

Autentificare pe telefon mobil prin recunoașterea feței

Julia Kovacs

Universitatea Tehnică Cluj-Napoca

Str. Memorandumului nr. 28, 400114,

Cluj-Napoca

kovacs_julia@student.utcluj.ro

Dorian Gorgan

Universitatea Tehnică Cluj-Napoca

Str. Memorandumului nr. 28, 400114,

Cluj-Napoca

dorian.gorgan@cs.utcluj.ro

REZUMAT

Lucrarea prezintă implementarea, testarea și evaluarea unei tehnici de autentificare pe telefoane Android prin recunoașterea feței. Metoda se bazează pe algoritmul Eigenface, care permite variații mari ale expresiei faciale. Abordarea Eigenface este des utilizată datorită simplității implementării și capacității de învățare, nefiind nevoie de cunoștințe de geometrie sau de descrieri specifice explicite ale feței. Performanțele algoritmului sunt evaluate experimental prin calcularea ratei de acceptare și a ratei de respingere. Algoritmii sunt implementați în mediul unui serviciu Web, ale cărui metode sunt apelate dintr-o aplicație dezvoltată pentru telefoane mobile având sistem de operare Android.

Cuvinte cheie

Recunoașterea feței, eigenface, telefoane Android.

Clasificare ACM

H5.2. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCERE

Scopul lucrării este implementarea și evaluarea unui sistem, care aplică metoda Eigenface pentru autentificarea bazată pe recunoașterea feței utilizatorului. Vrem ca acest sistem să fie simplu și eficient, să fie capabil să recunoască într-un domeniu limitat, cum ar fi persoanele unui grup predefinit, angajații unei firme, sau studenții unei facultăți. Pentru a realiza implementarea unui astfel de sistem pentru telefoane mobile, trebuie făcut un compromis între calitatea imaginii și complexitatea de calcul a algoritmului. Acest compromis este impus de capacitatea hardware și software limitată a unui telefon mobil inteligent.

Cum telefoanele mobile devin din ce în ce mai performante, devine foarte importantă și securitatea datelor stocate în telefoane, cum ar fi adresele de e-mail, parole, documente private etc. Cele mai multe telefoane inteligente folosesc parole pentru protecția datelor. Un sistem bazat pe recunoașterea feței ar putea fi o soluție mult mai sigură și flexibilă, deoarece nu trebuie să ținem minte parola, și putem să restricționăm accesul la date pentru diferiți utilizatori. Aplicația poate fi folosită pentru deblocarea telefonului sau pentru acordarea autorizației la o tranzacție, dacă utilizatorul este recunoscut [1].

Recunoașterea feței nu este un obiectiv simplu de realizat. Dificultatea constă în contextul și varietatea în care fețele sunt preluate, chiar și pentru o aceeași persoană. În ultimii

ani, cercetătorii au propus mai multe metode de recunoaștere a feței, printre care metoda Eigenface este una dintre cele mai populare [2]. Metoda se bazează pe proiectarea liniară a spațiului imaginii într-un spațiu cu dimensiune redusă [3].

Metoda Eigenface utilizează Analiza Componentelor Principale (ACP) pentru a reduce dimensionalitatea. Analiza ACP are rezultatele cele mai valoroase din algebra liniară aplicată. ACP este folosită în toate formele de analiză, de la neuroștiință la grafica pe calculator, pentru că este o metodă de extragere a informațiilor relevante din seturi de date complicate fără a necesita parametri suplimentari [4].

ALTE REALIZĂRI

Recunoașterea feței este un domeniu de cercetare foarte activ, care se referă atât la detectarea feței, cât și la identificarea (clasificarea) feței. Numeroase tehnici au fost implementate pentru a detecta și a recunoaște fețele. În lucrarea [1] autorii au realizat în Matlab o aplicație pentru recunoașterea feței folosind algoritmul Fisherface. Au implementat o aplicație pentru telefoane mobile, care pe baza fișierelor de ieșire obținute din Matlab, realizează recunoașterea. Îmbunătățirea față de acest proiect este că aplicația noastră pentru telefoane inteligente realizează recunoașterea în timp real. Ne permite să alegem un grup, cu număr variabil de utilizatori, dintr-o bază de date, pentru care se calculează vectorii de caracteristici. Aceste calcule sunt executate de către un serviciu Web, implementat în limbajul de programare Java.

