

Dezvoltarea unei soluții de eye tracking

Afloroaei Daniel

Mărț, Eduard

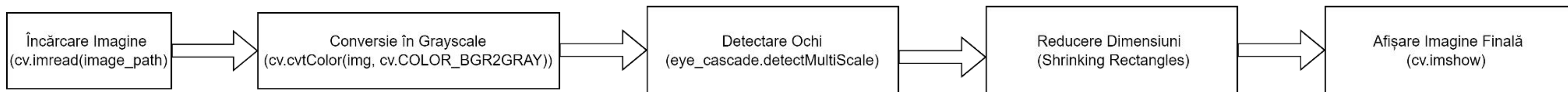
Dezvoltarea unei soluții de eye tracking

1. Context și Motivație

- **Context:** Proiectul dezvoltă o soluție de eye-tracking, esențială în asistența persoanelor cu dizabilități, marketing, analiza comportamentului utilizatorilor și interacțiunea om-mașină. Aceasta oferă informații valoroase despre modul în care oamenii interacționează cu dispozitive și medii virtuale.
- **Motivație:** Tehnologia eye-tracking îmbunătățește accesibilitatea și eficiența interfețelor, aducând beneficii în educație, sănătate și cercetare de piață. Automatizarea analizei privirii transformă interacțiunea utilizatorilor cu tehnologia, reducând barierele pentru persoanele cu dizabilități.
- **Obiectivul proiectului:** Scopul acestui proiect este să dezvolte și să testeze o soluție de eye-tracking care poate fi integrată în dispozitive moderne, oferind o performanță ridicată și o ușurință în utilizare.

Dezvoltarea unei soluții de eye tracking

2. Arhitectura preliminară a soluției



Încărcare Imagine: Utilizarea funcției `cv.imread(image_path)` pentru a încărca imaginea de la o cale specificată. Această funcție citește imaginea și o stochează într-o matrice de pixeli.

Filtre Aplicate: Conversie în Grayscale: Utilizarea funcției `cv.cvtColor(img, cv.COLOR_BGR2GRAY)` pentru a transforma imaginea color într-o imagine alb-negru. Acest pas simplifică procesarea ulterioară, deoarece elimină informațiile redundante de culoare și se concentrează pe lumină și contrast.

Detectare Ochi: Utilizarea modelului Haar Cascade pentru identificarea ochilor în imagine.

Reducere Dimensiuni: Ajustarea dimensiunilor dreptunghiurilor de detecție pentru o evidențiere mai precisă.

Afișare Imagine Finală: Imaginea cu ochii evidențiați este afișată utilizând `cv.imshow`, permițând utilizatorului să vadă rezultatele detecției.

Dezvoltarea unei soluții de eye tracking

Funcții folosite:

1. **`cv.CascadeClassifier`**: creează un obiect clasificator în cascadă, utilizat pentru detectarea obiectelor în imagini, în acest caz ochii.
2. **`cv.imread(image_path)`**: încarcă imaginea de la calea specificată. Returnează imaginea sub forma unui obiect matriceal (`numpy.ndarray`). Dacă imaginea nu poate fi încărcată, returnează `None`.
3. **`cv.cvtColor(img, cv.COLOR_BGR2GRAY)`**: convertește imaginea color din formatul BGR (al OpenCV) în tonuri de gri.
4. **`eye_cascade.detectMultiScale()`**: detectează obiectele (ochii, în acest caz) dintr-o imagine utilizând clasificatorul în cascadă.
5. **`cv.rectangle()`**: desenează un dreptunghi pe imagine.
6. **`cv.imshow()`**: afișează o fereastră cu o imagine.
7. **`cv.destroyAllWindows()`**: închide toate ferestrele deschise de OpenCV.

