



**UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI
INFORMATICĂ
PROGRAMUL DE STUDII DE LICENȚĂ:
MATEMATICĂ - INFORMATICĂ**

LUCRARE DE LICENȚĂ

COORDONATOR:

Lector Dr. Octavian-Liviu Mafteiu-Scai

ABSOLVENT:

Zoltan Mark

TIMIȘOARA

2018

UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI
INFORMATICĂ
PROGRAMUL DE STUDII DE LICENȚĂ:
MATEMATICĂ - INFORMATICĂ

Sisteme de Securitate bazate pe recunoașterea Irisului

COORDONATOR:

Lector Dr. Liviu-Octavian
Mafteiu-Scai

ABSOLVENT:

Zoltan Mark

TIMIȘOARA
2018

Rezumat

Performanța biometriei a crescut din ce în ce mai mult pe parcursul timpului. Tehnologia biometrică de recunoaștere a Irisului, a devenit foarte utilizată pentru mai multe aplicații de securitate la o scară foarte mare. Avantajele principale ale recunoașterii irisului, spectrului vizibil, includ posibilitatea de imagistică a irisului în scenarii de mișcare la apropiere sau distanță. În această lucrare propunem o nouă utilizare a acestei tehnologii biometrice de recunoaștere a irisului. Lucrarea constă într-un sistem de securitate care funcționează pe tehnologia biometrică menționată mai sus. Această lucrare este compusă din două componente principale: partea de detectare a ochilor care este realizată în limbajul de programare C++, iar a doua parte este realizarea a unei rețele neuronale în limbajul de programare Python.

Abstract

The performance of biometrics has increased steadily over time. Biometric Iris recognition technology has become widely used for many security applications at a very large scale. The main advantages of recognizing the iris, the visible spectrum, include the possibility of imaging the iris in proximity or distance scenarios. In this paper we propose a new use of this biometric iris recognition technology. The paper consists of a short system that works on the biometric technology mentioned above. This paper is composed of two main components: the eye detection part that is made in the C ++ programming language, and the second part is the realization of a neural network in the Python programming language.

Cuprins

1	Introducere	6
2	Stare de artă	8
2.1	Securitate	10
2.2	Sisteme existente	11
2.3	Avantaje	13
2.4	Dezavantaje	13
3	Rețele Neuronale	15
3.1	Backpropagation	17
3.2	Feedforward propagation	18
3.2.1	Multi-layer	18
3.3	Gradient descent	18
4	Aplicație	20
4.1	Tehnologii utilizate	20
4.1.1	OpenCV	20
4.1.2	Haar-Cascades	21
4.1.3	NumPy	21
4.1.4	Editoare utilizate	22
4.2	Implementare	22
4.2.1	Detectarea ochilor	25
4.2.2	Retea neuronală	29
4.3	Rezultate experimentale	34
5	Concluzii	35
5.1	Evaluarea Aplicației	35
5.2	Directii viitoare de dezvoltare	35
	Bibliografie	36

Capitolul 1

Introducere

În ultimii ani, putem spune că securitatea personală a devenit din ce în ce mai importantă în viața de zi cu zi. Iar tehnologiile biometrice au influențat foarte mult acest domeniu al securității. Se poate afirma că noile sisteme au dus la definirea unei alte clase de securitate, cu mult diferită de sistemele tradiționale bazate pe coduri de bare, chei fizice, sau cartele care pot fi pierdute sau multiplicare. Parolele pot fi sparte, uitate, observate de către alte persoane, dar trăsăturile anatomice ale corpului uman nu pot fi pierdute și nici copiate cu ușurință. Pentru a avea un acces protejat la diverse informații, o persoană se identifică prin diferite metode: cheie sau cartelă, Pin-uri sau parole, sau prin trăsături anatomice. Tehnica de acces prin analiza trăsăturilor anatomice se numește biometrie. Funcțiile biometrice se referă la metode automatizate de recunoaștere a unei persoane pe baza fizionomiei acesteia. Spre exemplu: amprente digitale, recunoaștere facială, recunoașterea irisului, retinei, sau a venelor. Metoda de autentificare biometrică bazată pe recunoașterea irisului este una dintre cele mai înalte forme de sisteme de securitate. Irisul se situează între pupil și sclera ochiului, iar de la vârsta de un an, irisul se fixează și rămâne constant pe parcursul vieții. Sistemele de recunoaștere a irisului au o pondere de precizie și de siguranță mult mai mare decât alte tipuri de sisteme biometrice, precum ar fi amprenta sau amprenta vocală. Aplicațiile de securitate bazate pe recunoașterea Irisului se dezvoltă pe camere video sau camere cu infraroșu, pentru autentificarea pe diverse platforme sau autentificări ale tranzacțiilor. Tehnicile de recunoaștere a irisului sunt utilizate pe platforme mobile, clădiri, rețele de calculatoare sau calculatoare personale. Tehnologia de recunoaștere a irisului poate fi folosită și de persoanele cu lipsă de vedere, deoarece această tehnologie este dependentă de modelul irisului și nu de vederea propriu zisă. Aplicația descrisă în această lucrare se numește I.R.U.N.N. (Iris Recognition using

Neural Networks). Această aplicație detectează ochiul uman urmând ca după alegerea unuia dintre ochi, imaginea va fi transmisă către rețeaua neuronală pentru ca imaginea să poată să fie verificată. Astfel putem să antrenăm rețeaua neuronală pentru a recunoaște imaginile cu un anumit tip de iris, iar rezultatele vor putea fi folosite în diverse sisteme de parole sau de identificare a persoanelor.

Capitolul 2

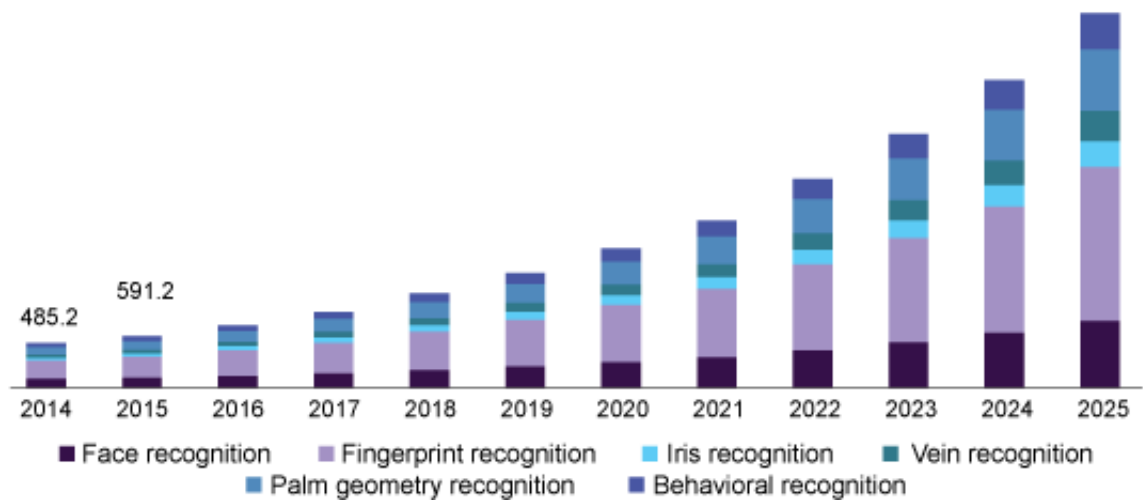
Stare de artă

Utilizarea metodelor de recunoaștere a Irisului pentru identificarea persoanelor ,a apărut prima dată în anul 1953 și a fost propusă de oftalmologul Frank Burch. Mai târziu John Daugman împreună cu Leonard Flom și Aron Sa'ar, au dezvoltat prima tehnologie de recunoaștere a Irisului la Universitatea Harvard. John Daugman , Leonard Flom și Aron Sa'ar, au fondat Iridian Technologys care a fost una dintre primele corporații care se ocupau de tehnologiile biometrice în acele timpuri. În 2008 aceste tehnologii au continuat să avanseze cu activități de cercetare la mai multe companii. Acumularea treptată a cunoștințelor despre modul cel mai adecvat de aplicare a acestei tehnologii de potrivire a Irisului, au dus la o gamă largă de aplicații militare importante. Biroul Federal American de Investigații a anunțat că va adăuga tehnologia de recunoaștere a Irisului în diferite programe și aplicații. Utilizatorii de sisteme de recunoaștere a irisului nu trebuie să își afirme identitatea. În schimb, doar prin prezentarea unuia dintre ochi sistemului, identitatea lor este determinată automat, cu o singură condiție, ca aceasta să fi fost deja înscrisă într-o bază de date. Identificarea se realizează integral, la o viteză tipică de 100.000 de persoane pe secundă pe un singur procesor de 300Mhz. [1] Sistemele optice pentru obținerea imaginii irisului au arătat cele mai impresionante progrese, permițând în general o interfață de utilizator mult mai flexibilă și o distanță mai bună între subiect și cameră. Activitatea efectuată de către Dr. J. Matey și echipa sa de la laboratoarele Sarnoff a condus la o nouă generație de sisteme care capturează "iris-at-a-distance" și "iris-on-the-move". A existat o cursă pe diferite distanțe pentru a demonstra cea mai lungă distanță de stand-off, cu unele cereri care se exindeau la zeci de metri. Implementările majore ale algoritmilor lui J. Daugman pentru sistemele de recunoaștere a irisului au fost la aeroporturi. În locul prezentării pașapoartelor și pentru securitate au utilizat liste de vizionare. După anul 2000, misiuni

importante au început să apară la aeroportul Schiphol din Amsterdam și la înca zece terminale de aeroport din Marea Britanie care au permis călătorilor frecvenți să-și prezinte irisul în locul pașaportului lor, într-un program numit IRIS.[2]

