețea este mai mică dacă se folosesc mai puține imagini pentru antrenarea sistemului sau numărul de persoane. Chiar și folosind 40 de imagini de referință, acuratețea a scăzut la Fisherfaces pentru 15 persoane.

## 4.2. COMPARAREA ALGORITMILOR

35

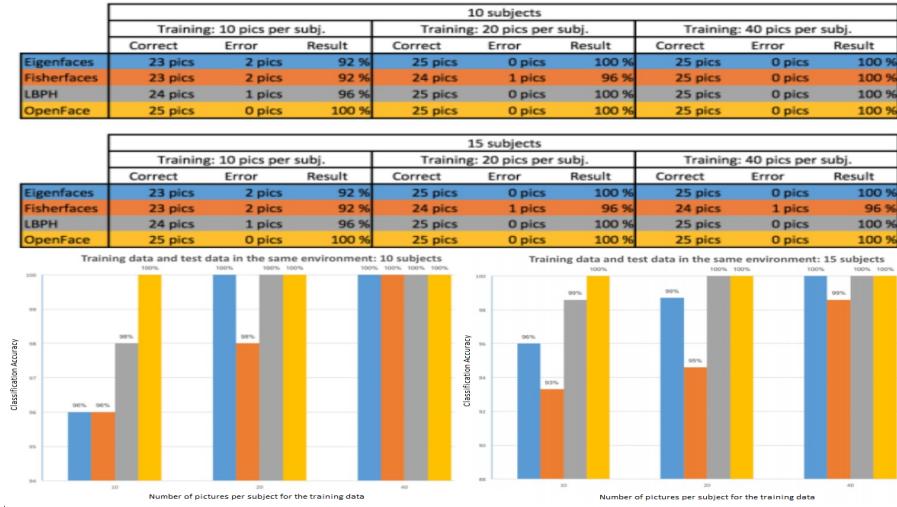


Figura 4.12: Același mediu de antrenare și testare [23]

- Folosind medii diferite (Figura 4.13)

Schimbarea mediului a afectat foarte mult acuratețea algoritmilor. Cel mai puțin afectat a fost cel de OpenFace, care totuși a pierdut din acuratețe pentru 15 persoane.

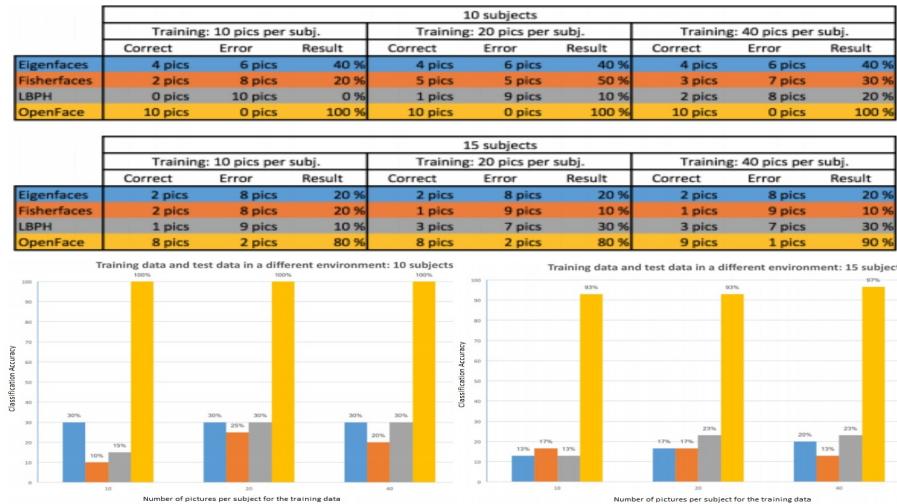


Figura 4.13: Medii diferite pentru testare și antrenare

Acest experiment a demonstrat că acești algoritmi au o acuratețe bună, dar sistemul fiind influențat de mai mulți factori fiecare algoritm a fost afectat în moduri diferite.

- Eigenfaces - nu a fost foarte sensibil la o schimbare a numărului de persoane și creșterea numărului de persoane a crescut acuratețea. Nu a avut rezultate foarte bune în a doua fază.

- Fisherfaces - a avut rezultate mai bune cu număr mai mare de date. Cu toate acestea, a avut comportament diferit în faza a doua. Uneori cu 20 de poze, a avut rezultate mai bune decât cu 40. Deobicei, în ambele faze a avut cea mai mică acuratețe.
- LBP - a avut o acuratețe foarte mare în prima fază, cel puțin 96%, dar a avut o scădere consecventă în a doua etapa.
- OpenFace - a avut cele mai bune rezultate, în majoritatea cazurilor având o acuratețe de 100%. A avut câteva erori la a doua etapa. OpenFace a fost afectată puțin de o schimbare a mediului înconjurător.

Pe lângă faptul că acuratețea de recunoaștere facială este importantă, un factor important este viteza de recunoaștere. Acest tip de sistem fiind folosit de multe firme cu motive diferite, preferă ca timpul de recunoaștere să fie foarte rapid, altfel ar provoca ceva inconfortabil pentru utilizatori. Ministerul Științei, Industriei și Tehnologiei din Turcia a făcut o testare pentru timpul de execuție și acuratețea algoritmilor Eigenfaces, Fisherfaces și LBP [24]. Ambele metode Eigenfaces și Fisherfaces sunt dintre cele mai cunoscute tehnici de recunoaștere a feței. Ele sunt cunoscute a fi foarte sensibile la variațiile de nivel ale pixelilor, cum ar fi iluminarea, expresia facială și prezintă variații. LBP este o metodă relativ nouă dezvoltată în primul rând pentru analiza texturii. Comparață cu Eigenfaces și Fisherfaces, s-a constatat că este semnificativ mai robustă la iluminare și posedă variații. Au fost făcute 2 experimente, în experimentul 1, o piesă a feței este aleasă ca set de referință și restul ca set de test. În experimentul 2, 60% din imagini sunt alese ca set de referință și 40% ca test set. Aceiași algoritmi sunt rulați pe platforme diferite, pentru a se vedea variația performanței și cum algoritmii beneficiază de hardware. S-au folosit două platforme diferite. Prima platformă este Intel bazat pe procesor de 2,7 GHz cu 4 GB RAM, etichetat ca Intel. Platforma a doua este ARM cu procesor de 1 GHz 512 MB RAM, etichetat ca Arm. Este folosită librăria OpenCV pentru procesarea de imagini, Microsoft Visual C++ 2010 Express Edition folosit pentru implementare și compilare la sistemul bazat pe Windows și g++ folosit pentru a compila codul pe arm-Linux.

- Extragerea fețelor

Au fost extrase fețele din Serialul Buffy, sezonul 5, episodul 2 și fețele a trei caracterere principale au fost folosite în experimente (Figura 4.14). Buffy, Michelle și Nicholas sunt personajele alese. Nicolae este ales să adauge cel puțin un caracter masculin la teste, deoarece algoritmii au o recunoaștere mai bună în cazul barbaților.

<b>Characters</b>	<b>Experiment 1</b>		<b>Experiment 2</b>	
	<i>Train(# of face images)</i>	<i>Test(# of face images)</i>	<i>Train(# of face images)</i>	<i>Test(# of face images)</i>
Buffy	14	739	451	302
Michelle	18	606	375	249
Nicholas	13	192	123	82

Figura 4.14: Fețele extrase din Serialul Buffy [24]

- Testarea duratei de timp (Figura 4.15)

Eiginfaces este cel mai rapid la primul experiment pe Intel, unde și Fisherfaces a dat un rezultat bun în 5.96 secunde. Per total, LBP este cel mai robust și are cel mai bune rezultate, mai ales pentru experimentul 2, care este aplicat cel mai mult în viață reală, deoarece sunt folosite mai multe imagini de antrenare decât cele testate.

<b>Methods</b>	<b>Experiment 1</b>		<b>Experiment 2</b>	
	<i>Intel (seconds)</i>	<i>Arm (seconds)</i>	<i>Intel (seconds)</i>	<i>Arm (seconds)</i>
Eigenfaces	1.91	30.65	432.08	3600
Fisherfaces	5.96	8.3	549.84	3360
LBPH	55.02	136.52	143.61	907.02

Figura 4.15: Rezultatul vitezei de recunoaștere [24]

- Testarea acurateței (Figura 4.16)

LBP oferă și cel mai bun rezultat al acurateței, în ambele experimente. Se vede foarte bine că dacă există puține imagini de antrenare, acuratețea este mult mai mică la toți cei 3 algoritmi.

<b>Methods</b>	<b>Experiment 1</b>			<b>Experiment 2</b>		
	<i>True</i>	<i>False</i>	<i>Hit Ratio (%)</i>	<i>True</i>	<i>False</i>	<i>Hit Ratio (%)</i>
Eigenfaces	509	1028	33	586	47	93
Fisherfaces	519	1018	34	540	93	85
LBPH	679	858	44	594	39	94

Figura 4.16: Rezultatul acurateței de recunoaștere [24]

### 4.3 Fuziunea algoritmilor

Recunoașterea facială fiind un domeniu foarte larg, există foarte mulți algoritmi și fiecare are punctele sale forte și ceva minusuri. Un algoritm este orientat pe anumite caracteristici ale imaginii și depinde mult de factorii care influențează imaginea. Diferite fețe au multe caracteristici și studiile recente au arătat că o metodă bună de a crește acuratețea de recunoaștere este folosirea a mai multor caracteristici. Între timp, mulți cercetători au dezvoltat diverse metode de recunoaștere pentru a coopera cu caracteristici multiple. Una din soluțiile pentru a analiza mai multe caracteristici a unui imagine care conține o față, este fuziunea a mai multor algoritmi de recunoaștere facială. Algoritmii aleși pentru implementarea acestui sistem, Histogram of Oriented Gradients și Local Binary Patterns s-au dovedit ca niște descriptori eficienți pentru recunoașterea feței, dar ambii având ceva puncte slabe se va face o fuziune (Figura 4.17) pentru obținerea unor rezultate mai bune.