Dezvoltarea unei soluții de eye tracking

3. Evaluarea Preliminară a Soluției

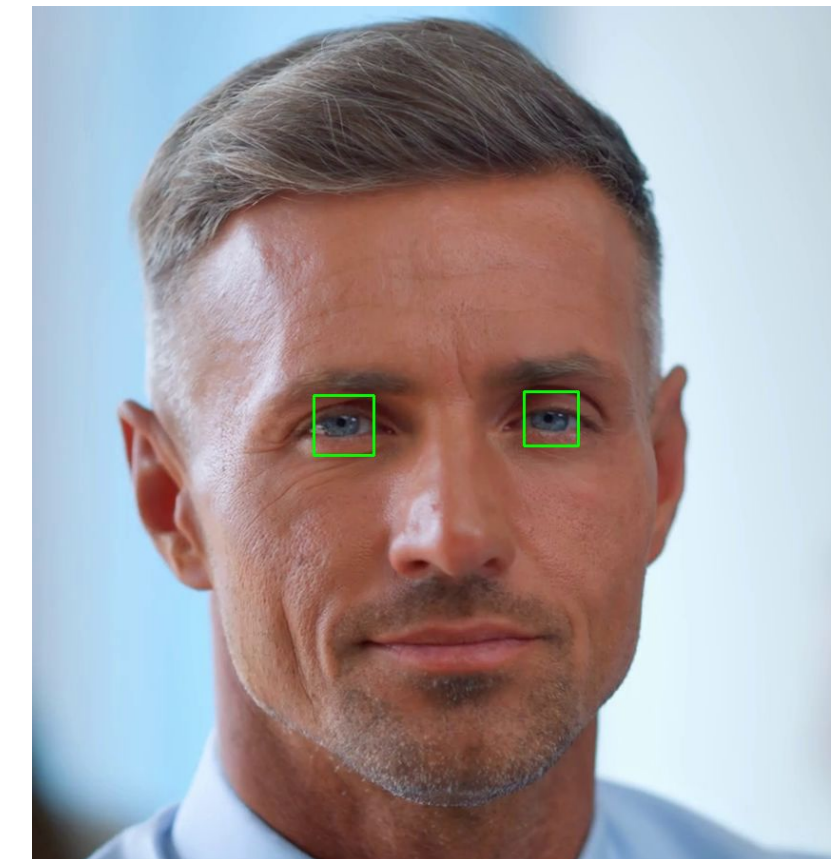
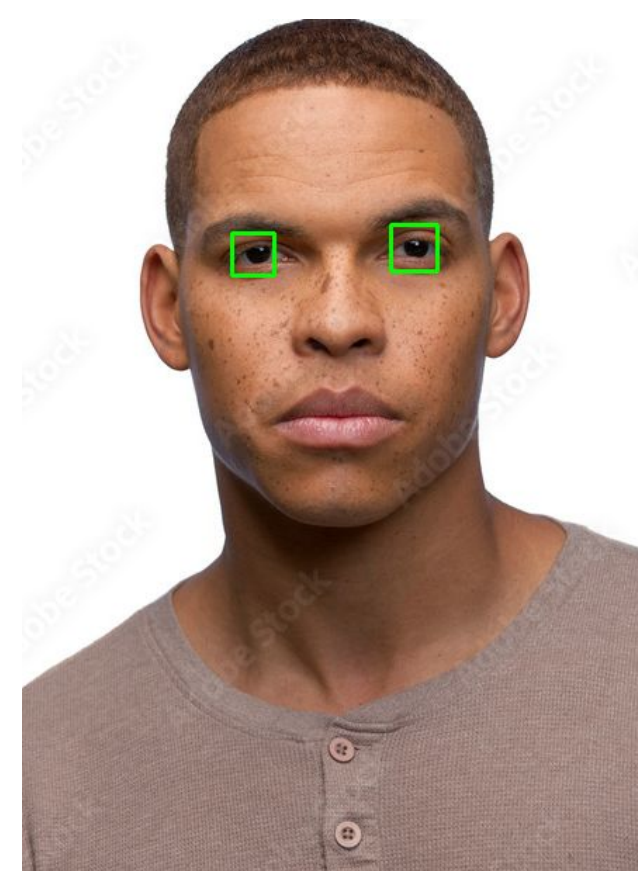
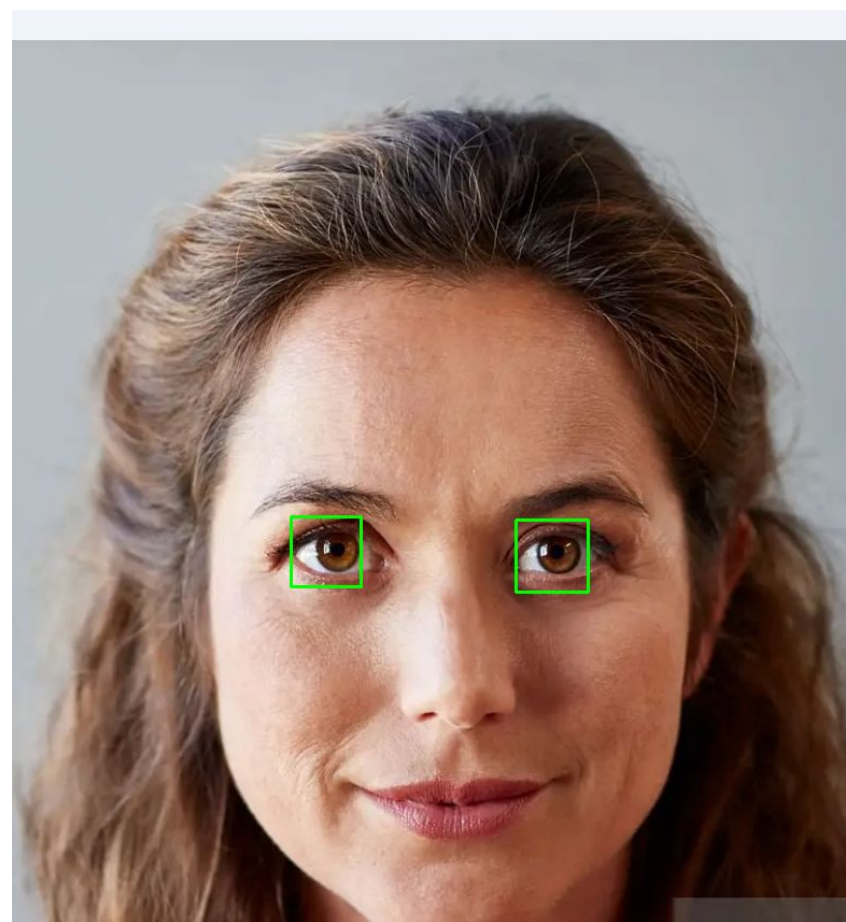
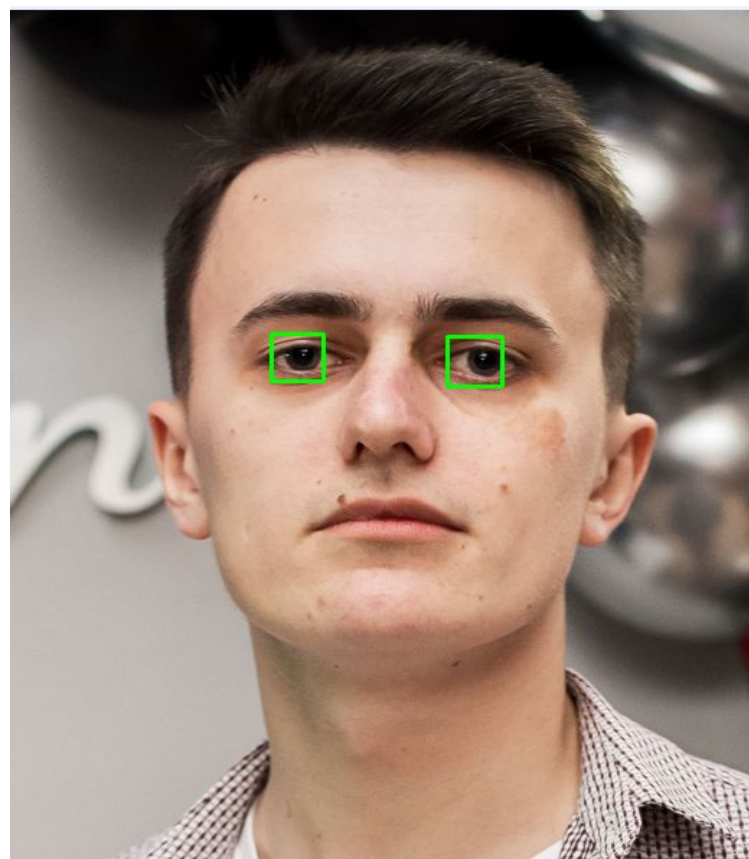
- **Metodologia de evaluare:** Am testat soluția pe mai multe imagini cu fețe, cu accent pe verificarea corectitudinii și preciziei pătratelor detectate. Rezultatul a fost vizualizat pentru a observa dacă ochii sunt detectați corect și evidențiați corespunzător.
- **Setul de date:** Pentru evaluare am folosit imagini proprii și de pe internet. Aceste imagini au fost alese pentru a verifica adaptabilitatea algoritmului în condiții diverse.
- **Exemple de imagini folosite:**