În prezent există câteva dispozitive de captură multi-biometrică pe piață. Dispozitivele existente pe piață sunt într-o variație de factori de forma: modele de captură a unuia sau al ambilor ochi. Există dispozitive care sunt simple, ușor de utilizat și sunt destinate exclusiv captării Irisului. Dispozitivele de ultimă oră captează automat irisul de la o distanță de zece metri sau chiar mai mare, în timp ce subiectul se află în mișcare. Pe parcursul anilor, tehnologiile biometrice au fost implementate și pentru telefoanele mobile. În 2013 compania Apple a introdus pentru prima dată pe Iphone 5S, tehnologia biometrică de fingerprint-scanner, iar aproximativ un an mai târziu Samsung, HTC și Huawei au adoptat aceeași tehnologie pentru device-urile din brand-ul personal. Compania Samsung a lansat după pe piață, telefonul mobil Samsung Galaxy Note 7 care a fost primul telefon care utiliza tehnologia de recunoaștere a Irisului. Acest device putea fi blocat sau deblocat de la o distanță de treisprezece, paisprezece centimetri distanță de ochii utilizatorului, iar telefonul nu trebuia să fie ținut exact în fața ochilor posesorului. Acesta putea să sincronizeze ușor cu ochii cu senzorul și putea să fie deblocat în una sau două secunde. Odată ce modelul de Iris a fost salvat în device, telefonul stoca informațiile ca o bucată de cod criptat, ceea ce face aceasta tehnologie destul de sigură. Când utilizatorul accesează telefonul, LED-ul și camera foto, ambele incluse în telefon, funcționează împreună pentru a realiza captura Irisului, iar apoi extrage codul criptat și compară cele două imagini înainte de a permite accesul utilizatorului. Compania Samsung a implementat tehnologia de scanare a Irisului și pentru a efectua tranzacții cu ajutorul aplicației Samsung Pay sau de a verifica contrurile bancare prin intermediul aplicației Samsung Pass. Imagistica vizibilă în apropierea razelor infraroșii Toate sistemele existente de recunoaștere a irisului, dobândesc imagini de iris în timp ce sunt iluminate de banda de lungime de undă apropiată de infraroșu a spectrului electromagnetic. O mare parte din persoanele de pe glob au ochi caprui întunecați, acest fenomen dominant al populației umane, care dezvăluie textura mai puțin vizibilă în bandă VW, dar care este mult mai bogată în structură, ca suprafața crapată a lunii, în banda NIR. Utilizarea spectrului NIR permite blocarea reflexelor speculare ale corneei dintr-un mediu ambiant luminos, care permite să se întoarcă în camera de iris doar acele lungimi de undă din iluminatorul cu bandă îngustă. În melanină, care este cunoscută și sub denumirea de cromofor, constă

în principal din două macromolecule heterogene distincte, denumite eumelanin și fenomelanin, a căror absorbție la lungimi de undă mai mari în spectrul NIR este neglijabilă. Cu toate acestea, la lungimi de undă mai scurte din spectrul VW, acești cromofori sunt excitați și pot genera modele bogate. Mahdi S. Hosseini oferă o comparație între aceste două modalități imagistice. A fost introdusă și o metodă de extragere a caracteristicilor alternative pentru codarea imaginilor de iris VW, care poate oferi o abordare alternativă pentru sistemele biometrice multimodale.



Grafic.
Figura 1[3]

În imaginea de mai sus (Figura 1) se poate observa creștea în popularitate a sistemelor biometrice, iar sistemele de recunoaștere a irisului au început să crească din ce în ce mai mult.

2.1 Securitate

În majoritatea cazurilor tehnologiile biometrice de identificare, au un aspect important și anume verificarea țesuturilor vii. Siguranța oricărei identificări biometrice depinde de faptul că semnalul dobândit și comparat să fie efectiv înregistrat dintr-o parte a corpului al unei persoane care urmează să fie identificată. Majoritatea tehnologiilor de recunoaștere a irisului disponibile pe piață sunt ușor de păcălit prin prezentarea unei fotografii de înaltă calitate a feței în locul unei fețe reale, ceea ce face astfel de dispozitive nepotrivite pentru aplicații nesupravegheate, cum ar fi sisteme de control al accesului la ușă. Dar acest lucru nu este

valabil în cazul tuturor algoritmilor și sistemelor de recunoaștere a irisului. Problema verificării țesuturilor vii este mai puțin îngrijorătoare în aplicații supravegheate, în care o persoană supraveghează acest proces.

Metode Diferite metode care au fost sugerate pentru a oferi o protecție cât mai bună împotriva folosirii ochilor și a iriselor false includ schimbarea iluminatului ambiental în timpul identificării, deoarece să se poată verifica reflexul pupilar și să se poată înregistra imaginea irisului la mai multe diametre diferite. Astfel analizând spectrul de frecvență spațială 2D a imaginii irisului pentru vârfurile cauzate de modelele de dithering ale imprimantelor gasite pe lentile de contact false, analizând astfel spectrul temporal de frecvență al imaginii pentru vârfurile cauzate de afișarea pe computer. Alte metode includ utilizarea analizei spectrale în loc de camerele monocromatice numai pentru a distinge țesutul de iris de alte materiale. Observarea mișcării naturale caracteristice ochiului uman, a globului ocular sau pentru reflexii din cele patru suprafețe optice ale ochiului, pentru a efectua verificarea prezenței, poziției și forma acestuia. Un raport din 2004 al Biroului Federal German pentru Securitatea Informației, a raportat că nici unul dintre tehnologiile de recunoaștere a irisului disponibile pe piață la aceea data, nu au implementat un algoritm de verificare a țesuturilor vii. Ca orice tehnologie de recunoaștere a modelelor, verificatorii de țesuturi vii vor avea propriile lor probabilități false de respingere a irisului, iar prin urmare vor reduce în continuare probabilitatea globală de a accepta un utilizator legitim de către senzor.

2.2 Sisteme existente

- În Emiratele Arabe Unite, Controlul de frontieră au utilizat IrisGuard la frontiere, ca și sistem de securitate pentru a putea verifica expulzarile în Emiratele Arabe Unite. În ziua de astăzi, toate porturile terestre, aeriene și maritime ale Emiratelor Arabe Unite de intrare sunt echipate cu acest sistem. Toți cetățenii străini care au nevoie de viză de vizită pentru a intra în Emiratele Arabe Unite sunt acum procesați prin sistemele de recunoaștere a irisului instalate la toate punctele primare și auxiliare de inspecție a imigrației.
- Din 2009 IrisGuard a implementat prima mașină automată de deblocare operațională (ATM) la Cairo Amman Bank, unde clienții băncii pot retrage fără numerar banii de pe ATM-uri fără carte

bancară sau PIN, ci doar prin prezentarea ochiului irisului camera de recunoaștere pe ATM.

- Forțele de poliție din America au să înceapă să utilizeze, în 2012, tehnologia mobilă MORIS care este un sistem de recunoaștere a infracțiunilor mobile și a sistemului de informații mobil al BI2 Technologies. Departamentul de Poliție din New York a fost primul, cu un sistem instalat în Manhattan în toamna lui 2010.
- SRI International Sarnoff dezvoltă un sistem și set de produse "Iris on the Move", în primul rând pentru clienții guvernului S.U.A., capabili să identifice 30 de persoane pe minut. Cel mai recent, ei s-au specializat într-un produs în care șoferii pot fi identificați fără a fi nevoie să-și părăsească vehiculul.[4]
- La sfârșitul anului 2015, Microsoft a lansat două telefoane Lumia (Lumia 950 și Lumia 950 XL) cu scanare de iris ca modalitate de autentificare a utilizatorului.