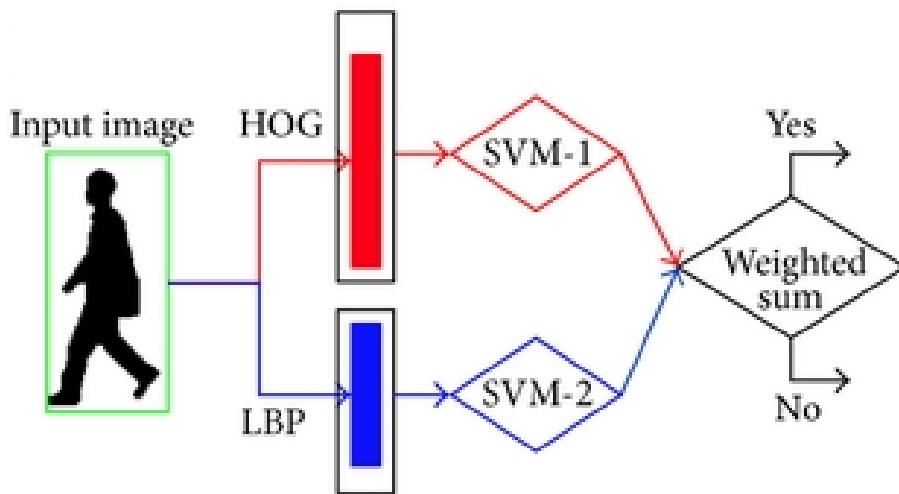


Figura 4.17: Fuziunea algoritmilor HOG și LBP

Local Binary Patterns sunt o limitare la extragerea informațiilor de margine și direcție, care este un punct foarte important pentru recunoașterea facială. Histogram Oriented Gradients fiind mai mult orientat pe detecția persoanelor nu are o acuratețe foarte bună la recunoașterea facială. În mare parte, HOG este folosit la fuziuni cu alți algoritmi, fiind axat pe extragere de margini și direcții. Pentru a vedea îmbunătățirea acurateței de recunoaștere facială, se vor analiza aceste experimente:

- Recunoaștere facială folosind sistemul robust HOG și LBP de la Universitatea Shahid Behshti [25]

Acest studiu arată cum fuziunea va crește acuratețea față de HOG. Toate imaginile au fost normalizate la dimensiune de 58x50 pixeli. Pentru prima testare a fost folosită baza de date Wild(LFW) (Figura 4.18) [26]

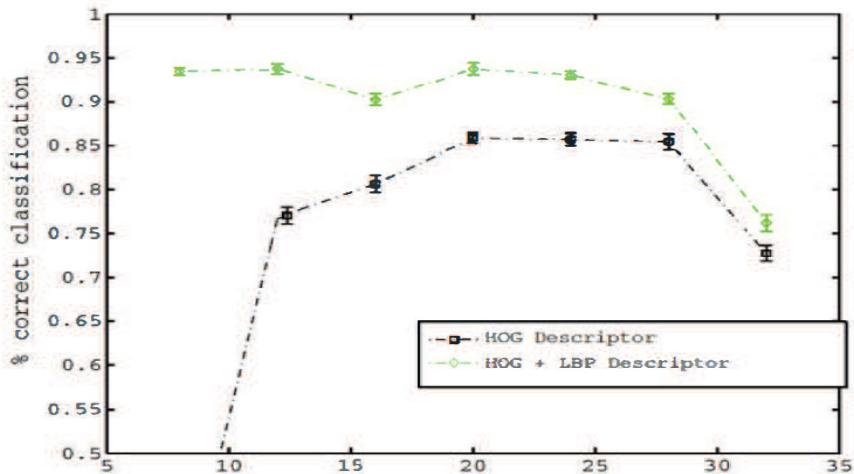


Figura 4.18: Experiment folosind baza de date Wild(LFW) [25]

Pentru al doilea test s-a folosit baza de date AR (Figura 4.19) [27]

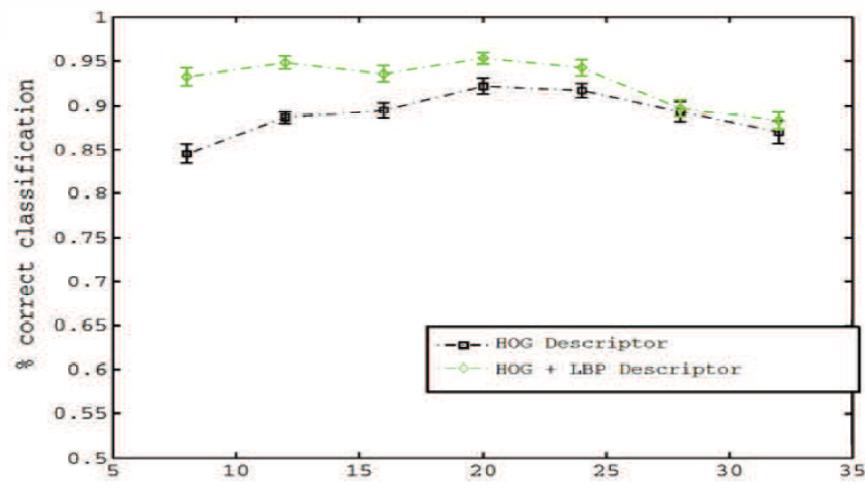


Figura 4.19: Experiment folosind baza de date AR [25]

Se observă clar că procentajul de recunoaștere crește folosind fuziunea algoritmilor. Rezultatele obținute diferă la cele două baze de date, fiindcă diferă imaginile, numărul de imagini per persoană, calitatea și mediul în care sunt făcute.

- Fuziunea LBP și HOG utilizând nuclee multiple pentru recunoașterea feței în infraroșu de la Universitatea Nanchang, Jiangxi din China. [28]  
Imaginiile cu infraroșu din baza de date au fost capturate cu ThermoVision A40 realizat de FLIR System INC. Această cameră este special concepută pentru o termper-

atură precisă de măsurare. Baza de date conține 800 de imagini termice de 40 de persoane, care au fost colectate atent la temperaturi ambiante variabile, cu temperatură de la  $18.2^{\circ}$  C la  $29.6^{\circ}$  C. Rezoluția originală a imaginii este de  $240 \times 320$ , iar a imaginii cu față se transformă în  $60 \times 80$  după detectarea feței și normalizarea geometrică. În acest experiment, parametrii (P,R) LBP sunt (8,1) și caracteristicile din histogramă se compară folosind statistica chi-square pentru HOG și LBP. Procentajul cu care fuziunea algoritmilor a crescut acuratețea de recunoaștere poate fi văzută în Figura 4.20.

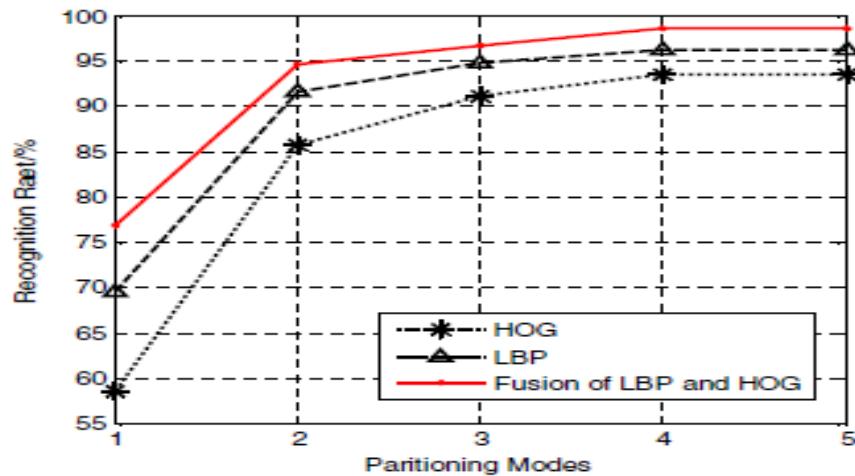


Figura 4.20: Fuziune HOG și LBP [28]

Contribuțiile principale ale fuziunii algoritmilor HOG și LBP sunt:

- S-au compensat erorile din detectarea caracteristicelor feței datorită ocluziilor, poziției și modificările iluminării folosind HOG.
- Fuziunea descriptorilor HOG la scări diferite cu cele LBP permit capturarea unei structuri importante pentru recunoașterea facială
- Este importantă realizarea selecției de caracteristici pentru a elimina caracteristicile redundante și irelevante. Asta face procesul de clasificare mai puțin predispus la suprasolicitare.

#### 4.4 Alegerea dispozitivelor și tehnologiilor

Algoritmii de detectie și recunoaștere facială joacă rolul cel mai principal la acuratețea sistemului. De asemenea, influențează viteza de oferire a răspunsului, care depinde foarte mult de dispozitivul pe care funcționează sistemul. Chiar dacă în capitolul 3 s-a

specificat că FPGA are o viteză mare de execuție a algoritmilor, deoarece spre deosebire de un procesor, care stochează și execută instrucțiuni, programarea unui FPGA înseamnă reconfigurarea hardware a acestuia pentru a realiza funcționalitatea dorită. Acest dispozitiv nu a fost ales pentru acest sistem, deoarece folosind un algoritmul Haar Cascade de detecție a feței și fuziunea algoritmilor HOG și LBP pentru recunoaștere feței, implementarea lor ar fi ceva foarte complex pentru FPGA și ar lua foarte mult timp. Pentru implementarea, testarea și dezvoltarea acestui sistem au fost alese:

1. Laptop ASUS, Intel(R) Core(TM) i7-4710HQ CPU @2.50GHz, cu 4,00 GB memorie RAM
2. Raspberry PI 3, Quad Core 1.2GHz Broadcom, 64bit CPU, 1GB RAM
3. Raspberry PI Camera Module V2
4. Monitor LED IPS Dell 23.8
5. Arduino
6. Afisor 16x2 LCD

Laptopul are ca scop să fie folosit pentru testarea algoritmilor, crearea unor testări pentru detectarea acurateței și compararea vitezei de execuție cu cea a sistemului de Raspberry PI 3. Acest dispozitiv are scopul ca sistemul să funcționeze mereu, primind imaginea de la cameră, să trimită răspunsul de recunoaștere pe afișorul LCD 16x2 prin intermediul la Arduino. Raspberry PI 3 a fost ales ca dispozitivul principal pentru acest sistem, fiind la un preț convenabil, ușor de programat și posibilitatea de a analiza viteza pentru același algoritm folosind CPU-ul de la laptop. De asemenea, un factor important este posibilitatea de a conecta alte dispozitive ca sistemul să fie cât mai ușor de folosit de utilizatori. Ei să fie informați că față lor a fost detectată și să vadă răspunsul de recunoaștere a sistemului afișat. Pentru implementarea acestui sistem a fost ales Python, un limbaj de programare de nivel înalt. Are ca avantaje că poate fi folosit ușor pe mai multe sisteme de operare, este ușor de utilizat și oferă o viteză mare de care sistemul de recunoaștere facială are nevoie. De asemenea, este folosită și librăria OpenCV care este axată pe procesarea de imagini, care joacă cel mai important rol la implementarea algoritmilor de recunoaștere.

# Capitolul 5

## Proiectare de Detaliu și Implementare

Scopul major al implementării acestui sistem este o acuratețe de recunoaștere foarte mare și oferirea răspunsului într-un timp scurt. Această aplicație va avea 2 metode de a fi utilizată de două tipuri diferite de utilizatori. Chiar dacă are ca prioritate recunoașterea persoanelor care doresc să intre în sediu este nevoie și de stocarea imaginilor noi pentru toți angajații.