METODA EIGENFACE

Metoda Eigenface pentru recunoașterea feței, este o metodă simplă și foarte practică. Conceptul de bază al metodei este reducerea informațiilor și, alegerea caracteristicilor semnificative. Chiar și atunci când prelucrăm o imagine mică obținem o cantitate mare de informații [4].

Ca date de intrare avem setul de învățare și setul de testare. Fiecare set de date conține imagini cu fețe, setul de învățare conține imagini etichetate cu numele persoanei căreia îi aparține fotografia, iar setul de testare conține imagini neetichetate. Scopul este clasificarea fiecărei persoane din setul de testare și determinarea apartenenței sau nu, la grupul de învățare.

Având în vedere ipoteza că fizionomiile umane sunt similare, putem să presupunem că orice imagine a unei fețe poate fi codificată ca o combinație a caracteristicilor semnificative. Fiecare caracteristică captează o "direcție" a variabilității fețelor [5]. Ideea, pe care se bazează

algoritmul propus de către Turk și Pentland este de a extrage din imaginea feței numai informațiile, caracteristicile relevante. Se codifică aceste informații cât se poate de eficient, după care se compară rezultatul cu modelele dintr-o bază de date, care au fost codificate similar.

În termeni matematici: găsim componentele principale ale feței sau vectorii proprii ai matricei de covarianță, generată din setul de imagini. Eigenface este un set de caracteristici care redă diferența dintre imaginile feței. Fiecare imagine din setul de învățare poate fi reprezentată ca o combinație liniară a secvențelor eigenface.

Baza de date conține imagini de dimensiune 150×200 pixeli. Fiecare astfel de matrice, presupusă de dimensiune 150×200 pixeli, este mai întâi transformată într-un vector, secvență de aceeași lungime, prin concatenarea coloanelor corespunzătoare. Algoritmul de procesare presupune parcurgerea următorilor pași:

1. Inițializare: pe baza imaginilor din setul de învățare calculăm secvențele eigenface, care vor defini spațiul feței;
2. Preprocesarea imaginii de intrare, care urmează să fie recunoscută;
3. Calcularea ponderilor pentru secvențele eigenface;
4. Proiectarea imaginii de intrare în spațiul feței;
5. Calcularea distanței euclidiene, alegerea distanței minime.

Obținerea vectorilor și valorilor proprii

În baza de date avem setul de învățare :

$$\Gamma_1, \Gamma_2, \dots, \Gamma_M$$

Unde Γ_n este o imagine a unei fețe, reprezentată printr-un vector coloană cu o dimensiune de (30000×1) .

Se calculează valoarea medie a imaginilor care formează setul de antrenare:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

Unde M este numărul imaginilor din setul de antrenare, iar rezultatul obținut va fi fața medie (Figura 1).

Pentru fiecare imagine din baza de date calculăm diferența față de fața medie :

$$\Phi_n = \Gamma_n - \Psi$$

Din aceste fețe construim matricea diferență cu o dimensiune de $(30000 \times M)$:

$$A = [\Phi_1, \Phi_2, \dots, \Phi_M]$$

Matricea de covarianță este :

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T$$



Figura 1. Fața medie

și are dimensiunea (30000×30000) .

Următorul pas este să calculăm valorile și vectorii proprii ai matricei de covarianță. În total sunt 30000 de perechi de vectori μ_n , și valori proprii λ_n .

Pentru fezabilitate computațională nu este nevoie să calculăm toate perechile de vectori și valori proprii, este destul dacă obținem $M-1$ perechi.