2.3 Avantaje

Irisul a fost ales ca fiind parte ideală a corpului uman pentru identificarea biometrică din mai mult puncte de vedere:

- Ochiul uman este un organ intern care este bine protejat împotriva deteriorării și uzurii de către o membrană foarte transparentă și sensibilă. Acest lucru îl deosebește de amprentele digitale, care cu trecerea anilor pot să fie din ce în ce mai greu de recunoscut. Irisul este în mare parte plat, iar în configurația sa geometrică este controlată doar de doi mușchi contempleri (pupilele sfincterului și pupilele dilatatoare) care controlează diametrul pupilei. Acest lucru face forma irisului mult mai previzibilă decât cea a feței.
- Irisul are o textură fină care, asemenea amprentelor digitale, este determinată aleatoriu în timpul gestației embrionare. Ca și amprenta, este foarte greu de dovedit că irisul este unic, poate chiar imposibil. Cu toate acestea, există mulți factori care intră în formarea acestor texturi, șansa de potrivire falsă pentru oricare dintre ele este extrem de scăzută. Chiar și indivizii identici genetic au texturi de iris complet diferite. O scanare a irisului este similară cu fotografierea și poate fi efectuată de la aproximativ 10 cm până la câțiva metri distanță.
- Algoritmul de recunoaștere a irisului al lui John Daugman, IrisCode, are o rată de potrivire falsă fără precedent, deoarece se folosește de un prag de 0,66 de la Hamming, ceea ce înseamnă că până la 26% din biții din două coduri IrisCode au voie să nu fie de acord datorită zgomotului imaginilor și reflexiilor, în timp ce încă le declară o potrivire.

2.4 Dezavantaje

- Multe dintre sistemele comerciale de scanare a irisului pot fi ușor induse în eroare de o imagine de înaltă calitate a unui iris în loc de unul real. Tehnologiile de scanare sunt adesea greu de reglat și pot deveni destul de deranjante pentru diferite persoane. Acuratețea și procentajele pot fi foarte afectate și din cauza iluminării încăperi.
- Sistemele de recunoașterea a irisului sunt mai scumpe decât alte tehnologii biometrice precum sistemele de proximitate și de parole.

Scanarea irisului este o tehnologie destul de nouă, dar nu multe persoane își pot permite acest sistem.

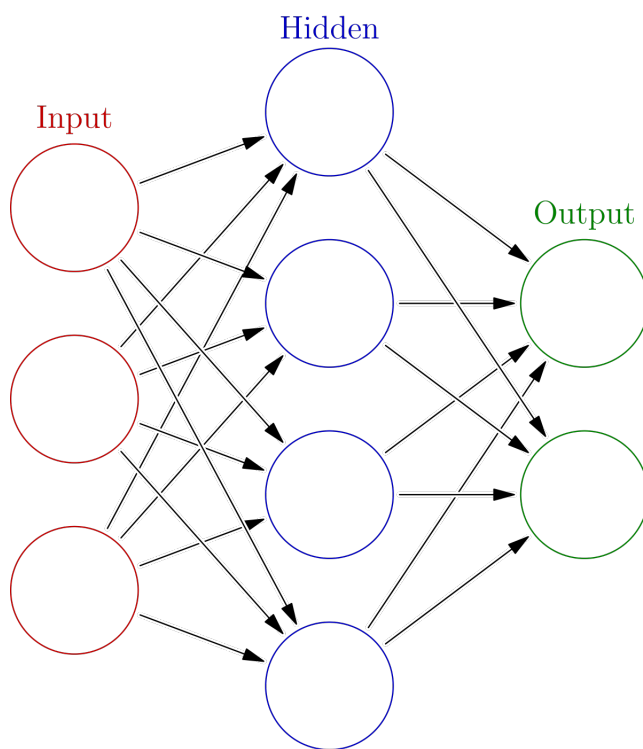
- Această tehnologie este foarte dificil de utilizat de la o distanță mai mare de câțiva metri, iar în cazul în care persoana urmează să fie identificată nu cooperează cu sistemul, acesta nu va returna nici un rezultat sau va afișa un mesaj de eroare.
- Ca și alte tehnologii biometrice fotografice, recunoașterea irisului poate suferi modificări din cauza clarității slabe a imaginii, asociată cu incapacitatea de a înregistra datele.
- În cazul altor infrastructuri de identificare cum ar fi cardurile de identitate, activiștii pentru drepturile civile și-au exprimat îngrijorările cum că sistemele de recunoaștere a irisului ar putea ajuta guvernele să urmărească oamenii dincolo de voința lor. Unii cercetători au pacălit scanerile de iris folosind imagini generate digital ale altor imagini stocate care au la bază un iris, iar persoanele răuvoitoare ar putea utiliza acest defect pentru a fura identitatea altora.

Capitolul 3

Retele Neuronale

Retele neuronale artificiale sunt sistemele de calcul care au fost inspirate de către rețelele neuronale biologice. Astfel de sisteme "învăta" să rezolve sarcini având în considerare diferite exemple și în general fără a fi programate cu reguli specifice fiecărei sarcini. De exemplu, în recunoașterea imaginilor cu chipuri de oameni. Acestea analizează exemple de imagini cu fețe omenești și le compară cu imagini marcate manual ca "față" sau "fără față" și utilizează rezultatele pentru a identifica fețele din alte imagini. Rețelele neuronale fac acest lucru fără nici o cunoaștere prealabilă despre fețe, cum ar fi: ochi, nas, gură. Ele generează automat caracteristici de identificare din materialul de învățare pe care îl prelucreză. Un sistem de rețele neuronale artificiale se bazează pe o colecție de unități sau noduri conectate numite neuroni artificiali, care sunt preluați după modelul de neuroni dintr-un creier biologic. Fiecare conexiune precum sinapsele dintr-un creier biologic, pot transmite un semnal de la un neuron artificial la altul. Un neuron artificial care primește un semnal îl poate procesa și apoi semnalează neuronilor artificiali suplimentari conectați la el. În implementările comune ale rețelelor neuronale artificiale, semnalul la o conexiune între neuronii artificiali este un număr real, iar ieșirea fiecărui neuron artificial este calculată de o anumită funcție neliniară a sumei intrărilor sale. Denumirea legăturilor dintre neuronii artificiali este "margină" sau "marginile". Neuronii și marginile artificiale au în mod obișnuit o greutate care se ajustează ca învățare. Această greutate în diferite cazuri poate să crească sau să scadă puterea semnalului la o conexiune. Neuronii artificiali pot avea un prag, astfel încât semnalul să fie trimis numai dacă semnalul agregat traversează acest prag. În mod obișnuit, neuronii artificiali sunt agregați în mai multe straturi, iar aceste straturi pot efectua diferite tipuri de transformări pe intrările lor. Semnalele se deplasează de la stratul de intrare adică primul strat, până la stratul de ieșire adică ultimul strat,

eventual după traversarea straturilor de mai multe ori. Scopul inițial al abordării sistemelor de rețele neuronale a fost de a ”rezolva problemele în același mod în care l-ar fi rezolvat un creier uman”. Dar, în timp, atenția sa mutat pe îndeplinirea sarcinilor specifice, ceea ce a dus la abateri de la natura umană și de la biologie. Rețelele neuronale artificiale au fost utilizate pentru diferite sarcini, inclusiv traducerea automată, filtrarea rețelelor sociale, viziunea pe calculator, recunoașterea vorbirii și a dialectelor, jocurile video, precum și diagnosticul medical. Warren McCulloch și Walter Pitts (1943) au creat un model computațional pentru rețelele neuronale bazat pe aritmetică și algoritmi numiți logică de prag. Acest model a deschis calea pentru cercetarea în domeniul rețelelor neuronale pentru a se împărți în două abordări.[10] Prima abordare este bazată și axată pe procesele biologice din creierul uman. Iar cea de a doua abordare este bazată pe aplicarea rețelelor neuronale asupra inteligenței artificiale. Aceasta activitate a dus la munca asupra rețelelor nervoase și la legătura lor cu automatele finite. În figura 2 este prezentată o rețea neuronală cu două ieșiri.



Rețea neuronală cu două ieșiri.
Figura 2.