Dezvoltarea unei soluții de eye tracking

4. Rezultate Preliminare

- **Rezultate obținute:** Timpul de execuție este între 150 și 400 ms, dar acesta poate crește, în funcție de scaleFactor, minNeighbors și dimensiunea imaginii. Performanța scade drastic pentru fețele parțial obstructionate, imaginile cu unghiuri neobișnuite sau imagini mai mici.
- **Interpretarea rezultatelor:** Rezultatele sunt destul de precise, dar pot apărea false pozitive, în funcție de imagine.
- **Vizualizări:**



Dezvoltarea unei soluții de eye tracking

Matricea de confuzie

	Predicție: Ochi Detectați	Predicție: Ochi Nedetectați	Total
Există ochi	True Positives (TP) = 8	False Negatives (FN) = 2	10
Nu există ochi	False Positives (FP) = 1	True Negatives (TN) = 9	10

True Positives (TP): Numărul de cazuri în care ochii au fost corect detectați.

False Positives (FP): Numărul de cazuri în care algoritmul a detectat ochi acolo unde nu există.

True Negatives (TN): Numărul de cazuri în care algoritmul a corect nedetectat ochi acolo unde nu există

False Negatives (FN): Numărul de cazuri în care algoritmul nu a detectat ochii care există în imagine.

Precizie: 88,89%

Rată de Adevăr: 80%

Acuratețea: 85%

5. Concluzii Preliminare

- **Rezumatul progresului:** Am realizat algoritmul pentru detectarea ochilor, având ca parametru imaginea, iar rezultatul este imaginea cu ochii detectați de algoritm.
-
- **Limitările soluției actuale:** Algoritmul este limitat la imagini în care persoana se uită direct la cameră și rezoluția imaginilor trebuie să fie înaltă.
-
- **Potențiale îmbunătățiri:** Creșterea preciziei pentru imaginile în care persoana nu se uită direct la cameră, la diferite luminozități.

Dezvoltarea unei soluții de eye tracking

6. Direcții Viitoare

1. Pași următori:

- Revizuirea codului, astfel încât să detecteze în timp real ochii la rularea programului.
- Îmbunătățirea localizării pupilei
- Optimizarea resurselor
- Finalizarea documentației

Obiectivele Finale:

Realizarea detecției ochilor în timp real, având o precizie cât mai bună.