Fie L o matrice auxiliară, obținută în felul următor :

$$L = A^T A$$

Matricea L are o dimensiune de $(M \times M)$, care este o dimensiune mult mai mică decât dimensiunea matricei de covarianță $(M \ll 30000)$.

Prin înmulțirea cu matricea A, calculăm perechile de vectori și valori proprii (v_n, λ_n) ale matricei L.

$$A^T A v_n = \lambda_n v_n \Rightarrow$$

$$A A^T A v_n = \lambda_n A v_n$$

De aici ajungem la concluzia că $A v_n$ este vectorul propriu, iar λ_n este valoarea propriu al lui C :

$$C = A A^T$$

$$\mu_n = \sum_{k=1}^M v_{nk} \Phi_k = A v_n$$

Valorile proprii asociate ne permit să clasificăm vectorii proprii în funcție de utilitatea lor în caracterizarea variației între imagini.

Identificare cu Eigenface

Imaginile din setul de antrenare sunt proiectate în spațiul feței printr-o simplă operație :

$$\omega_{ik} = \mu_k^T (\Gamma_i - \Psi)$$

Unde $i = 1, 2, 3, \dots, M$ reprezintă persoana i din baza de date, iar μ_k reprezintă vectorul propriu.

Vectorul pattern al persoanei i fiind :

$$\Omega_i = [\omega_{i1}, \omega_{i2}, \dots, \omega_{iM'}]$$

Unde M' este de fapt numărul perechilor proprii alese.

Dacă o persoană are mai multe imagini, se calculează media vectorilor pattern al acestei persoane. Clasificarea persoanelor se face pe baza acestor vectori pattern.

Având o imagine de intrare, calculăm vectorul pattern al acestuia:

$$\Omega = [\omega_1, \omega_2, \dots, \omega_{M'}]$$

Următorul pas este să identificăm dacă persoana din imaginea de intrare face parte sau nu din grupul de utilizatori ales. Pentru asta avem nevoie de un prag [6]. Selectarea pragului optim pentru clasificare este foarte important pentru performanțele de recunoaștere a sistemului. Fiecare grup de utilizator ales necesită o valoare de prag unică. Mai des sunt folosite metode ad hoc pentru selecția pragului, dar cele mai multe metode sunt de testare manuală și avem nevoie și de validarea pragului selectat.

Trebuie să calculăm distanța euclideană dintre vectorul pattern al imaginii de intrare și al imaginii din baza de date.

$$\varepsilon_k^2 = \|(\Omega - \Omega_k)^2\|$$

Următorul pas este alegerea distanței minime:

$$\min\{\varepsilon_k, k = 1, 2, \dots, M'\} = \varepsilon_k$$

Dacă această distanță este mai mică decât pragul predefinit (θ_p) atunci persoana din imaginea de intrare va fi recunoscută ca și persoana k , altfel aplicația identifică ca necunoscut.

PREPROCESAREA IMAGINII

Preprocesarea este un pas important în recunoașterea feței. Ca să îmbunătățim performanța de recunoaștere trebuie să urmărim pașii următori:

1. După detectarea feței decupăm fața și aplicăm o scalare a imaginii ca să ajungem la dimensiunea de 150×300 .
2. Convertim imaginea într-o imagine greyscale, și între timp calculăm histograma.

```
for i = 0; i < faceImage.Width; i++
    for j = 0; j < faceImage.Height; j++
        clr = faceImage.getRGB(i, j);
        g = (clr.getR() + clr.getG() + clr.getB())/3;
        faceImage.setRGB(i, j, g);
        hist[g]++;
    end for;
end for;
```

3. Egalizăm histograma imaginii, mai întâi calculând histograma cumulativă.

```
for (j = 1; j < hist.length; j++)
    hist[j] = hist[j - 1] + hist[j];
for (v = 0; v < h; v++)
    for (u = 0; u < w; u++)
        clr = faceImage.getRGB(u, v);
```

```
g = (hist[clr.getB()] * (255)) / (h*w);
```

```
faceImage.setRGB(u, v, g);
```

```
end for;
```

```
end for;
```