3.1 Backpropagation

Un declansator cheie pentru interesul reînnoit în rețelele neuronale și învățarea a fost algoritmul de Backpropagation a lui Werbos (1975), care a reușit să rezolve în mod eficient problema exclusivă, făcând ca formarea rețelelor multi-strat să fie eficientă. Backpropagation a distribuit termenul de eroare înapoi prin straturi, modificând total greutățile fiecărui nod. Modalitățile de suport vectoriale și alte metode mult mai simple, cum ar fi clasificatorii liniari, au depășit treptat rețelele neuronale în popularitatea învățării mașinilor. Cu toate acestea, utilizarea rețelelor neuronale au transformat anumite domenii, cum ar fi predicția structurilor proiectate. În 1992, s-a introdus max-polling-ul pentru a susține și ajuta varianta minimă a schimbării și la toleranța de deformare a obiectelor pentru a ajuta la recunoașterea de obiecte tridimensionale. În 2010 formarea Backpropagation prin maximizarea pool-urilor a fost accelerată de GPU-uri și demonstrat că este mult mai bună decât alte variante de pooling. Problema dispariției gradientului afectează destul de mult rețelele de alimentare cu mai multe straturi care utilizează Backpropagation și rețelele neuronale recurente. Pe măsura ce erorile se propagă de la strat la strat acesta scade exponențial cu numărul de straturi aferente, astfel împiedicând reglarea greutăților neuronilor care au la bază aceste erori, afectând în special rețelele profunde. [11] Pentru o greutate care unește un nod în stratul k cu un nod din stratul j , modificarea greutății este dată de:

$$\Delta w_{i,j}(n) = \alpha \delta_j y_k + \eta \delta w_{k,j}(n-1)$$

Unde:

- α este rata de învățare, o valoare reală pe interval $(0,1]$.
- y_k este activarea nodului în stratul k , adică activarea nodului pre-sinaptic, cel în amonte față de greutate;
- n și $n - 1$ se referă la iterația prin buclă
- η este impulsul, o valoare reală pe intervalul $[0,1]$
- δ_j este "termenul de eroare" asociat cu nodul după greutate, adică nodul postsinaptic.

3.2 Feedforward propagation

O retea neuronală de alimentare înainte este o retea neuronală artificială în care conexiunile dintre noduri nu formează un ciclu[1]. Din acest motiv sunt diferite de rețelele neuronale recurente. Rețeaua neuronală de alimentare înainte a fost primul și cel mai simplu tip de retea neuronală artificială concepută[2]. În această retea, informațiile se deplasează într-o singură direcție, înainte, de la nodurile de intrare, prin nodurile ascunse și prin nodurile de ieșire. Nu există cicluri sau bucle în retea.[12]

3.2.1 Multi-layer

Această clasă de rețele neuronale constă din mai multe straturi de unități computaționale, de obicei interconectate într-un mod feed-forward. Fiecare neuron într-un singur strat, direcționează conexiunile la neuronii stratului următor. În multe aplicații, unitățile acestor rețele aplică o funcție sigmoidă ca funcție de activare. Teorema universală de aproximare pentru rețelele neuronale afirmă că orice funcție continuă care măpăază intervalele de numere reale la un anumit interval de ieșire de numere reale poate fi aproximată în mod arbitrar de un receptor cu mai multe straturi, cu un singur strat ascuns. Acest rezultat este valabil pentru o gamă largă de funcții de activare, de exemplu: pentru funcțiile sigmoidale.[13]

3.3 Gradient descent

Gradient descent este un algoritm de optimizare iterativă de ordinul întâi pentru a găsi minimumul unei funcții. Pentru a găsi un minim local al unei funcții folosind Gradient Descent se iau pași proporționali cu negativul gradientului sau un gradient aproximativ al funcției la punctul curent.[14] Dar dacă în schimb se iau pași proporționali cu poziția gradientului, se poate ajunge la un maxim local de această funcție, procedura este apoi cunoscută ca ascensiune în gradient.

Pseudocod

```
x,y <- next training data point
y_hat = model(x; weights)
this_loss = loss(y_hat, y)
grad_loss = d/dy_hat loss(y_hat, y)
grad_weights = grad_loss * d/dweights model(x; weights)
weights <- weights - step_size * grad_weights
```

Capitolul 4

Aplicație

4.1 Tehnologii utilizate

4.1.1 OpenCV

OpenCv (Open Source Computer Vision) este o ”bibliotecă de funcții de programare care vizează în primul rând viziunea computerului în timp real”. Această bibliotecă a fost dezvoltată inițial de către compania Intel , iar ulterior a fost susținut de Willow Garage, apoi de Itseez. Biblioteca este o platforma cross-platform și este gratuită pentru utilizarea sub licența BSD openn-source. Cercetarea avansată a viziunii oferind nu numai cod deschis , dar și optimizat, pentru infrastructura de viziune de baza. Împartășirea diferitelor cunoștințe despre viziunea prin furnizarea unei infrastructuri comune pe care dezvoltatorii ar putea să o construiască, astfel încât codul să fie mult mai ușor de citit și mai ușor transferat. Aceasta facand codul portabil, optimizat pentru performanță cu o licență care nu necesită ca codul să fie deschis. Prima versiune alfa OpenCv a fost lansată publicului la Conferința IEEE privind Computer Vision și Recognition Tatter în anul 2000 și alte cinci etape-uri au fost lansate pe parcursul anilor 2001 și 2005. Prima versiune de 1.0 a fost lansată în anul 2006. Cea de a doua versiune lansată a programului OpenCv a fost în anul 2009. Această aduce schimbări majore în interfața C++, având ca și scop modele mai ușoare, mai sigure de tip, noi funcții și implementări mai bune pentru cele deja existente la vremea aceea. Biblioteca OpenCv este în scrisă în C++ și interfața sa primară este scrisă tot în C++, dar păstrează o interfață de C mai puțin cuprinzătoare. OpenCv are legături în limbaje precum Python, Java, sau Matlab. Legăturile pentru alte limbaje de programare cum ar fi C#, Perl sau Ruby au fost dezvoltate pentru a încuraja adoptarea de către public cu mai multă ușurință. Toate noile dezvoltări și toți algoritmi

din OpenCv sunt acum dezvoltate în interfața C++. Aplicații OpenCv: Robotica mobilă, Recunoaștere gesturilor, Urmărirea mișcării, Realitate augmentată, Structura din Mișcare, Înțelegerea Mișcării, Interacțiunea on-calculator, Sisteme de recunoaștere a feței, Segmentare și recunoaștere. Pentru a susține câteva dintre domeniile prezentate mai sus, OpenCv include o bibliotecă statică de învățare a mașinilor (machine-learning) care conține următoarele: Gradient descent, Rețele Neuronale Artificiale(ANN), Rețele Neuronale Profunde(DNN), sau stimulare.[5]

4.1.2 Haar-Cascades

Detectarea diferitelor obiecte utilizând Haar-Cascades bazați pe caracteristici Haar este o metodă eficientă de detectare a obiectelor recomandată de Paul Viola și Michael Jones în lucrarea "Detectarea rapidă a obiectelor utilizând o cascadă de caracteristici simple" din 2001. Haar-Cascades sunt disponibile în OpenCV, cu cascade pre-instruite pentru fețele frontale și partea superioară a corpului. Formarea unei noi cascade în OpenCV este posibilă și cu metodele `haar_training` sau `train_cascades`. Aceasta poate fi utilizată pentru detectarea rapidă a diferitelor obiecte cu obiective mai specifice, inclusiv obiecte non-umane cu caracteristici asemănătoare cu Haar. Procesul necesită două seturi de probe: una negativă și una pozitivă, unde mostrele negative corespund imaginilor arbitrare non-obiect. Constrângerea de timp în formarea unui clasificator în cascadă poate fi eludată utilizând metode computerizate în cloud.[6]

Cod

```
cv::CascadeClassifier face_cascade;  
cv::CascadeClassifier eye_cascade;
```

4.1.3 NumPy

NumPy este o bibliotecă pentru limbajul de programare Python, adăugând suport pentru matrici, matrici mari, multidimensionale, împreună cu o mare colecție de funcții matematice de nivel înalt care funcționează pe aceste matrici. NumPy a fost inițial creat de Jim Hugunin, cu ajutor din partea a câtorva alți dezvoltatori. În 2005, Travis Oliphant a creat NumPy prin încorporarea caracteristicilor Numarray concurente în Numeric, cu modificări extinse. NumPy este un software open source și are mulți contribuitori. NumPy vizează implementarea de referință CPython a

Python, care este un interpret de tip bytecode care nu este optimizat. Algoritmii matematici scrise pentru această versiune de Python, rulează adesea mult mai încet decât echivalentele compilate. NumPy abordează problema lentă parțial prin furnizarea de rețele și funcții multidimensionale și operatori care funcționează eficient pe array, necesitând rescrierea unor coduri, mai ales a unor bucle interioare folosind NumPy.[7]