### 5.1 Fluxul de recunoaștere a persoanelor

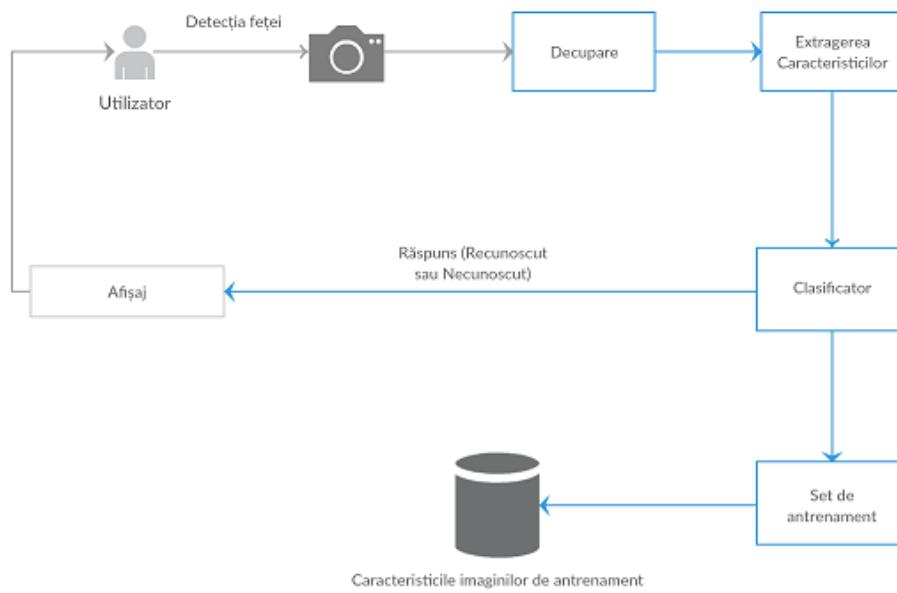


Figura 5.1: Fluxul de recunoaștere

În Figura 5.1 se vede fluxul sistemului pentru recunoașterea unei persoane. Camera va fi fixată la intrarea în sediu și persoana va fi informată că trebuie să se poziționeze în fața camerei. Camera va face o poză și va fi transmisă mai departe către sistem. Se va face detectia feței, dacă nu este detectată utilizatorul va primi acest mesaj, altfel imaginea va fi decupată și va ajunge la algoritmii de recunoaștere facială. Se vor extrage vectorii de caracteristici ai imaginii și vor fi comparați de clasificator cu toți ceilalți vectori de caracteristici ale angajaților. Persoana va fi informată dacă este recunoscută sau nu.

### 5.1.1 Capturarea imaginilor și detectia feței

Pentru căpturarea imaginilor este folosită un Raspberry Pi Camera Module V2. Camera va face imagini la fiecare 0.1 secunde și se execută detectia feței. Dacă nu este nimeni în fața camerei sau persoana nu are poziția potrivita, camera va continua să facă imagini și să aplice detectia feței. După cum a fost specificat și în capitolul precedent, pentru detectia feței este folosit algoritmul Haar Cascades. Metoda responsabilă de acest lucru este *faceDetection()*, care primește ca variabilă de intrare imaginea făcută de cameră. Pași execuția în această metodă sunt:

- Se încarcă clasificatorul *haarcascade\_frontalface\_default.xml*, responsabil de detectia feței frontale
- Imaginea se transformă în grayscale
- Se aplică metoda *detectMultiScale()* pentru găsirea coordonatelor imaginii feței
- Se decupează imaginea pe baza coordonatelor obținute și este returnată

Pentru a fi mai clar fluxul sistemului și lucrul care îl face fiecare parte, vom folosi ca test imaginea din stânga Figurii 5.2. Presupunem că este imaginea capturată de cameră și transformarea care o face funcția *faceDetection()* se poate vedea în figura de mai jos.

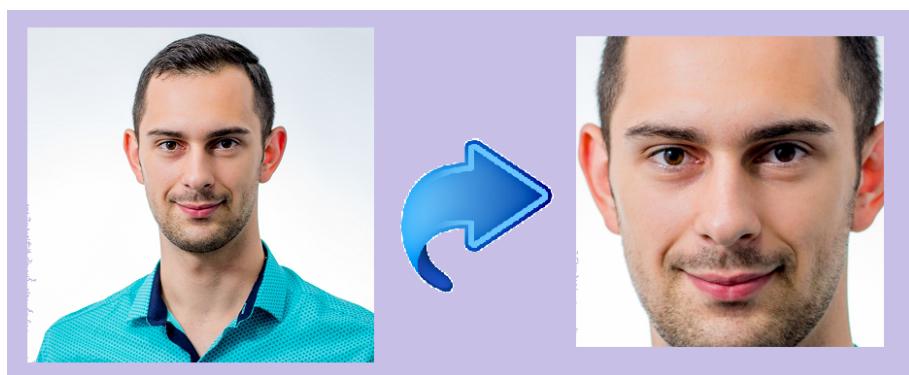


Figura 5.2: Detectia feței și decuparea pozei

Obiectivele de folosire a detecției feței sunt:

1. Va detecta când există o persoană în fața camerei și doar atunci va aplica algoritmii de recunoaștere facială
2. Decuparea reduce numărul de pixeli ai imaginii, ceea ce va mări viteza de recunoștere
3. Decuparea elimină fundalul din imagine, ceea ce va îmbunătăți acuratețea de recunoaștere a persoanelor
4. Utilizatorul sistemului va putea să mai clar dacă sistemul îi face recunoașterea sau nu a avut o poziție bună de a fi detectat

### 5.1.2 Extragerea Caracteristicilor

La această etapă a sistemului se ajunge doar dacă este detectată fața unei persoane. Extragerea caracteristicilor imaginii este cea mai importantă parte a recunoașterei persoanei. Pentru creșterea acurateței acestui sistem sunt folosiți doi algoritmi de recunoaștere Histogram of Oriented Gradients și Local Binary Pattern. Metodele responsabile de extragere a caracteristicilor pentru ambii algoritmi primesc ca intrare imaginea decupată și caracteristicile sunt stocate într-o histogramă.

#### Histogram of Oriented Gradients

Metoda responsabilă de această funcție este *histogramOfGradients()* și pașii sunt:

1. Calcularea gradientilor orizontali(5.1) și verticali(5.2) prin utilizarea operatorului Sobel în OpenCV cu mărimea nucleului 1 (Figura 5.3).

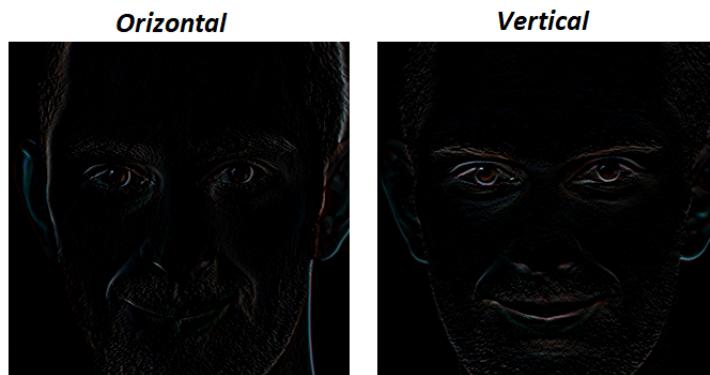


Figura 5.3: Gradientii

$$gx = cv2.Sobel(img, cv2.CV_32F, 1, 0, ksize = 1) \quad (5.1)$$

$$gy = cv2.Sobel(img, cv2.CV_32F, 0, 1, ksize = 1) \quad (5.2)$$

2. Se calculează magnitudinea și gradientul de direcție (5.3), folosind gradienții orizontali și verticali Figura(4.4).

$$mag, angle = cv2.cartToPolar(gx, gy, angleInDegrees = True) \quad (5.3)$$

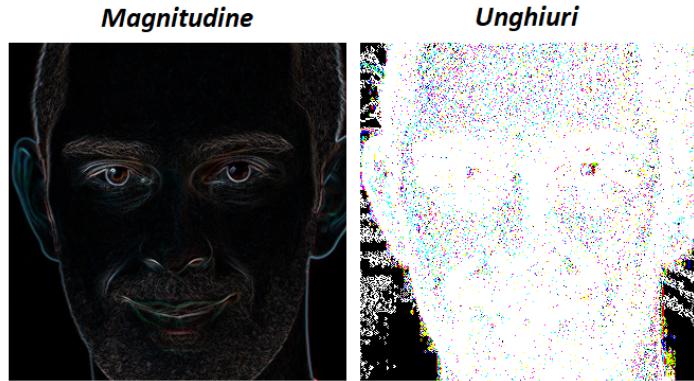


Figura 5.4: Magnitudine și unghiuri

3. Se construiește histograma care este un vector de 9 poziții corespunzătoare unghilor 0,20,40 ... 160.

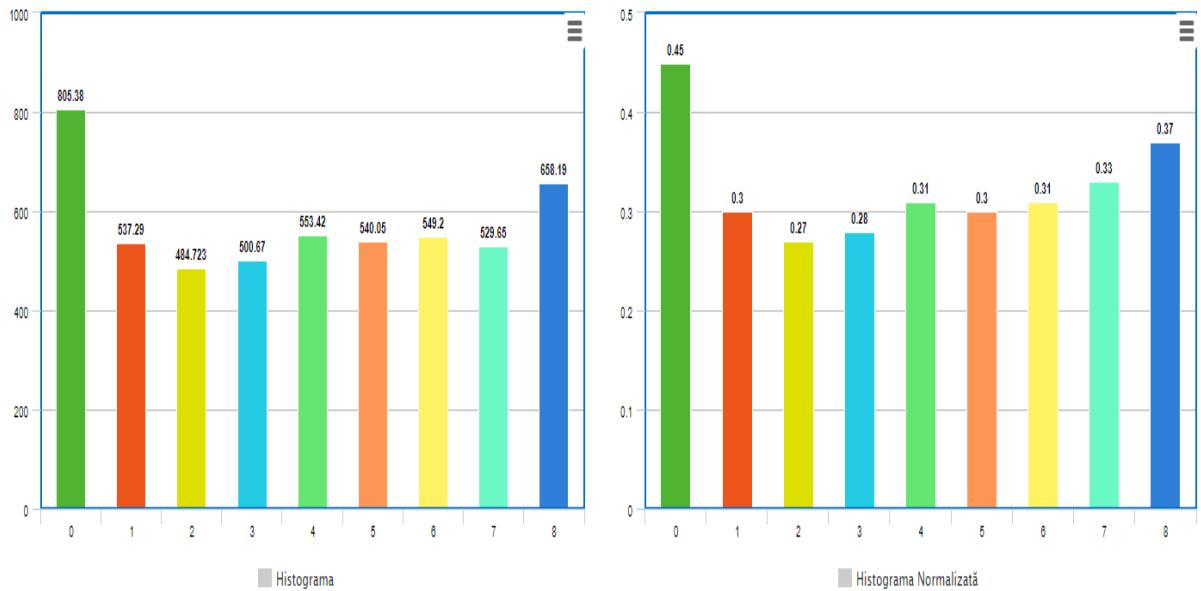


Figura 5.5: Histograma imaginii folosind HOG

Matricea cu unghiuri conține și valori până la 360 pentru asta se folosește un vector cu pozițiile coșului la care trebuie adăugată magnitudinea. Unghiurile mai mari

decât 180 sunt luate ca  $360^\circ$  - valoarea unghiului. Histograma obținută este normalizată pentru a crește acuratețea de recunoaștere. Normalizarea imaginii va aduce valorile histogramei în intervalul 0 și 1, independent de dimensiunile imaginilor dividerea fiecărui element din histogramă cu radicalul din suma pătratelor a celor 9 valori din histogramă (5.4). În Figura 5.5 se poate vedea histograma normală și cea normalizată obținută pentru poza folosită ca test.

$$h_i = \frac{h_i}{\sqrt{h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 + h_7^2 + h_8^2}} \quad (5.4)$$

### Local Binary Pattern

Metoda responsabilă de această funcție este `lbpHistogram()` și pașii sunt:

1. Se transformă imaginea în grayscale (Figura 5.6), deoarece algoritmul este axat pe compararea valorilor de 0 și 1.

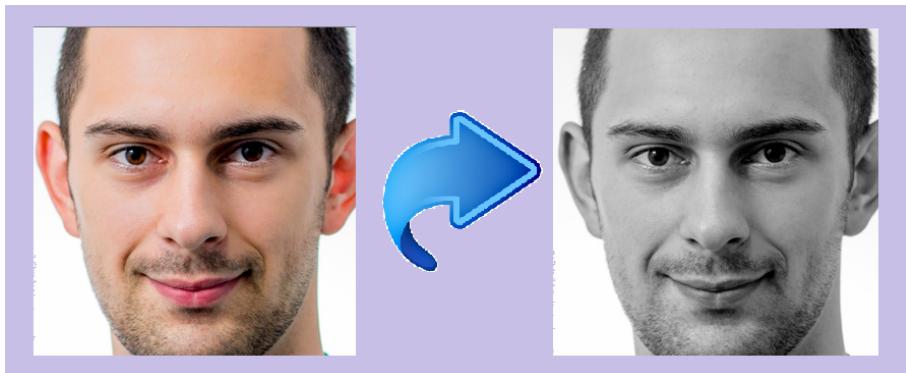


Figura 5.6: Transformare în grayscale

2. A fost ales tipul de vecinătăți  $P = 1$  și  $R = 8$ , ceea ce înseamnă că vectorul de biți este construit din cei 8 vecini aflați la o distanță egală cu 1 de pixelul curent. Histograma va avea o lungime  $= P * (P - 1) + 3 = 59$ .
3. Vecinii vor fi găsiți fiind calculați cu sinus și cosinus,  $(x_c, y_c)$  fiind coordonatele pixelului central (5.5).

$$x_p = x_c + R \cdot \cos(2\pi/P) \quad y_p = y_c + R \cdot \sin(2\pi/P) \quad (5.5)$$

4. După ce se calculează valoarea zecimală din vectorul obținut pe baza comparațiilor pixelului curent cu vecini, se va incrementa poziția histogramei cărui corespunde combinația binară. Pentru a ști poziția histogramei cărei corespunde, este folosit un vector deja construit pe bază numărului de biți și lungimea histogramei.

5. Se obține histograma și la fel ca la algoritmul de HOG este normalizată pentru a crește acuratețea de recunoaștere. (Figura 5.7)

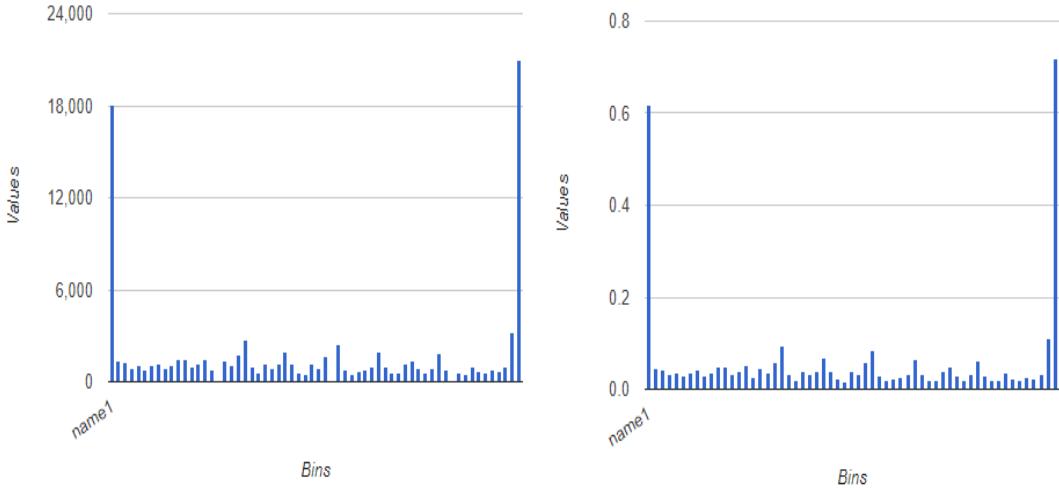


Figura 5.7: Histograma imaginii folosind LBP

### 5.1.3 Clasificator

Pentru ca imaginea persoanei care stă în fața camerei să fie recunoscută, trebuie să existe mai multe imagini pentru angajații firmei. Pentru toate persoanele din companie au fost făcute mai multe imagini și au fost aplicăți algoritmii de extragere a caracteristicilor care au fost stocate într-o formă tabelară, folosind pentru asta excel. Pe fiecare rând este stocată câte o histogramă pentru fiecare imagine, iar prima celulă este numele persoanei. Caracteristicile HOG și LBP a imaginii care se află în procesul de recunoaștere vor fi comparate cu toate caracteristicile angajaților stocate. Există mai multe metode de comparare a histogramelor, iar pentru acest sistem este folosită distanța Euclidiană (5.6).

$$D(S, M) = \sqrt{\sum_{i=1}^N (S_i - M_i)^2} \quad (5.6)$$

Se ia top 3 poze cu cea mai mică distanță dintre histograma sa cu cea a imaginii curente. Există 6 voturi, câte 3 de la fiecare algoritm. Chiar dacă o imagine este votată cel mai mult, nu va însemna neapărat că persoana a fost recunoscută. Conform unei testări au fost stabilite niște rezultate maxime și minime obținute (5.7) în urma calculării distanței pentru fiecare algoritm (5.7).

$$\maxHog = 0.3 \quad \minHog = 0.2 \quad \maxLbp = 0.2 \quad \minLbp = 0.1 \quad (5.7)$$

Rezultatul va fi obținut din următorii pași:

1. Dacă imaginile de pe primul loc la ambii algoritmii corespund la aceeași persoană și distanța celui de la HOG < maxHog și celui de la LBP < maxLbp persoana este recunoscută cu numele acestui angajat
2. Dacă o imagine de la HOG < minHog sau de la LBP < minLbp și este pe primul loc la Vot, acesta va fi rezultatul recunoașterii
3. Chiar dacă nu sunt îndeplinite nici una din precedentele reguli și există o persoană cu 3 voturi, ea va fi rezultatul
4. Dacă nu este sătisfăcută nici o regulă, persoana va primi rezultat negativ de recunoaștere.

#### 5.1.4 Afișarea mesajului pentru utilizator

Pentru a oferi comoditate utilizatorului este folosit un Arduino, care este conectat cu Raspberry PI 3. La arduino este legat un Afișaj LCD pe care va fi afișat mesajul și de asemenea un board la care sunt conectate 3 leduri(roșu, galben, albastru). Legătura între aceste 2 dispozitive este făcută cu librăria *pyfirmata* din care sunt importate *STRING\_DATA*, *util*. Cu instrucțiunea (5.8) se trimit mesajul care trebuie afișat pentru utilizator.

```
board.send_sysex(STRING_DATA, util.str_to_two_byte_iter("Message")) (5.8)
```

- Mesajul = "Waiting", înseamnă că sistemul este în aşteptarea unei persoane și nici o față nu a fost detectată. Nici un led nu va fi conectat
- Mesajul = "Face Detected", înseamnă că sistemul a detectat fața persoanei și este în proces de recunoaștere. Va fi pornit ledul de culoare galbenă
- Mesajul = "Hello Employee\_Name", sistemul a recunoscut persoana și o salută cu numele corespondent. Va fi conectat ledul de culoare albastră
- Mesajul = "Not Recognized", persoana nu a fost recunoscută și ledul de culoare roșie va fi pornit

## 5.2 Fluxul de stocare a imaginilor noi

Un alt rol important al sistemului este stocarea de noi imagini a angajaților și calcularea caracteristicilor folosind cei doi algoritmi și salvarea lor. Vor fi făcute poze noi a angajaților, cu cât mai multe poziții, emoții, vestimentație astfel încât să crească acuratețea de recunoaștere. Toate imaginile care vor într-un anumit fișier vor fi parcuse și se va aplica algoritmul Haar Cascades pentru detecția feței. Cele care nu vor fi detectate vor fi șterse cu metoda `deleteImages()`. După se va aplica algoritmii de extragere a caracteristicilor care au fost descriși mai sus și toți vectorii vor fi stocați în formă tabelară, prima coloană fiind numele persoanei, care va fi luat pe bază la numele fișierului din care se parcuse imaginile. Acest flux de date este reprezentat în Figura 5.8.

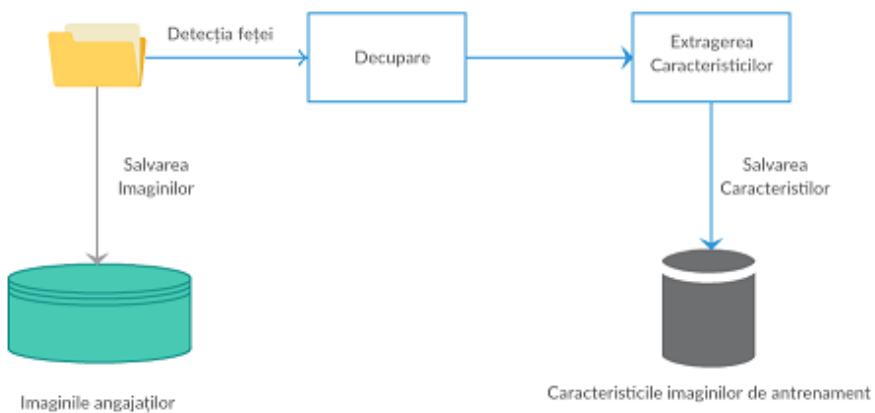


Figura 5.8: Fluxul de stocare a imaginilor

## 5.3 Eficientizarea sistemului

Pentru îmbunătățirea rezultatelor pe care este axat sistemul și sunt așteptate de conducerea companiei (ofering unei securități majore informației interne) și de utilizatori (comoditate de folosire și o viteză mare de recunoaștere) au fost efectuate următorii pași de dezvoltare a sistemului:

### 1. Decuparea imaginii în 4 regiuni

Pentru creșterea procentajului de recunoaștere a persoanelor, imaginea care conține față decupată este împărțită în 4 regiuni. Ambii algoritmi de recunoaștere vor fi aplicati pe toate 4 imagini și histograma întregii imagini va fi obținută prin concaterea celor 4. Histograma algoritmului Histogram of Oriented Gradients va avea 36 bins și cea a Local Binary Pattern 237 bins. Împărțirea imaginii folosite în 4 regiuni poate fi văzută în Figura 5.9. Aceasta crește acuratețea de recunoaștere, deoarece este făcută

câte o histogramă pentru fiecare. Nu sunt comparate histogramele imaginii întregi, dar fiecare parte a feței.