4. Eliminarea fundalului prin aplicarea unei măști ovale.

```
for (i = 0; i < h; i++)
```

```
    for (j = 0; j < w; j++)
```

```
        c1 = ((2/h) * i - 1);
```

```
        c2 = ((2/w) * j - 1);
```

```
        if (c1 * c1 + c2 * c2 > 1)
```

```
            faceImage.setRGB(j, i, 0);
```

```
        end if;
```

```
    end for;
```

```
end for;
```



Figura 2. Imagini după preprocesare

LOCALIZAREA ȘI DETECTAREA FEȚEI

Detectarea feței se face în aplicația Android, care localizează fețele pe baza ochilor. Ca rezultat returnează distanța dintre ochi și punctul de mijloc dintre ochi.

SCENARIUL UTILIZĂRII APLICAȚIEI

Înainte să lansăm aplicația Android trebuie să pornim serverul care găzduiește serviciul Web. Pentru accesarea metodelor din serviciul Web trebuie să ne asigurăm că avem internetul pornit pe telefon. Aplicația oferă opțiunea de alegere a unui grup de utilizatori, sau chiar pe toți din baza de date (Figura 3.A.). Următorul pas este să facem o poză. Când camera telefonului este pornită apare un dreptunghi verde care arată unde să fie fața utilizatorului (Figura 3.B.).

Recunoașterea se face pe baza unui prag calculat, după numărul utilizatorilor din grupul ales. Pe baza acestui prag se decide dacă poza aparține unei persoane din grup sau nu. Dacă distanța calculată depășește pragul, persoana va fi necunoscută.

REZULTATE EXPERIMENTALE

Identificarea unei persoane se face în felul următor: calculăm distanța euclideană dintre coordonatele imaginii de test și coordonatele imaginilor din setul de antrenare.

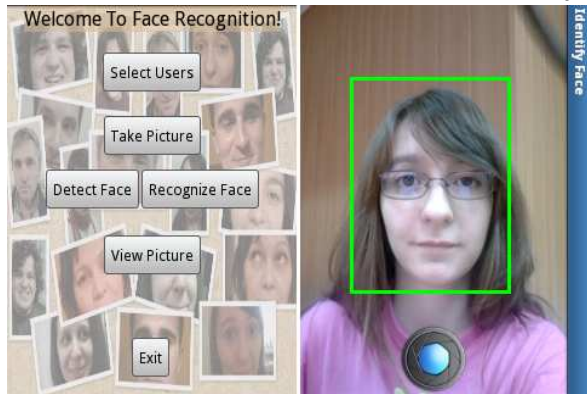


Figura 3. A. Interfața aplicației B. Camera când facem poza

Alegem distanța minimă, care dacă este mai mică decât un prag prestabilit atunci persoana este recunoscută, dacă depășește pragul este considerată necunoscută.

Astfel după rezultat avem patru cazuri [1]:

1. Respingere falsă (RF): în cazul în care fața este asociată cu fața corectă, iar distanța este mai mare decât pragul.
2. Acceptare falsă (AF): în cazul în care fața este asociată cu o față greșită, iar distanța este mai mică decât pragul.
3. Respingere corectă (RC): dacă fața este asociată cu fața greșită, iar distanța este mai mare decât pragul.
4. Acceptare corectă (AC): dacă fața este asociată cu fața corectă, iar distanța este mai mică decât pragul.

Testul s-a făcut pentru 25 de imagini de fețe diferite ale aceluiași utilizator cu id = 5 (Tabelul 1).