4.1.4 Editoare utilizate

Microsoft Visual Studio este un mediu integrat de dezvoltare (IDE) al companiei Microsoft. Acesta este folosit pentru a dezvolta programe de calculator, precum și site-uri Web, aplicații web, servicii web și aplicații mobile. Visual Studio utilizează platforme de dezvoltare software Microsoft cum ar fi Windows API, Windows Forms, Windows Presentation Foundation, Windows Store și Microsoft Silverlight. Poate produce atât cod nativ, cât și cod gestionat. Visual Studio suportă 36 de limbi de programare diferite la ora actuală și permite editorului de cod și depanatorului să suporte aproape orice limbaj de programare, cu condiția să existe un serviciu specific limbajului de preprogramare. Limbile încorporate includ C, C++, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML și CSS. Suportul pentru alte limbi, cum ar fi Python, Ruby, Node.js și M, printre altele, este disponibil prin plug-in-uri. Ediția cea mai de bază a Visual Studio, ediția comunitară, este disponibilă gratuit.[8]

Python IDLE IDLE (scurt pentru mediul integrat de dezvoltare sau mediul integrat de dezvoltare și învățare) este un mediu de dezvoltare integrat pentru limbajul de programare Python, care a fost asociat cu implementarea implicită a limbajului de la 1.5.2b1. Este ambalat ca o parte opțională a ambalajului Python cu multe distribuții pentru sistemul de operare Linux. Este scris complet în Python și setul de instrumente Tkinter GUI.[9] IDLE este destinat a fi un IDE simplu și potrivit pentru începători, în special într-un mediu educațional. În acest scop, este cross-platform și evită confuzia caracteristicilor.

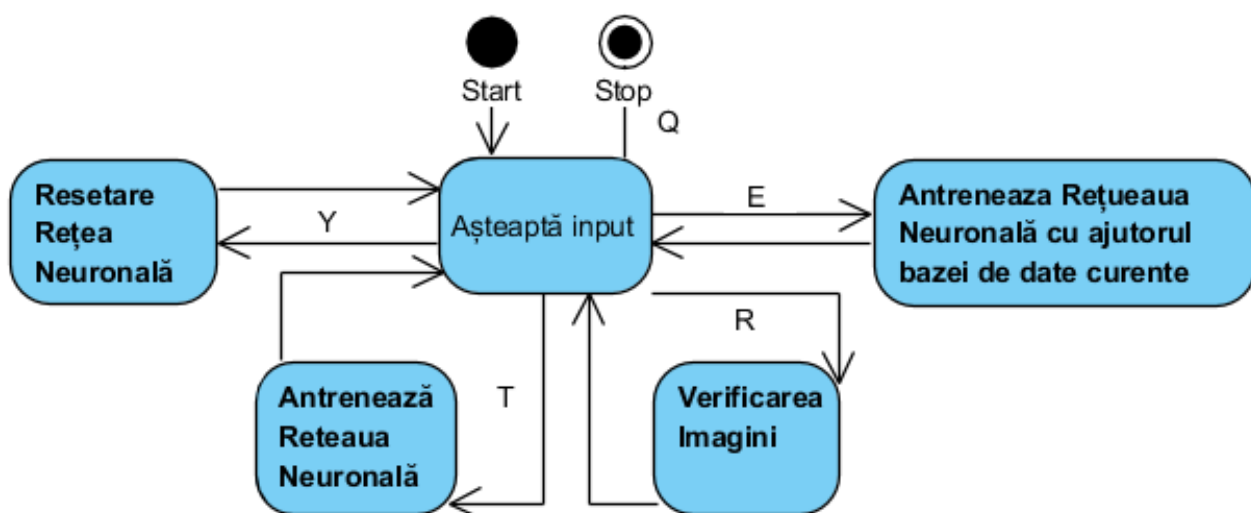
4.2 Implementare

Aplicația I.R.U.N.N. a fost implementată în două părți:

- Prima parte în limbajul de programare C++ cu ajutorul bibliotecii OpenCv.

- A doua parte în limbajul de programare Python cu ajutorul bibliotecilor NumPy si PIL(pentru încărcarea imaginilor).

Aplicația I.R.U.N.N.(Iris Recognition using Neural Netorks) funcționează conform următoarei figuri(Figura 3):

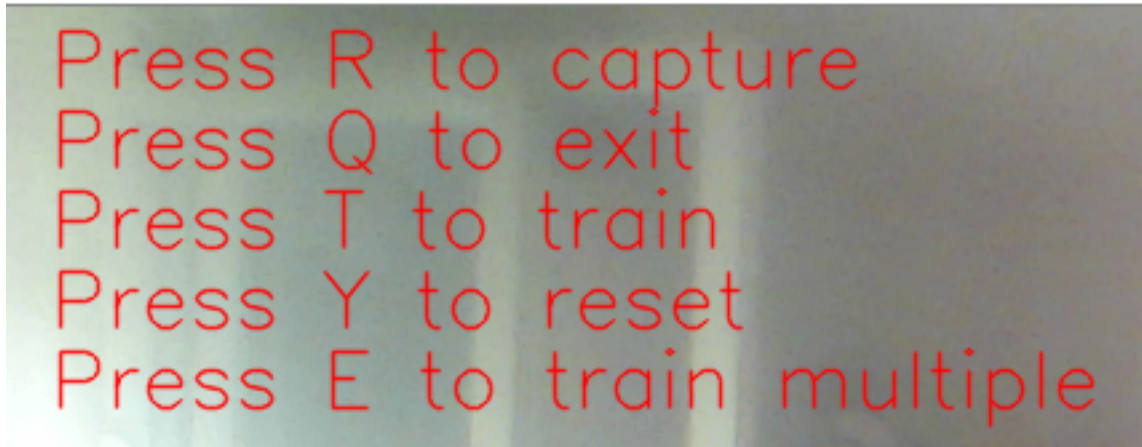


Funcționarea aplicației.

Figura 3

Aplicația I.R.U.N.N.(Iris Recognition using Neural Netorks) înregistrează tot ce se afala pe video-feed pâna la apăsarea urmatoarelor taste :

- Prima este tastă Q, care realizează ieșirea din aplicație și închiderea tuturor ferestrelor deschise.
- Cea de a doua este tastă R, care realizează captura imagini și verificarea acesteia în cele din urmă.
- Cea de a treia tastă este T, această tastă antrenează rețeaua neuronală. De fiecare dată cand tasta T este utilizată, rețeaua neuronală va fi antrenată în funcție de imaginea selectată.
- Cea de a patra tastă este E. Această tastă antrenează rețeaua neuronală cu ajutorul bazei de date existente folosind imaginile create anterior în modul de antrenare a rețelei neuronale(tasta T).
- Cea de a cincea tastă este Y. Aceasta realizeaza resetarea rețelei neuronale, reinițializând toate greutatea și bias-urile rețelei neuronale cu valori aleatori.



Meniul aplicației.

Figura 4

În figura (Figura 4) de mai sus se poate observa meniul aplicației.

4.2.1 Detectarea ochilor

Pentru capturarea imaginii, utilizăm biblioteca OpenCV pentru a prelua input-ul de la camera video. După ce imaginea este capturată se realizează detectarea feței prin librăria Haar-Cascades cu ajutorul următoarelor secvențe de cod:

```
cv::CascadeClassifier face_cascade;
```

- Se realizează declararea.

```
face_cascade.detectMultiScale(im, faces, 1.1, 2, 0 |  
CV_HAAR_SCALE_IMAGE, cv::Size(30, 30));
```

- Se realizează inițializarea.

```
face_cascade.load("haarcascade_frontalface_alt2.xml");
```

- Se realizează detectarea propriu zisă.

După ce imaginea este capturată urmează să se detecteze ochii în zona fețelor capturate prin utilizarea următoarelor funcții:

```
cv::CascadeClassifier eye_cascade;
```

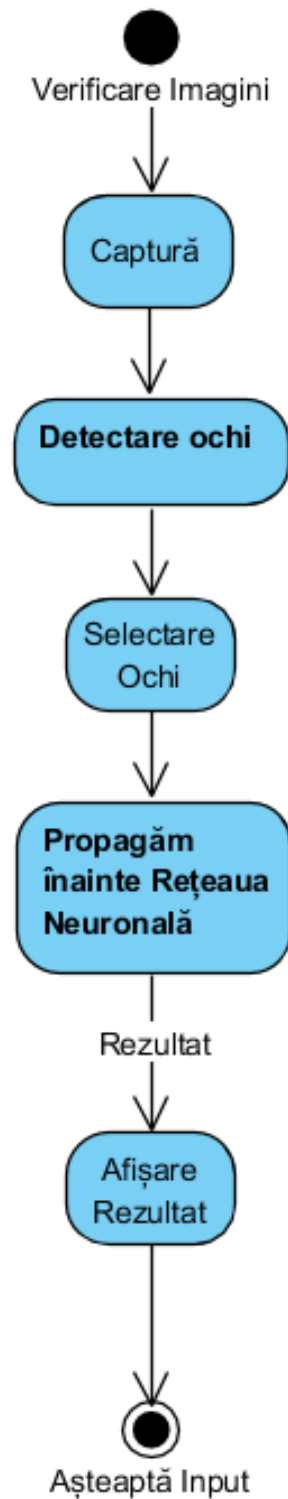
- Se realizează declararea.

```
eye_cascade.detectMultiScale(face, eyes, 1.1, 2, 0 |  
CV_HAAR_SCALE_IMAGE, cv::Size(20, 20));
```

- Se realizează inițializarea.

```
eye_cascade.load("haarcascade_eye.xml");
```

- Se realizează detectarea propriu zisă.