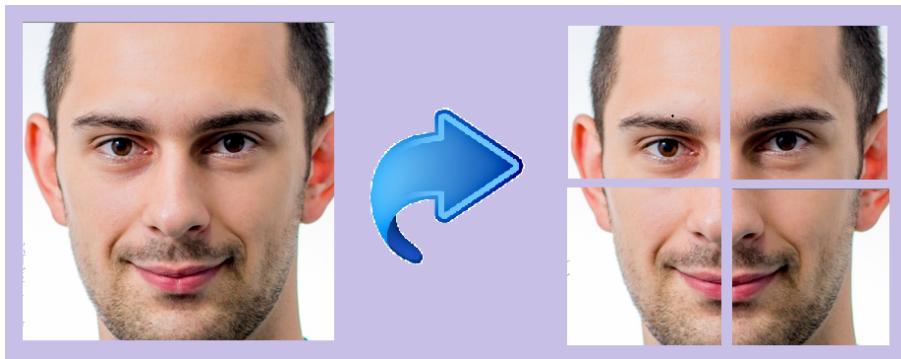


Figura 5.9: Împărțirea imaginii în 4 regiuni

## 2. Folosirea programării paralele

Dacă unele imagini sunt mari, aceasta poate să ia foarte mult timp pentru extragerea caracteristicilor. Pentru a oferi o viteză mai mare se folosesc cele 4 procese de la Raspberry PI 3. Sunt create 4 procese și o coadă pentru HOG și LBP. Fiecare proces este responsabil de calcularea caracteristicilor pentru una din regiuni și când finalizează, acea histogramă este pusă în coadă. În metodele de *lbpFaceRecognition()* și *hogFaceRecognition()* se extrag histogramele pentru cele 4 regiuni și sunt concatenate. Aceasta duce semnificativ la îmbunătățirea vitezei de recunoaștere a sistemului.

## 3. Calcularea și stocarea caracteristicilor

Sistemul nu are ca scop să compare direct imaginea care se află în procesul de recunoaștere cu celealte imagini, dar totul este axat pe vectorii de trăsături. Ar lua foarte mult timp dacă la testarea unei imagini vor trebui calculați toți vectorii de caracteristici a pozelor angajaților. Din acest motiv, pentru pozele noi făcute sunt calculați vectorii de trăsături și sunt stocați într-un fișier text de tip excel. Fiind foarte mulți angajați și fiecare având cate 5-15 poze, extragerea lor din excel ia ceva timp. Pentru ca utilizatorii să nu fie afectați de acest lucru, datele sunt extrase din excel la pornirea excel-ului și utilizate pentru recunoașterea persoanelor care se pun în fața camerei.

## 4. Refactorizarea codului pentru dezvoltarea vitezei

### (a) Eliminarea înmulțirii pentru numere float

În algoritmul HOG de recunoaștere se folosea o înmulțire float pentru fiecare pixel din imagine. Aceasta adăuga câteva secunde la execuție și a fost eliminată. Este folosit un round al numărului ceea ce scade puțin din acuratețea de recunoaștere dar se execută mult mai rapid.

- (b) Eliminarea recalculatei anumitor valori

Unii vectori și valori care trebuie folosiți de algoritmi HOG și LBP sunt calculați din prima și folosite ca attribute. Dacă pentru fiecare apelare a sistemului vor fi calculate din nou, aceasta va mări timpul de execuție. La LBP sunt stocate numărul de vecini, lungimea histogramei, vectorii de sinus și cosinus, vectorul pentru a căre histogramă este incrementată conform valorii vectorului de biți și chiar vectorul de  $2^0$  pana la  $2^7$ . De asemenea, la algoritmul HOG este stocat un vector care reprezintă poziția bin-ului care trebuie incrementat în dependență de valoarea unghiului.

- (c) Evitarea unor calcule complexe și diferite instrucțiuni de if, care pot fi făcute mai rapid

## 5.4 Rezultate finale

În urma aplicării de fuziune a algoritmilor și refactorizare a codului, în Figura 5.10 putem vedea schema finală a sistemului.

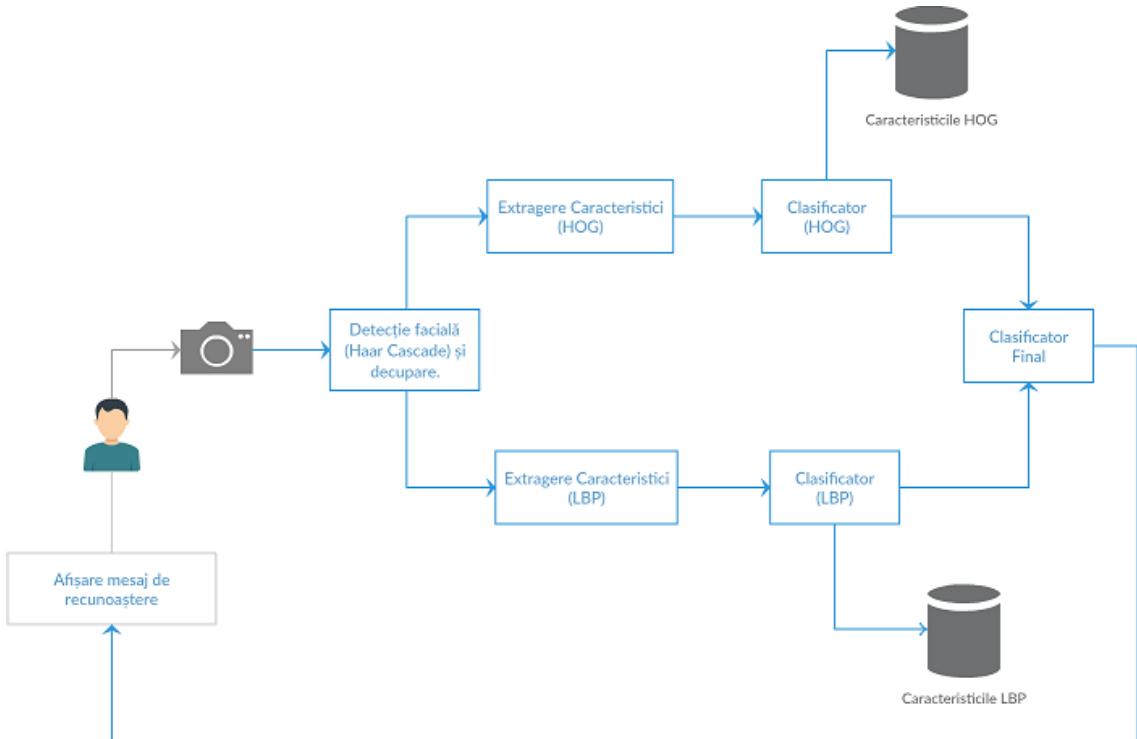


Figura 5.10: Schema Finală

Dupa cum am zis, sistemul va fi folosit în mare parte de angajații care vor să primească accesul intrării în sediu. Există 3 metode în care poate reaționa sistemul la

apropierea unei persoane de cameră. Diagramele de secvențe de mai jos arată fluxul prin care trece imaginea capturată și rezultatul dat de sistem, atunci când:

- Sistemul nu detectează fața persoanei în imaginea capturată și utilizatorul va primi răspuns că față lui nu este detectată și sistemul este în continuare în așteptare

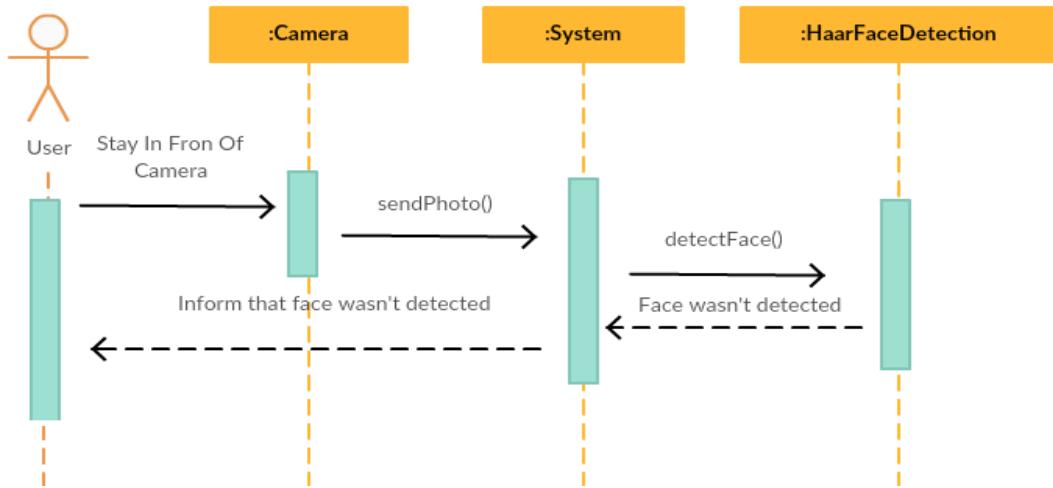


Figura 5.11: Diagrama de secvențe când persoana nu este detectată

- Sistemul nu recunoaște persoana în imaginea capturată și va primi răspuns că nu este recunoscută și se va conecta și ledul de culoare roșie

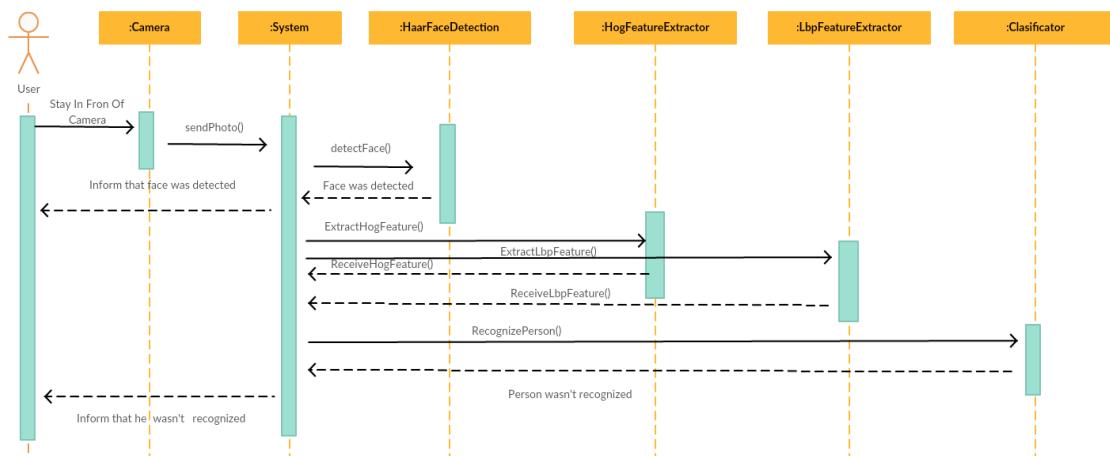


Figura 5.12: Diagrama de secvențe când persoana nu este recunoscută

- Sistemul recunoaște persoana și va trimite un mesaj că este recunoscută și se va porni ledul de culoare albastră.