Tabelul 1. Rezultate experimentale

Idul utilizatorilor aleși	Nr. Util.	Prag /10 ⁶	RF [%]	AF [%]	RC [%]	AC [%]
1, 2, 3, 4, 5, 6, 7, 8, 9	9	6,55	16,00	8,00	8,00	68,00
1, 2, 3, 4, 5, 6, 7, 8	8	6,55	28,00	8,00	12,00	52,00
1, 2, 3, 4, 5, 6, 7, 9	8	6,55	24,00	0,00	20,00	56,00
1, 2, 4, 5, 6, 7, 8, 9	8	6,55	16,00	4,00	16,00	64,00
1, 2, 3, 4, 5, 7, 8, 9	8	6,55	24,00	12,0	12,00	52,00
2, 3, 4, 5, 6, 7, 8, 9	8	6,55	24,00	8,00	20,00	48,00
3, 4, 5, 6, 7, 8, 9	7	6,55	28,00	4,00	28,00	40,00
1, 2, 5, 6, 7, 8, 9	7	6,55	24,00	0,00	32,00	44,00
1, 2, 3, 4, 5, 8, 9	7	6,55	40,00	8,00	4,00	48,00

2, 3, 4, 5, 7, 9	6	6,55	48,00	0,00	12,00	40,00
1, 2, 3, 4, 5, 6	6	6,55	32,00	4,00	20,00	44,00
1, 5, 6, 7, 8, 9	6	6,55	36,00	0,00	20,00	44,00
1, 2, 3, 4, 5	5	6,55	40,00	0,00	12,00	48,00
3, 4, 5, 8, 9	5	6,55	44,00	4,00	8,00	44,00

CONCLUZII

În această lucrare am implementat metoda Eigenface, fiind o metodă des utilizată datorită simplității și eficienței sale. Pentru a realiza implementarea unui astfel de sistem pentru telefoane mobile, trebuia să facem un compromis datorită limitărilor capacităților hardware, de aceea am ales poze de dimensiune mică (150×200).

Din experimentele făcute rezultă că pragul trebuie determinat în funcție de numărul utilizatorilor din setul de învățare. Dacă avem un număr mai mic de utilizatori e bine să avem un prag mai mare. Dacă a scăzut numărul utilizatorilor a crescut rata respingerilor false, ceea ce înseamnă că persoana a fost recunoscută, dar din cauza pragului a fost respinsă. Din experimente rezultă și faptul că dacă avem mai multe imagini ale aceluiași persoane, crește performanța de recunoaștere.

Experimentele vor continua prin alegerea unor seturi de teste complexe și variate, și prin dezvoltarea unui algoritm, pe baza rezultatelor obținute, pentru determinarea adaptivă a pragului.

REFERINȚE

- [1]. Guillaume Dave, Xing Chao, Kishore Sriadibhatla, „Face Recognition in Mobile Phones”. Department of Electrical Engineering Stanford University Stanford, USA, http://www.stanford.edu/class/ee368/Project_10/Reports/Sriadibhatla_Davo_Chao_FaceRecognition.pdf
- [2]. Zhifeng Li, Xiaou Tang, „Eigenface Recognition Using Different Training Data Sets”. Information Conference on Information Security, Shanghai, 2002, http://www.scourge.fr/mathdesc/documents/facerecog/igen_face_.pdf
- [3]. Matthew Turk, Alex Pentland, „Eigenfaces for Recognition”. J. of Cognitive Neuroscience, vol. 3, no. 1, 1991, <http://dl.acm.org/citation.cfm?id=1326894>
- [4]. Parvinder Sandhu, Iqbaldeep Kaur, Amit Verma, Samriti Jindal, Inderpreet Kaur, Shilpi Kumari, „Face Recognition Using Eigenface Coefficients and Principal Component Analysis”. International Journal of Electronics and Electrical Engineering, Vol 3/8, 2009, pp. 498-502, <http://www.waset.org/journals/ijee/v3/v3-8-77.pdf>
- [5]. Carmen Au, Jean-Sebastien Legare, „Face Recognition: Robustness of the ‘Eigenface’ Approach”, <http://www.cs.mcgill.ca/~rshaik1/papers/FaceRec.pdf>
- [6]. Ronny Tjahyadi, Wanquan Liu, Svetha Venkatesh, „Automatic Parameter Selection for Eigenfaces”, http://www.computing.edu.au/~svetha/papers/papers2004/ronny/Tjahyadi_ICOTA2004.pdf