Algoritmul de detectare a ochilor.
Figura 5.

În figura de mai sus (Figura 5) este prezentat algoritmul utilizat pentru detectarea ochilor. Acest algoritm constă în următorii pași:

- Primul pas constă în capturarea imaginii de pe camera video. După

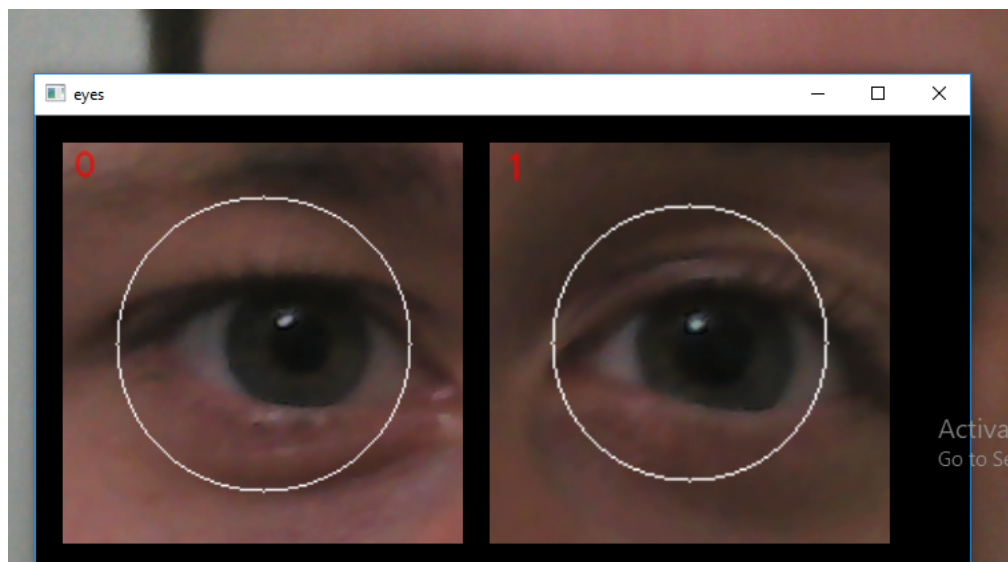
cum se poate observa și în figura de mai jos (Figura 6) .



Captura imaginii.

Figura 6

- În al doilea pas, utilizând captura de imagine de la pasul anterior, se realizează detectarea ochilor din imagine. În figura de mai jos (Figura 7) se pot observa toți ochii capturați la momentul dat.



Captura ochilor din imagine.

Figura 7

Pentru a se realiza detectarea tuturor ochilor din imaginea capturată utilizăm următoarea funcție:

```

std::vector<cv::Rect> detectEye(cv::Mat& im)
{
    std::vector<cv::Rect> faces, eyes, eys;
    face_cascade.detectMultiScale(im, faces, 1.1, 2, 0 |
        CV_HAAR_SCALE_IMAGE, cv::Size(30, 30));
    for (int i = 0; i < faces.size(); i++)
    {
        cv::Mat face = im(faces[i]);
        eye_cascade.detectMultiScale(face, eyes, 1.1, 2, 0 |
            CV_HAAR_SCALE_IMAGE, cv::Size(20, 20));
        for (int j = 0; j < eyes.size(); j++)
        {
            cv::Rect rect = eyes[j] + cv::Point(faces[i].x, faces[i].y);
            eys.push_back(rect);
        }
    }
    return eys;
}

```

Pentru început algoritmul detectează toate fețele din imagine. Pentru fiecare față care este încadrată într-un dreptunghi se vor căuta posibili ochi în respectivul dreptunghi. Dacă au fost găsiți ochii din dreptunghiul care delimitează fața, aceștia se vor adăuga într-un vector, care va fi returnat la finalul funcției.

- În al treilea pas se selectează unul dintre ochii capturați în imagine pentru ca imaginea să poată fi transmisă mai apoi către rețeaua neuronală.
- Al patrulea pas constă în propagarea înainte (Forward propagation) a rețelei neuronale. Acesta va returna rezultatul verificării.
- Iar în al cincelea pas se va afișa rezultatul pe ecran.

4.2.2 Rețea neuronală

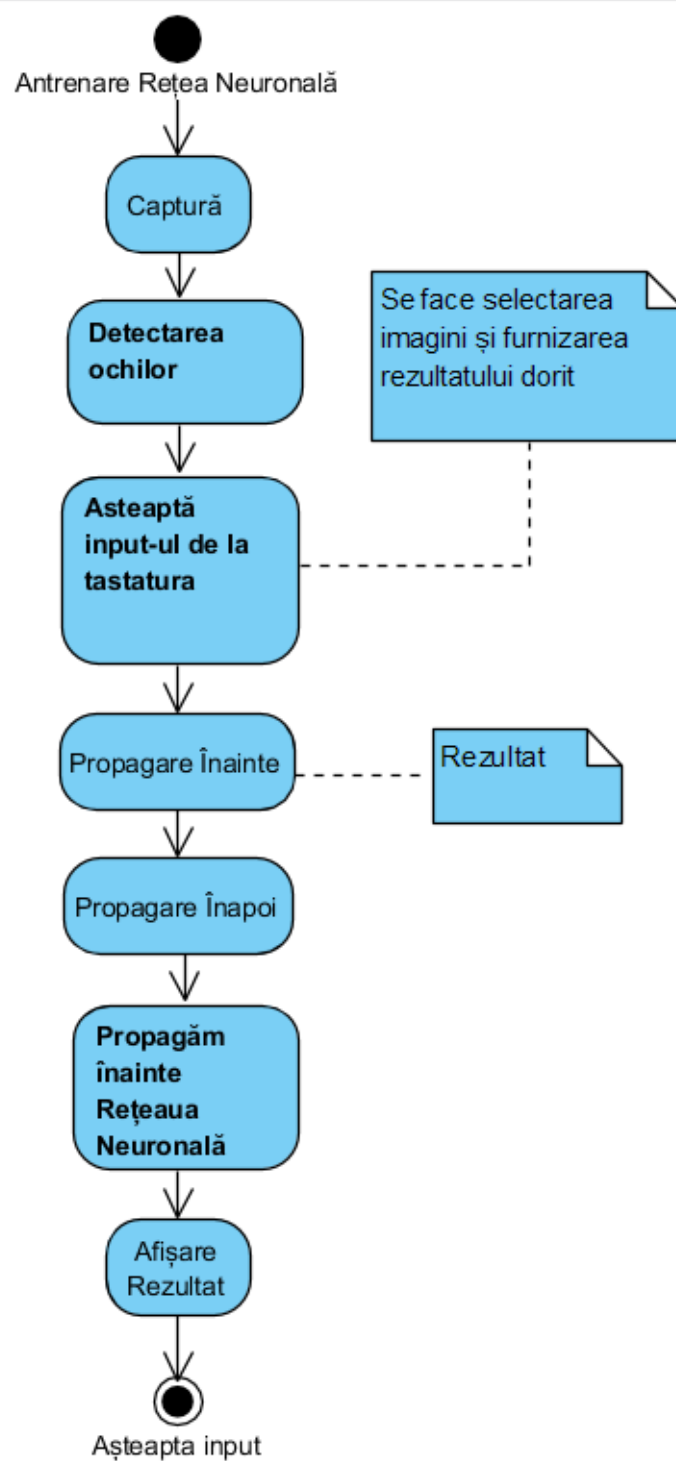
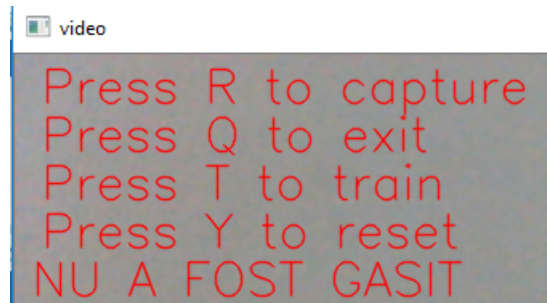


Figura de antrenare a rețelei neuronale.

Figura 8.