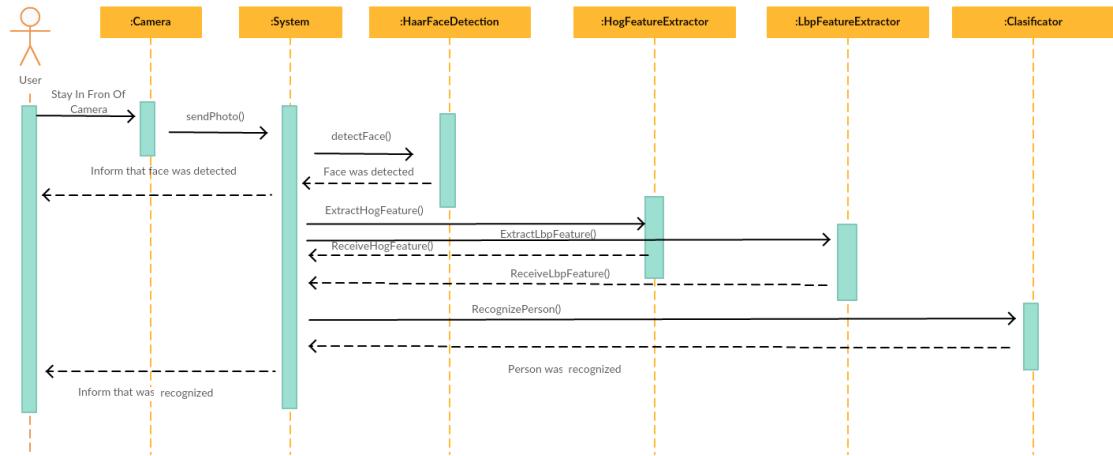


Figura 5.13: Diagrama de secvențe când persoana este recunoscută

# Capitolul 6

## Testare și Validare

### 6.1 Testarea versiunii initiale și celei refactorizate

Testările sistemului au fost făcuțe pe ambele dispozitive *Laptop ASUS, Intel(R) Core(TM) i7-4710HQ CPU @2.50GHz, 4,00 GB memorie RAM* și *Raspberry PI 3*. Prima fază pentru testarea algoritmilor a constat în obținerea a mai multor imagini de la color FERET [29] pentru testarea algoritmilor. Au fost extrase 100 de persoane, fiecare având mai multe imagini cu diferite poziții și emoții. Înainte de testare, imaginile celor 4 persoane au fost testate pentru detectarea feței și au fost sterse cele în care algoritmul Haar Cascades nu a găsit față. În Figura 6.1 se văd imaginile unei persoane care sunt de tip *.ppm* și pot fi ușor vizualizate cu *Irfan View*, un vizualizator de imagini.

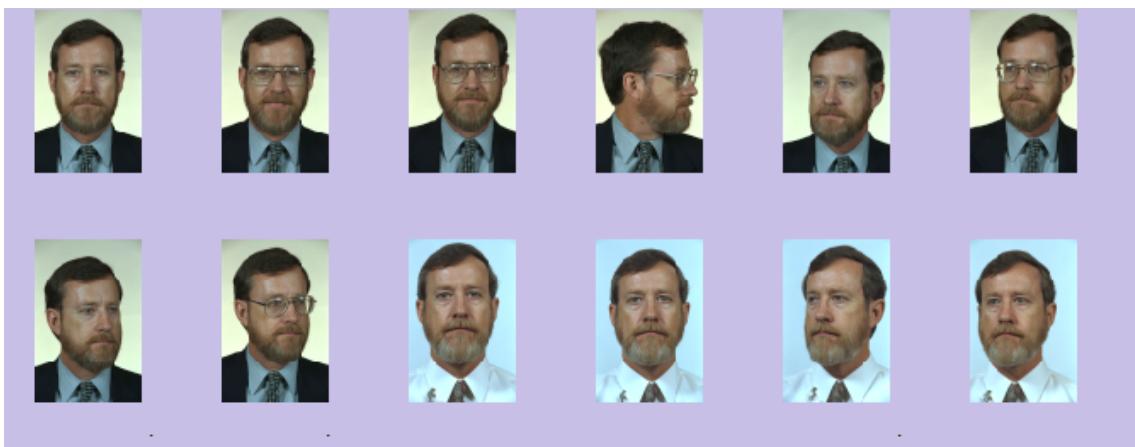


Figura 6.1: Poze pentru o persoană [29]

Au fost aplicăți algoritmii de extragere a caracteristicilor din 736 de imagini a celor 100 de persoane. Au fost stocate în două fișiere excel pentru algoritmii HOG și LBP separat. Pe fiecare rând este câte o histogramă și pe prima coloană fiind numele persoanei (Figura 6.2).

197 Employee (35)	0.4577083 0.3098285 0.4039376 0.3619002 0.2938739 0.2658547 0.2425413 0.2868887 0.3196870 0.4853758 0.3579302 0.3120293 0.2697034 0.2788692 0.3036136 0.2931153 0.3413085
198 Employee (35)	0.6012562 0.2895457 0.2906342 0.2721179 0.2835182 0.2556699 0.2420915 0.2564337 0.3549948 0.3777699 0.2051922 0.2111776 0.2320947 0.3202773 0.3695833 0.4103142 0.4337354
199 Employee (35)	0.3330464 0.3528536 0.5018896 0.4103527 0.3755946 0.3897620 0.1772685 0.1322498 0.1325979 0.4697564 0.2801462 0.3452153 0.2793591 0.3006013 0.3167807 0.3095673 0.3433784
200 Employee (35)	0.5596142 0.2736744 0.3248332 0.3020846 0.2936578 0.2635867 0.2364934 0.2419006 0.3807945 0.3629386 0.1743813 0.1894420 0.2446644 0.3225743 0.3844578 0.4178167 0.4504696
201 Employee (35)	0.5069964 0.3036500 0.3688450 0.2960929 0.2500645 0.2436199 0.2572928 0.3145750 0.3741570 0.4211473 0.2326554 0.2368742 0.2089500 0.2943990 0.3671089 0.3638494 0.4275088
202 Employee (35)	0.3681347 0.3241174 0.4379648 0.4341631 0.3974822 0.3156709 0.2339591 0.1951805 0.1692373 0.5670345 0.3952648 0.2592618 0.2474128 0.2765854 0.2857571 0.2775504 0.2921059
203 Employee (35)	0.3750734 0.3318732 0.4490806 0.3793871 0.3326898 0.3258159 0.2507408 0.2620855 0.2348907 0.5608591 0.3595111 0.3181375 0.2513774 0.2711491 0.2747180 0.2633902 0.2916742
204 Employee (36)	0.4522944 0.3349453 0.4491990 0.3870748 0.3109016 0.2529539 0.2291637 0.2248206 0.2606315 0.4162064 0.2156459 0.2452412 0.2483237 0.2782329 0.3223897 0.3992785 0.4370729
205 Employee (36)	0.4331840 0.3274103 0.4521616 0.3818879 0.2884554 0.2996145 0.2411086 0.2488441 0.2486623 0.4366101 0.2621678 0.2845070 0.2480984 0.3007146 0.3707562 0.3548829 0.3873492
206 Employee (36)	0.4211026 0.3342615 0.4577535 0.3861669 0.2979068 0.3127174 0.2393130 0.2405391 0.2249624 0.4206375 0.2109513 0.2253073 0.2383230 0.3174346 0.3933322 0.3612484 0.4141553
207 Employee (36)	0.4424226 0.3075057 0.3786938 0.3604831 0.3042500 0.2813878 0.2694333 0.2976150 0.3216037 0.4100821 0.2048536 0.2415224 0.2524329 0.2926126 0.3663938 0.4013028 0.4236278
208 Employee (36)	0.4011522 0.3210226 0.4035108 0.3977452 0.3610206 0.3221573 0.2775016 0.2330080 0.2226607 0.5027755 0.3238829 0.2803537 0.2926927 0.2809231 0.3043329 0.3271924 0.3415683
209 Employee (37)	0.4803317 0.4141626 0.4222492 0.3487308 0.2897214 0.2496066 0.2162477 0.2217296 0.2359535 0.5507417 0.2950790 0.2560358 0.2164732 0.2357560 0.2430887 0.2897199 0.3826059
210 Employee (37)	0.4218254 0.3580009 0.4110874 0.4209500 0.3494891 0.2904005 0.2264804 0.2067560 0.2172330 0.4788264 0.2910753 0.2803854 0.2204125 0.3237259 0.3478443 0.2843306 0.3538316
211 Employee (37)	0.4442835 0.3964787 0.3997888 0.3724688 0.3405817 0.2626050 0.2106536 0.2415426 0.2432641 0.4613743 0.2561960 0.2563082 0.2072268 0.2507164 0.3294926 0.3132789 0.4125755
212 Employee (37)	0.4907052 0.3737657 0.4247740 0.3540647 0.2974659 0.2423639 0.1998841 0.2154447 0.2830482 0.5272807 0.2533784 0.2334379 0.2162635 0.2309084 0.2503603 0.3035327 0.4091636
213 Employee (37)	0.4874324 0.3675212 0.3599746 0.3582293 0.3355875 0.2745283 0.2190572 0.2458846 0.2701806 0.4473961 0.2486420 0.2883886 0.2149233 0.2778143 0.3792344 0.3103135 0.3969288

Figura 6.2: Histogramele stocate în excel

Pentru fiecare test se vor folosi vectorii de caracteristici ale acestor 100 de persoane. După ce sistemul se pornește, primul lucru care se face e citirea tuturor vectorilor din ambele fișiere. Timpul de execuție a acestui proces pentru ambele dispozitive poate fi văzută în Tabelul de mai jos. E evident că viteza laptop-ului e mult mai mare și acest lucru va fi văzut și în următoarele exemple. De asemenea, din fișierul de LBP sunt extrase mai greu, deoarece histogramele sunt de lungimi mai mari 237 versus 37 ale celor de la HOG.

Tabelul 6.1: Timpul de extragere a caracteristicilor din excel (secunde)

Tipul Caracteristicilor	Laptop ASUS	Raspberry PI 3
Histogram of Oriented Gradients	0,36	2,80
Local Binary Pattern	2,73	16,73

Sistemul a trecut prin mai multe etape pentru a îmbunătăți acuratețea și viteza de recunoaștere. Vor fi testați și analizați tesările efectuate pe aceste două versiuni:

1. Versiunea 1 - reprezintă prima fază a sistemului. Erau implementați ambii algoritmi fără refactorizarea codului. În algoritmul HOG se folosea înmulțire cu număr float și un set de if-uri pentru detectarea bin-ului în care trebuie adăugată magnitudinea. În algoritmul LBP se calculau anumite valori de fiecare dată, care rămâneau constante pentru toate imaginile.

2. Versiunea 2 - este a două etapă după ce a avut loc refactorizarea. A fost eliminată înmulțirea cu float din implementarea HOG și valorile care setează tipul metodei de LBP au fost setate ca constante.

Prima testare se va face cu imaginea prezentată în capitolul anterior, care are rezoluția 1679x1679. Timpul necesar pe ambele dispozitive pentru execuția detectiei feței, algoritmilor de recunoaștere și timpul total necesar de a oferi răspunsul sunt prezentate în Tabelul 6.2. Se poate observa că timpii de execuție sunt mult mai mici la rularea sistemului pe Laptop decât Raspberry PI 3.

Tabelul 6.2: Timpul de recunoaștere (secunde)

Tipul Caracteristicilor	Laptop ASUS	Raspberry PI 3
Detectia Feței	0,01	0,06
Histogram of Oriented Gradients	0,68	8,80
Local Binary Pattern	0,89	8,66
Clasificator	1,58	17,57

Pentru a vedea îmbunătățirea sistemului, în Tabelul 6.3 sun prezenți timpi de execuție a metodelor discutate mai sus.