În figura de mai sus (Figura 8) este prezentat algoritmul utilizat pentru antrenarea rețelei neuronale utilizate. Acest algoritm constă în următorii pași:

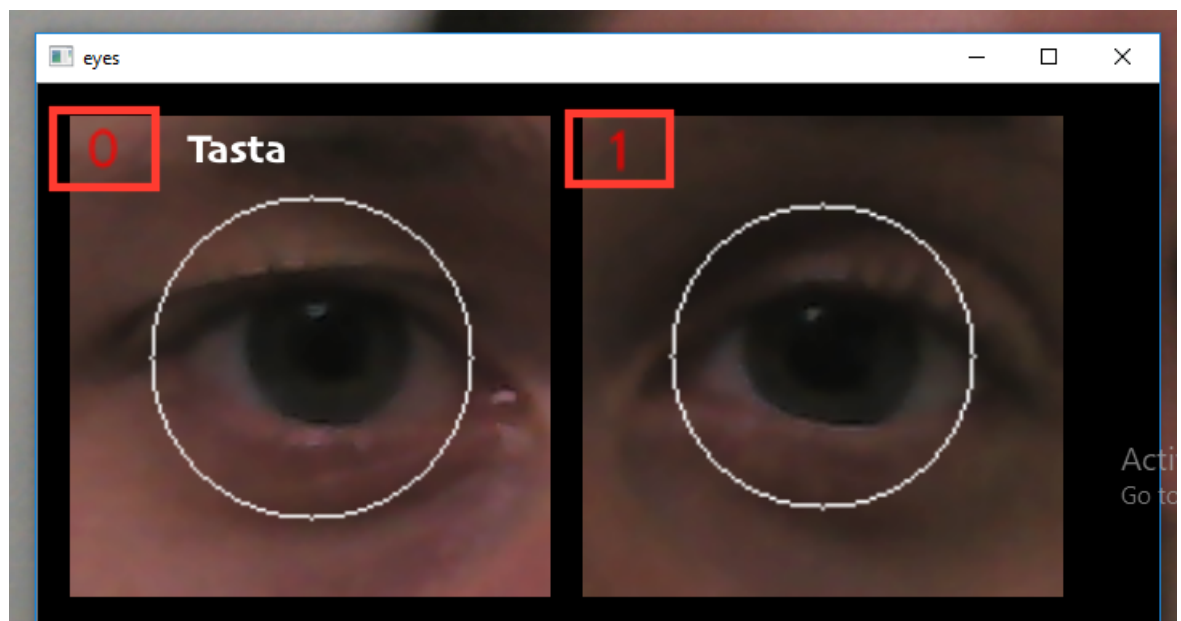
- În primul pas, se realizează o captură a imaginii de pe camera video.
- În al doilea pas, se realizează detectarea ochilor prin imaginea din pasul anterior. Dacă nu se găsește nici un ochi la detectare, se va afișa mesajul "NU A FOST GASIT". Acest mesaj apare sub butoanele din meniu. Mesajul observă se observă și în figura de mai jos (Figura 9).



Mesajul "NU A FOST GASIT".

Figura 9.

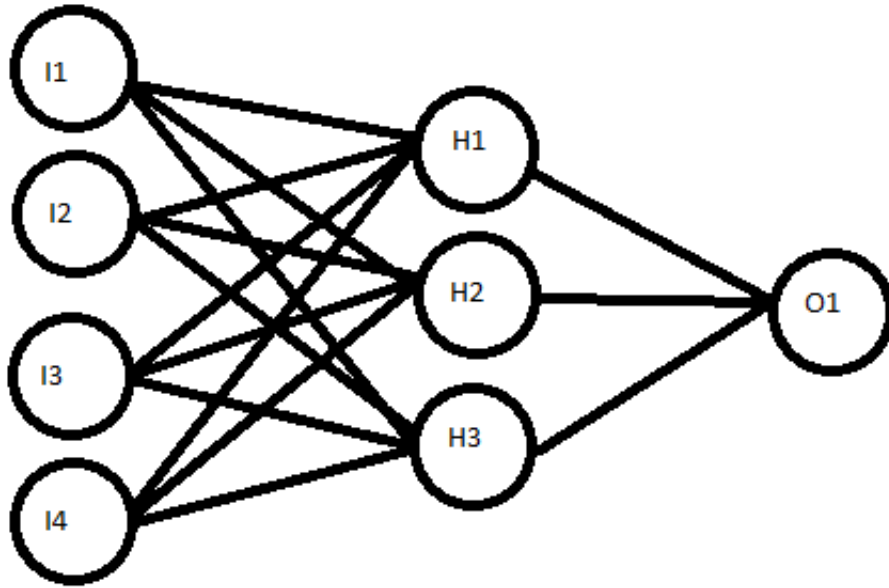
- În al treilea pas, se așteaptă un input de la tastatură, iar aceasta în selectarea imaginii și furnizarea rezultatului dorit. În figura (Figura 10) de mai jos se poate observa acest lucru.



Slecetare imagine dorita.

Figura 10.

- În al patrulea pas se realizează propagarea înainte (Forward propagation), fiecare intrare de input se înmulțește cu greutatea respectivă pentru a fi adunate împreună, după care se va aduga valoarea bias, iar acest rezultat va fi trecut în funcția de activare ca și output a nodului următor.



Forward propagation.

Figura 11.

În figura de mai sus (Figura 11) se află o rețea neuronală cu o singură ieșire. Aceasta este mult mai mică decât rețeaua neuronală utilizată în aplicație, dar funcționează pe exact același principiu.

$$I = \begin{pmatrix} I1 \\ I2 \\ I3 \\ I4 \end{pmatrix}$$

Matricea I reprezintă datele de intrare vectoriale, iar în cazul nostru imaginea a fost vectorizată, prin punerea pentru fiecare punct în vector a trei valori care reprezintă culoarea(RGB).

$$W = \begin{bmatrix} W11 & W12 & W13 \\ W21 & W22 & W23 \\ W31 & W32 & W33 \end{bmatrix}$$

Matricea W se află greutatea dintre noduri. $W_{i,j}$ reprezintă greutatea legături dintre nodul de intrare I_i și nodul H_j .

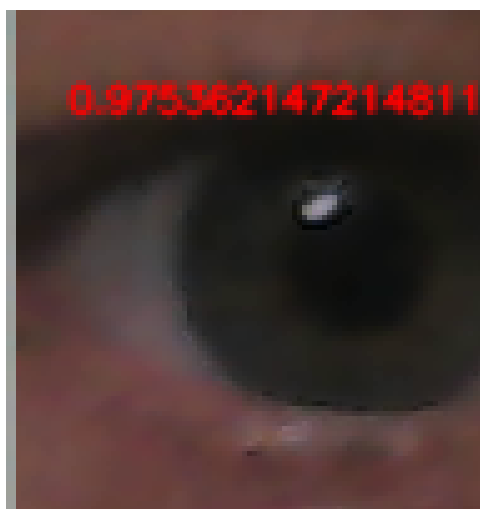
$$b = \begin{pmatrix} b1 \\ b2 \\ b3 \\ b4 \end{pmatrix}$$

Matricea b este vectorul bias-urilor care se adună pentru a obține valoarea nodului înainte ca el să fie trecut prin funcția de activare.

$$H = \sin(I * W + b)$$

Prin formula prezentată anterior se calculează vectorul de noduri H și analog se procedează pentru a obține valorile următoarelor nivele de noduri.

- În al cincelea pas se realizează porcedeul de propagare înapoi (Backward propagation).
- În al șaselea pas, se aplică din nou porcedeul de propagare înainte, doar că de această dată este aplicată pe rețeaua neuronală.
- Iar în ultimul pas se realizează afișarea rezultatului. Se poate observa rezultatul în figura de mai jos (Figura 12).



Rezultatul.

Figura 12.

Pentru a se realiza antrenarea rețelei neuronale se utilizează următoarea structură de cod:

```
void train(cv::Mat & frame,int e) {
std::cout << "1 sau 0:";
e = (e == 0 ? 0 : 1);
saveImage("ochi.jpg", frame);
std::string filename1 = "./savedImgs/" +
    std::to_string(e) + "/" + getTime() + ".jpg";
cv::imwrite(filename1, frame);
std::string m = cmdPyth("./ochi.jpg", e);
    cv::Scalar(0, 0, 255), 1, CV_AA);
analizImg(frame);
cv::imshow("eyes", frame);
}
```

În primul rând, aceasta preia ca și parametru imaginea cu ochiul selectat și raspunsul pe care îl așteaptă de la rețeaua neuronală. Pentru a putea să utilizăm rețeaua neuronală scrisă în Python, salvăm imaginea primită pe disk într-un fișier temporal cu numele Ochi.jpg. În al doilea rând pentru a putea reantrena în viitor rețeaua neuronală, imaginea se mai salvează și în locația: /savedImgs/1 , în fișierul cu numărul 1 se vor salva mereu imaginile care sunt compatibile, iar în /savedImgs/0 se vor salva cele care nu sunt compatibile. După acestea se va apela rețeaua neuronală care va primi ca și argumente imaginea și rezultatul așteptat.