Versiunea 2(imagine mica)

Tabelul 6.3: Timpul de recunoaștere (secunde)

Tipul Caracteristicilor	Laptop ASUS	Raspberry PI 3
Detectia Feței	0,01	0,06
Histogram of Oriented Gradients	0,14	1,13
Local Binary Pattern	0,62	4,27
Clasificator	0,78	5,41

Primul pas al sistemului este detectia feței, care se va face cu aceeași viteză pentru ambele versiuni, deoarece acest algoritm nu a avut o schimbare. Rezoluția obținută după decuparea feței este 186x186 care se va face de 6 ori mai repede folosind laptop-ul. Se observă că refactorizarea a îmbunătățit seminificativ timpii de execuție a metodelor. Acest lucru poate fi văzut în speed-upul (6.1) pentru ambele dispozitive pe baza metodei de clasificator, care execută ambii algoritmi de recunoaștere și oferă răspunsul sistemului.

$$Speedup_{LAPTOP} = \frac{1,58}{0,76} = 2,07 \quad Speedup_{RASPBERRY} = \frac{17,57}{5,41} = 3,24 \quad (6.1)$$

De asemenea, timpul depinde foarte mult de numărul de pixeli prezenți în imagine. Pentru a vedea diferențele se va face test pe o imagine cu rezoluția 1679x1679. Detectia

feței se va face cu câteva milisecunde mai greu și imaginea obținută după decupare va avea dimensiunea de 862x862. În schim se vede foarte evident că execuția algoritmilor va avea nevoie de mai mult timp. În Tabelul 6.4 sunt timpii de execuții folosind prima versiune și în Tabelul 6.5 cei folosind versiunea refactorizată.

Tabelul 6.4: Versiunea 1 - Timpul de recunoaștere (secunde)

Tipul Caracteristicilor	Laptop ASUS	Raspberry PI 3
Detectia Feței	0,29	1,20
Histogram of Oriented Gradients	11,70	175,61
Local Binary Pattern	7,66	116,25
Clasificator	19,37	290,86

Tabelul 6.5: Versiunea 2 - Timpul de recunoaștere (secunde)

Tipul Caracteristicilor	Laptop ASUS	Raspberry PI 3
Detectia Feței	0,29	1,20
Histogram of Oriented Gradients	1,38	10,06
Local Binary Pattern	2,31	14,67
Clasificator	3,11	24,73

Din aceste testări, este evident că ar fi bine ca imaginea să fie cât mai mică, astfel încât utilizatorul să nu trebuiască să aștepte mult răspunsul. Iar această imagine mai mare a fost folosită pentru a scoate în evidență îmbunătățirea vitezei după refactorizare (6.2).

$$Speedup_{LAPTOP} = \frac{19,37}{3,11} = 6,22 \quad Speedup_{RASPBERRY} = \frac{290,86}{24,73} = 11,76 \quad (6.2)$$

Pe lângă viteza de execuție a algoritmilor de recunoaștere facială care joacă rolul important pentru a oferi comoditate utilizatorilor, este foarte importantă și acuratețea de recunoaștere. Ea nu este afectată de dispozitivul folosit, doar puțin diferă la Versiunea 2, din cauza eliminării valorii de float și este folosit un round pentru această valoare. Ambele verisuni a sistemului au fost testate pe cele 100 de persoane și au fost folosite câte o poză de test pentru fiecare persoană, care nu fac parte din imaginile folosite pentru stocarea vectorilor de caracteristici. Rezultatele pot fi văzute în Tabelul 6.6. Se observă că după refactorizarea care a crescut semnificativ viteza a scăzut puțin acuratețea. De asemenea, se vedea foarte bine rolul fuziunii algoritmilor, care este mai mare decât acuratețea fiecărui algoritm individual.

Tabelul 6.6: Acuratețea de recunoaștere (%)

Algoritmul	Versiunea 1	Versiunea 2
Histogram of Oriented Gradients	75	67
Local Binary Pattern	78	73
Fuziunea Algoritmilor	84	77

## 6.2 Testarea versiunii paralelizate

Este folosită programarea paralelă, fiind adăugate 4 procese pentru extragerea caracteristicilor din cele 4 regiuni ale imaginii, pentru a îmbunătăți viteza sistemului de recunoaștere facială folosind Raspberry PI. Aceasta este dispozitivul pe care sistemul va fi implementat pentru recunoașterea angajaților. În Tabelele 6.7 și 6.8 sunt prezentate timpii de execuție a algoritmilor serial vs paralel pentru ambele imagini.

Tabelul 6.7: Timpul de recunoaștere (secunde)

Metoda	Serial	Paralel
Histogram of Oriented Gradients	1,13	0,86
Local Binary Pattern	4,27	3,86
Clasificator	5,41	4,75

Tabelul 6.8: Timpul de recunoaștere (secunde)

Metoda	Serial	Paralel
Histogram of Oriented Gradients	10,06	3,46
Local Binary Pattern	14,67	7,23
Clasificator	24,73	11,42

$$Speedup_1 = \frac{5,41}{4,75} = 1,13 \quad Speedup_2 = \frac{24,73}{11,42} = 2,16 \quad (6.3)$$

Se vede clar că viteza de oferire a răspunsului crește și acest lucru va oferi o comoditate mare utilizatorilor.

## 6.3 Testarea sistemului final

După cum s-a specificat și în capitolul precedent, sistemul va fi executat pe Raspberry PI 3, la care sunt conectate o Cameră Module V2 care va captura imagini la fiecare

secundă și va efectua detecția feței. De asemenea, este conectat și un Arduino cu un Afisor LCD și un board cu 3 leduri pentru afișarea răspunsului la utilizator. În Figura 6.2 se poate vedea cum arată sistemul.

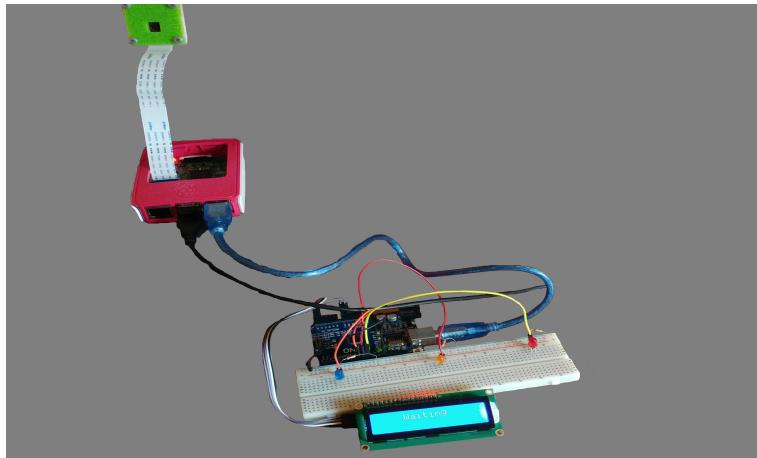


Figura 6.3: Sistemul de recunoaștere facială

Doar dacă detectează ceva, va captura regiunea care conține față și va executa recunoaștere, altfel va continua să facă imagini. Timpul mediu necesar de recunoaștere a persoanelor este 5 - 10 secunde, depinde de dimensiunea imaginii. După cum ziceam, sunt 4 faze în care se poate afla sistemul:

1. Sistemul este în așteptare - înseamnă că continuă să facă poze și execută detecția feței, dar rezultatul este negativ. Este afișat mesajul *Waiting* pe Afisor și nici un led nu este conectat (Figura 6.3).

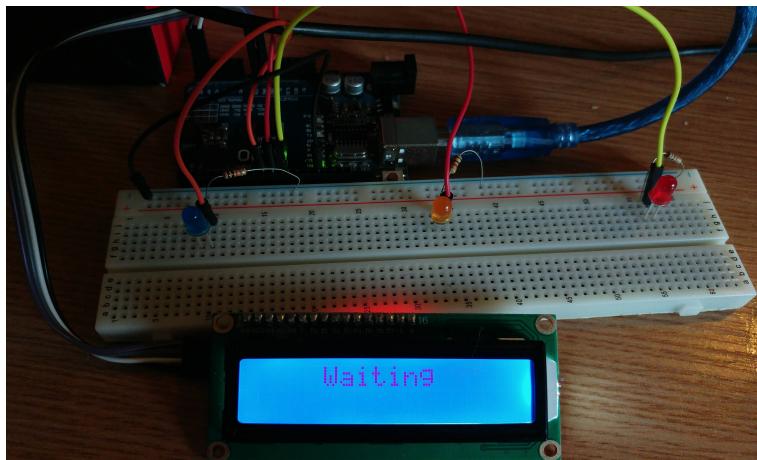


Figura 6.4: Sistemul este în așteptare

2. Sistemul a detectat față - reprezintă faptul că o persoană este în fața camerei și a fost detectată. S-a executat decuparea feței și atâtă timp cât mesajul *Face Detected* este afișat se face recunoașterea persoanei. De asemenea, în acest moment este conectat ledul de culoare galbenă (Figura 6.4).

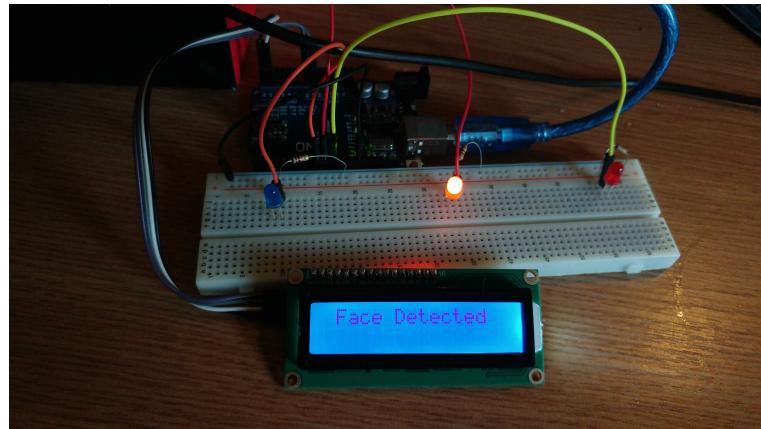


Figura 6.5: Sistemul a detectat față

3. Persoana nu a fost recunoscută - înseamnă ca răspunsul de recunoaștere este negativ și va fi afișat mesajul *Not Recognized* și led-ul de culoare roșie se va conecta (Figura 6.5).

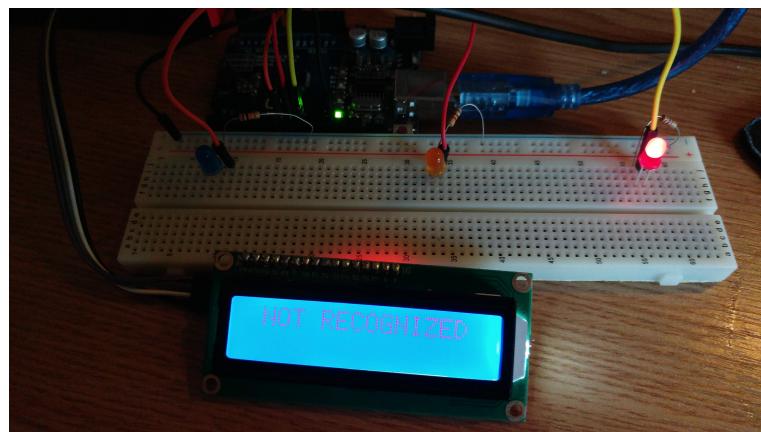


Figura 6.6: Sistemul nu a recunoscut persoana

4. Persoana este recunoscută - se va afișa mesajul de *Hello Numele\_Persoanei!*. Acest test este făcut pe imaginea mea proprie și mesajul este *Hello Dumitru!*. De asemenea, se va conecta ledul de culoare albastră (Figura 6.6).

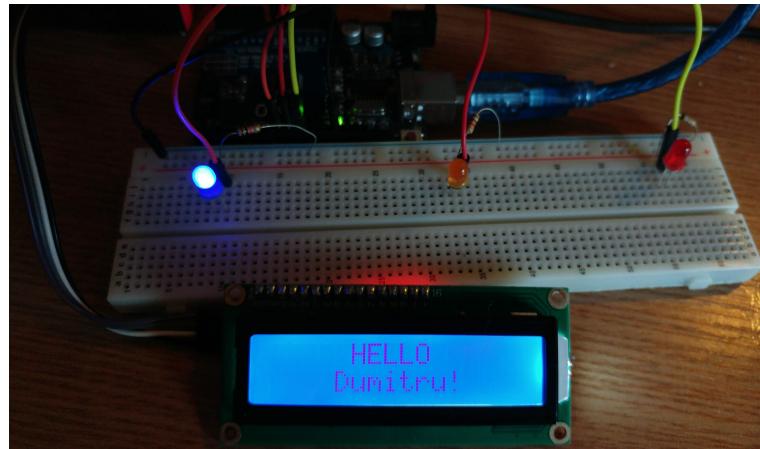


Figura 6.7: Sistemul a recunoscut persoana

## Capitolul 7

# Manual de Instalare și Utilizare

Acest sistem va fi executat pe Raspberry PI 3, la care va fi conectată o Camera Module V2 și de asemenea un Arduino cu un Afisor LCD 16x2 și un board cu 3 leduri. Pentru a putea utiliza acest sistem, este necesară achiziția acestor lucruri. De asemenea poate fi folosită altă versiune de Raspberry sau chiar alt sistem, altă camera și de asemenea altă modalitate de afișare a rezultatelor, doar că asta va avea nevoie de unele modificări ale sistemului. Sistemul fiind flexibil și adaptabil, acest lucru se va face ușor. Dacă scopul este doar de a folosi acest sistem și dispuneți de dispozitivile enumerate mai sus, trebuie execută următorii pași:

1. Instalarea Python 3
  - Se deschide Terminal Console
  - Se actualizează listă de pachete a sistemului executând următoarea comandă:*sudo apt-get update*
  - Se actualizează pachetele instalate la cele mai recente versiuni cu comanda:*sudo apt-get upgrade*
  - Python 3 va fi instalat cu comanda *sudo apt-get install python3*
2. Instalare OpenCV - constă din mai mulți pași care vor fi găsiți la link-ul adăugat în bibliografie [30]. Instalarea acestei librării va lua mai mult de 9 ore.
3. Se conectează Camera Module V2 la Raspberry Pi 3
4. Se instalează Arduino
  - Se deschide Terminal Console
  - Se actualizează listă de pachete a sistemului executând următoarea comandă:*sudo apt-get update*
  - Se actualizează pachetele instalate la cele mai recente versiuni cu comanda:*sudo apt-get upgrade*

- Arduino va fi instalat cu comanda `sudo apt-get install arduino`
5. Se conectează Arduino la Raspberry Pi 3
  6. Se verifică portul la care a fost atașat Arduino prin executarea comenții `ls /dev/tty*` din Terminal Console. Este afișată o listă de conexiuni și se uita la `/dev/tty/USB` și se va edita în fișierul `main.py` cu cifra după USB
  7. Se conectează afișorul LCD la Arduino
  8. Se conectează la board 3 leduri și este unit la Arduino. Trebuie să fie folosie Pin-urile corespunzătoare culorilor: 11(albastru),12(galben),13(roșu)
  9. Se deschide Terminal Console în mapa care conține toate fișierile aplicației
  10. Se pornește fișierul `arduino.ino`
  11. Se execută comanda `python main.py` pentru a porni sistemul
  12. Se poziționează capul astfel încât să fie prins de camera și se văd rezultatele afișate pe LCD și leduri

# Capitolul 8

## Concluzii

Existând multe probleme care pot fi provocate de acțiunile oamenilor, sistemul de recunoaștere facială este o soluție. Poate spori securitatea informațiilor secrete, înălătura frauda, furtul și de asemenea poate încuraja oamenii la dezvoltarea lor personală. Acest sistem având rolul de a recunoaște și a oferi acces angajaților în sediu și de a interzice acest lucru persoanelor străine, va asigura o securitate majoră informației interne și de asemenea va oferi o comoditate utilizatorilor. Pe lângă aceasta, cu acest sistem poate fi stocat timpul petrecut de angajat în birou, ceea ce nu ar permite ca persoana să mintă firma și de asemenea îi va oferi o dezvoltare profesională mai mare. Conform testărilor executate în capitolul precedent, utilizatorul va trebui să aștepte 4-7 secunde care este un timp bun, dar sigur acest sistem are nevoie de o dezvoltarea a acestui indiciu. De asemenea, acuratețea de recunoaștere este de 80% fiind un lucru bun, dar nu atinge obiectivul stabilit la începutul acestui proiect. După punerea în funcțiune a acestui sistem, se vor cunoaște mai bine punctele tari și cele slabe. Înainte de aceasta, doar conform testărilor executate și scopurilor inițial stabilite, există posibile dezvoltări și îmbunătățiri ulterioare:

### 1. Creșterea acurateței de recunoaștere

- Dezvoltarea algoritmilor utilizați
- Testarea altor algoritmi de recunoaștere și fuziunii lor
- Adăugarea mai multor imagini cu mai multe tipuri de poziție a feței, emoții, stil, vestimentație
- Folosirea unei camere mai performante
- Împărțirea imaginii în mai multe regiuni
- Efectuarea și detectia ochilor, care să fie recunoscuți separat

### 2. Creșterea vitezei de recunoaștere

- Folosirea unui calculator mai puternic

- Efectuarea unei paralelizări mai mari, dispozitivul având mai multe nuclee
- Redimensionarea imaginilor, astfel încat algoritmii de recunoaștere să se execute mai repede
- Folosirea altori algoritmi care pot avea viteză mai mare
- Dezvoltarea modalității de comparare a vectorilor de caracteristici, astfel încât să nu fie parcursi toți

### 3. Creșterea comodității de utilizare

- Folosirea unui dispozitiv mai clar pentru afișarea răspunsului
- Oferirea rapidă a răspunsului
- Să poată fi detectat și din mișcare

### 4. Toate obiectivele stabilite inițial să fie atinse

- Dacă persoana este recunoscută, să se deschidă ușa
- Persoana recunoscută să primească un mesaj pe telefon că a obținut accesul. Asta ar crește securitatea, fiindcă dacă un străin este recunoscut, un angajat va primi acest mesaj.
- Dacă intra mai multe persoane, să se facă recunoașterea la toate

Proiectul are stabilit fundamentul, e bine organizat și adaptabil, încât să suporte schimbări și dezvoltări ulterioare descrise mai sus. Chiar dacă sistemul are mici probleme la acuratețea recunoașterei și viteza de execuție nu oferă o comoditate majoră utilizatorilor, pentru mine a fost un început foarte bun pe acest domeniu de procesare a imaginilor. Fiind cel mai mare proiect pe care am lucrat individual, am învațat foarte multe lucruri necesare:

1. Procesare de imagini
2. Python și Raspberry PI 3, cu care am lucrat prima data
3. Rolul și cât de multe organizații folosesc recunoașterea facială
4. Gestionarea timpului
5. Rolul programării paralele
6. Conexiunea a mai multor dispozitive
7. Rolul refactorizării și analizei codului

Scopul pe care aş vrea să îl ating în viitor este o aprofundare în acest domeniu, astfel încât să fiu apt să dezvolt acest sistem la un nivel cât mai înalt ( *Acuratețe = 99,9%* și *Timpul de recunoaștere < 1 secundă* ) . De asemenea, sper să înceapă să fie utilizat de o companie și chiar să aducă un benefiu imens conducerilor și angajaților firmei.

# Bibliografie

- [1] S. Fussell, “Facebook.’ [Online]. Available: <https://gizmodo.com/facebook-new-face-recognition-features-what-we-do-an-1823359911/>
- [2] D. Smith, “Passport officers using face recognition technology better at detecting fraud.’ [Online]. Available: <https://gizmodo.com/facebook-new-face-recognition-features-what-we-do-an-1823359911/>
- [3] C. Middlehurst, “China first facial recognition atm.’ [Online]. Available: <https://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html/>
- [4] O. Solon, “Facial recognition database used by fbi.’ [Online]. Available: <https://www.theguardian.com/technology/2017/mar/27/us-facial-recognition-database-fbi-drivers-licenses-passports/>
- [5] J. Y. Jessica Li, “The potentials of facial recognition.’ [Online]. Available: <https://lhsepic.com/1432/in-depth/the-potentials-of-facial-recognition/>
- [6] A. P. Matthew A.Turk, “Face recognition using eigenfaces,’ 1991.
- [7] M. Brems, “A one-stop shop for principal component analysis,’ 2017.
- [8] M. Tran, “Example for eigenface eigenfaces look somewhat like generic faces,’ 2008.
- [9] J.-N. C. Hyung-Ji Lee, Wan-Su Lee, “Face recognition using fisherface algorithm and elastic graph matching.’
- [10] L. D. Analysis, “Sebastian raschka,’ 2014.
- [11] A. E. W. Vijayshree A. More, “Improved fisher face approach for human recognition system using facial biometrics,’ 2012.
- [12] F. C. SHU Chang, DING Xiaoqing, “Histogram of the oriented gradient for face recognition,’ 2011.
- [13] T. W. M. S. A. Md. Abdur Rahim, Md. Najmul Hossain, “Face recognition using local binary patterns (lbp),’ 2013.

- [14] A. Rosebrock, "Local binary patterns with python and opencv,' 2015.
- [15] F. Doukkali, "One-shot learning: Face recognition using siamese neural network,' 2017.
- [16] S. Kim, "Understanding facial recognition through openface,' 2018.
- [17] "Face detection using haar cascades.' [Online]. Available: [https://docs.opencv.org/3.4.1/d7/d8b/tutorial\\_py\\_face\\_detection.html/](https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html/)
- [18] C. Gaowei, "The face detection system based on gpu+cpu desktop cluster.'
- [19] R. K. Janarbek Matai, Ali Irturk, "Design and implementation of an fpga-based real-time face recognition system.'
- [20] C. K. S. D. Ishita Gupta, Varsha Patil, "Face detection and recognition using raspberry pi,' 2016.
- [21] S. Mallick, "Histogram of oriented gradients.' [Online]. Available: <https://www.learnopencv.com/histogram-of-oriented-gradients/>
- [22] B. Julsing, "Face recognition with local binary patterns,' 2007.
- [23] N. Delbiaggio, "A comparison of facial recognitions algorithms,' 2017.
- [24] A. Ozdil, "A survey on comparison of face recognition algorithms.'
- [25] M. M. D. Mohsen Ghorbani, Alireza Tavakoli Targhi, "Hog and lbp: Towards a robust face recognition system,' 2015.
- [26] T. B. E. L.-M. G.B. Huang, M. Ramesh, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in, technical report 07-49,' 2007.
- [27] A. Martinez, "The ar face database,' 1997.
- [28] S. Z. Zhihua Xie, Peng Jiang, "Fusion of lbp and hog using multiple kernel learning for infrared face recognition,' 2013.
- [29] P. J. Phillips, "color feret database.' [Online]. Available: <https://www.nist.gov/itl/iad/image-group/color-feret-database/>
- [30] A. Rosebrock, "How to install opencv 3 on raspbian jessie.' [Online]. Available: <https://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/>