4.3 Rezultate experimentale

În tabelul de mai jos (Figura 13) se pot observa rezultatele experimentale ale acestei aplicații:

```
===== RESTART: C:\Users\blank\Desktop\Proiect\Mark\Mark\trainModule.py =====
./savedImgs/0/2018.08.29-03h01m19s.jpg 0 [[0.70448168]] 0.9925888773511966
./savedImgs/0/2018.08.29-03h03m00s.jpg 0 [[0.03680018]] 0.00270850605171624
./savedImgs/1/2018.08.29-21h29m11s.jpg 1 [[0.93906037]] 1.763668753141199
./savedImgs/1/2018.08.29-02h03m54s.jpg 1 [[0.00575706]] 6.628754184859268e-05
./savedImgs/0/2018.08.29-03h15m05s.jpg 0 [[0.96565342]] 1.86497305462484
./savedImgs/0/2018.08.29-01h00m36s.jpg 0 [[0.00233794]] 1.0931922587104983e-05
./savedImgs/0/2018.08.29-03h03m18s.jpg 0 [[0.39549915]] 0.31283915112176486
./savedImgs/0/2018.08.29-03h00m51s.jpg 0 [[0.68509576]] 0.9387123982022209
./savedImgs/0/2018.08.29-03h06m40s.jpg 0 [[0.0210369]] 0.0008851023665807912
./savedImgs/1/2018.08.29-02h55m35s.jpg 1 [[0.08266364]] 0.013666553852714713
./savedImgs/0/2018.08.29-01h56m40s.jpg 0 [[0.02489034]] 0.0012390576657620178
./savedImgs/1/2018.08.29-03h18m27s.jpg 1 [[0.14503285]] 0.042069052760669065
./savedImgs/1/2018.08.29-02h59m20s.jpg 1 [[0.10527371]] 0.022165105966760742
./savedImgs/0/2018.08.29-02h59m53s.jpg 0 [[0.63000501]] 0.7938126219599358
./savedImgs/0/2018.08.29-03h17m04s.jpg 0 [[0.01187081]] 0.00028183222713596
./savedImgs/0/2018.08.29-02h53m29s.jpg 0 [[0.22262349]] 0.09912243433054536
./savedImgs/0/2018.08.29-03h23m13s.jpg 0 [[0.55190508]] 0.6091984284921403
./savedImgs/0/2018.08.29-03h03m25s.jpg 0 [[0.02020866]] 0.0008167795039235925
./savedImgs/0/2018.08.29-03h03m10s.jpg 0 [[0.0772488]] 0.011934754799844072
./savedImgs/0/2018.08.29-03h02m16s.jpg 0 [[0.22255331]] 0.0990599526740943
./savedImgs/1/2018.08.29-02h01m22s.jpg 1 [[0.48959106]] 0.4793988089632024
./savedImgs/1/2018.08.29-02h55m17s.jpg 1 [[0.08564882]] 0.014671439911882007
./savedImgs/0/2018.08.29-01h14m29s.jpg 0 [[0.0169752]] 0.0005763147039247752
./savedImgs/1/2018.08.29-21h28m23s.jpg 1 [[0.02565768]] 0.0013166329143494443
./savedImgs/0/2018.08.29-02h52m56s.jpg 0 [[0.95118425]] 1.8095029546326298
./savedImgs/0/2018.08.29-01h57m42s.jpg 0 [[0.99186308]] 1.9675847486868443
./savedImgs/0/2018.08.29-03h07m00s.jpg 0 [[0.19942056]] 0.07953711976233603
./savedImgs/0/2018.08.29-03h07m17s.jpg 0 [[0.7330093]] 1.0746052790239158
./savedImgs/0/2018.08.29-01h01m02s.jpg 0 [[0.25363923]] 0.1286657138447499
./savedImgs/0/2018.08.29-00h58m05s.jpg 0 [[0.07831425]] 0.012266242921707972
./savedImgs/0/2018.08.29-03h06m52s.jpg 0 [[0.07070327]] 0.009997903505110232
./savedImgs/0/2018.08.29-02h58m46s.jpg 0 [[0.04319175]] 0.003731055095035438
```

Tabelul experimental.

Figura 13

- Pe prima coloană din imaginea de mai sus: ./savedImgs/0/2018.08.29-03h01m19s.jpg, se află imaginea salvată.
- În a doua coloană se află rezultatele așteptate, de exemplu: 0 [[0.70448168]]
- Iar pe a treia coloană sunt afișate rezultatele obținute după ce imaginea a fost trecută odată prin rețeaua neuronală (Feedforward) de exemplu: 0.9925888773511966.

Cu cât rezultatul furnizat de rețeaua neuronală este mai aproape de 1 cu atât eroarea va fi mai mică, ceea ce înseamnă ca imaginea din baza de date cu imaginea proaspăt capturată sunt foarte identice.

Capitolul 5

Concluzii

5.1 Evaluarea Aplicatiei

În aceasta lucrare am analizat modul în care se comportă rețeaua atunci când este dată o intrare și pentru care are o rată de eroare specificată. Rețeaua a fost pregătită și testată pentru o serie de imagini oculare. I.R.U.N.N. este o aplicație care detectează ochiul și verifică irisul cu ajutorul unei rețele neuronale pentru a determina dacă irisul este valid sau nu. I.R.U.N.N. antrenează o rețea neuronală pentru a detecta diferite tipuri de irisuri. Aplicația se află momentan în stadiul în care recunoaște irisul unei singure persoane.

Accentul este pus pe software pentru a efectua recunoașterea, și nu hardware-ul pentru captarea unei imagini.

5.2 Directii viitoare de dezvoltare

În urma rezultatelor favorabile obținute, aplicația I.R.U.N.N. se mai poate îmbunătăți prin următoarele metode:

- În primul rând se vor realiza câteva teste pe diferite plăci de: Arduino și Raspberry pie.
- În al doilea rând se va implementa o aplicație pentru tablete, telefoane mobile și se vor efectua câteva teste împreună cu câteva plăci de Arduino pentru a se verifica conexiunile dintre acestea două.
- După primii doi pași menționați mai sus, aplicația va fi prelucrată și instalată ca și sistem de securitate pentru o casă inteligentă (Smart House).

- Creerea unei baze de date mult mai mari pentru a putea fi utilizat de cât mai multe persoane.
- Creerea unui sistem de criptare pentru a crește nivelul imaginilor stocate în sistem și pentru a păstra confidențialitatea utilizatorilor aplicației.
- Direcția pe care se va axa aplicația cel mai mult, este implementarea acesteia pe sistemele de căști V/R (Virtual Reality) pentru utilizarea aplicației I.R.U.N.N. ca și un sistem de parole pentru utilizatori.

Bibliografie

- [1] John Daugman, Introduction to Iris Recognition, URL: <https://www.cl.cam.ac.uk/~jgd1000/search.html>
- [2] John Daugman, How Iris Recognition Works, URL: <https://www.cl.cam.ac.uk/~jgd1000/history.html>
- [3] Industry Insights, URL: <https://www.grandviewresearch.com/industry-analysis/biometrics-in-healthcare-market>
- [4] Deployed applications, URL: https://en.wikipedia.org/wiki/Iris_recognition
- [5] OpenCv, URL: <https://en.wikipedia.org/wiki/OpenCV>
- [6] Haar-Cascades, URL: https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html
- [7] David Ascher Paul F. Dubois Konrad Hinsien Jim Hugunin Travis Oliphant, NumPy URL: <https://www.cs.mcgill.ca/~hv/articles/Numerical/numpy.pdf>
- [8] Microsoft Visual Studio URL: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- [9] Python IDLE(integrated development and learning environment), URL: <https://en.wikipedia.org/wiki/IDLE>
- [10] Neural Networks URL: https://en.wikipedia.org/wiki/Neural_network
- [11] J.G. Makin, Backpropagation, J.G. Makin,2006 , URL: <https://inst.eecs.berkeley.edu/~cs182/sp06/notes/backprop.pdf>
- [12] Feedforward propagation, URL: https://en.wikipedia.org/wiki/Feedforward_neural_network

- [13] Multi-layer perceptron, URL: https://en.wikipedia.org/wiki/Feedforward_neural_network
- [14] Gradient descent, URL: https://en.wikipedia.org/wiki/Gradient_descent