

# Controlul mouse-ului bazat pe mișcările irisului

## Proiect realizat de:

- Murariu-Tănăsache Iulian – 1311A
- Enache Ștefan – 1311A

## Rezumat:

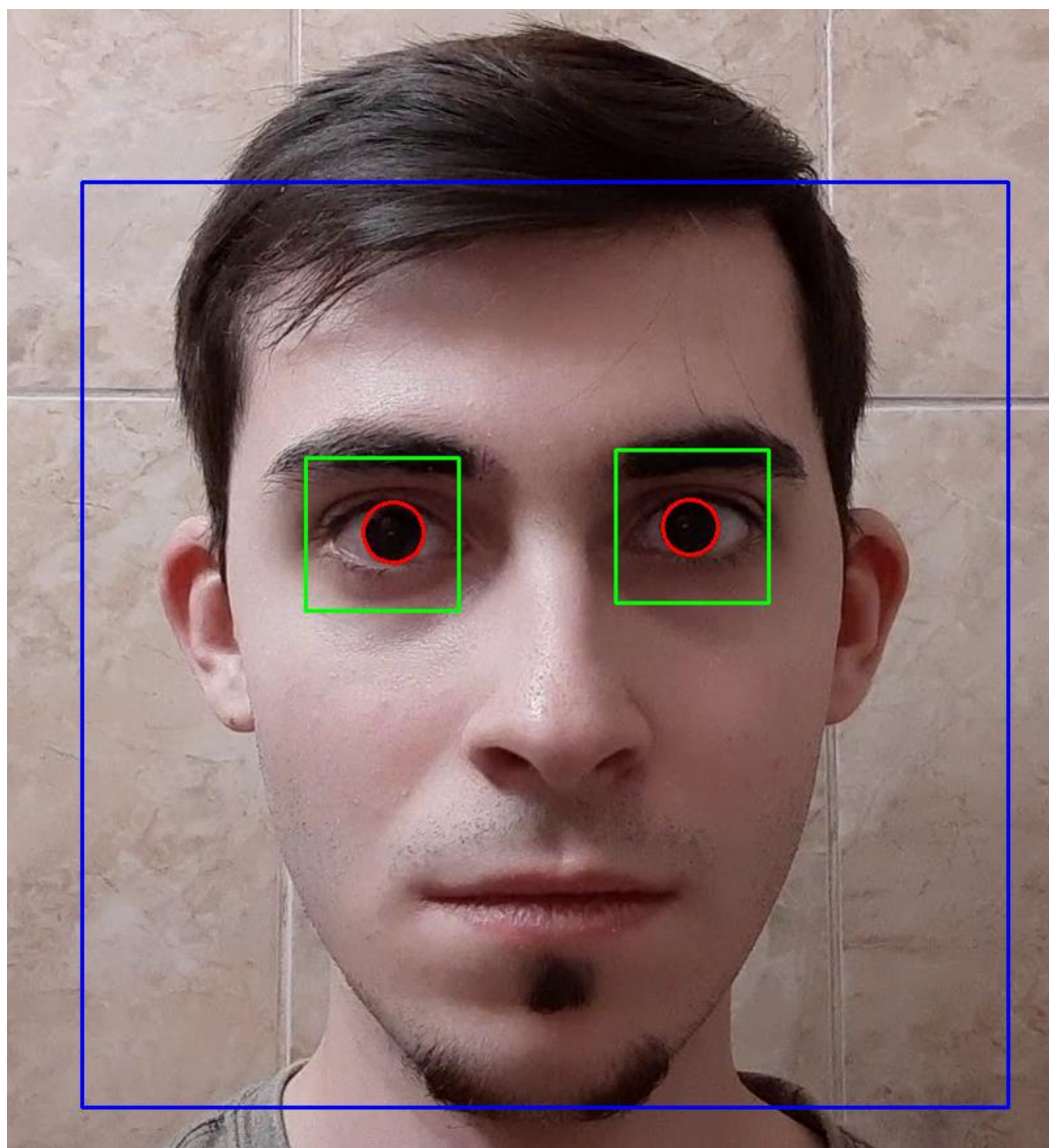
Proiectul nostru constă în detecția irisului unei persoane care se uită la webcam și urmărirea acestuia. După o perioadă de calibrare, utilizatorul va putea controla mouse-ul privind în diferite direcții sau va putea să acționeze click dreapta sau stânga prin închiderea ochiului respectiv pentru câteva secunde.

Feed-ul camerei va fi preluat de aplicația noastră și pe fiecare cadru, utilizând algoritmul Haar cascade, vom detecta fața persoanei din imagine. Apoi vom detecta locația ochilor, iar după, utilizând tehnici precum binarizarea imaginii, operații morfologice și transformări Hough, vom localiza irisul. Acest algoritm de detecție se va aplica succesiv pe fiecare cadru al camerei web pentru a urmări irisul și mișcarea acestuia. Inițial va fi o perioadă de calibrare pentru a determina poziția irisului când utilizatorul se uită drept în față. Calibrarea va reîncepe când se pierde urma irisului pentru mai multe cadre sau când utilizatorul dorește, apăsând tasta *R*. Se va face o medie a primelor poziții obținute pentru ca mai apoi aceasta să fie folosită pentru a corectat eventuale detecții greșite și pentru a fi folosită ca punct de referință pentru a determina direcția de mișcare a irisului. Utilizatorul va porni sau opri controlul mouse-ului dacă închide ambii ochi pentru câteva secunde. Atunci când poate controla mouse-ul, acesta se va mișca constant în direcția de mișcare a irisului, iar click stânga și dreapta vor putea fi acționați prin închiderea ochiului stâng sau drept.

Scopul proiectului este de a localiza irisul oricărei persoane din fața unei camere și de a putea controla mouse-ul prin diferite acțiuni, precum mișcarea irisului sau închiderea ochilor.

Obiectivele principale ale proiectului sunt de a localiza în mod corect fața, ochii și irisul unei persoane dintr-un cadru. De asemenea, de a recunoaște diferitele acțiuni ale ochiului, precum mișcarea în diferite direcții sau închiderea lui, prin care să fie controlat mouse-ul corespunzător.

La momentul actual, programul nostru poate în teorie să efectueze toate operațiile descrise, însă detecția ochiului și a irisului nu oferă rezultate constante, fapt ce face ca acțiunile mouse-ului să fie imprevizibile, iar modul în care sunt acționate butoanelor mouse-ului ar putea să fie îmbunătățit.



Rezultat dorit

### Introducere:

Am venit cu această idee de proiect deoarece eye tracker-ele, pecum cel de la Tobii, au devenit celebre în ultimul timp și acest lucru ne-a inspirat să încercăm și noi să urmărim ochiul, folosind diferite metode de prelucrare a imaginii, pentru a fi în temă cu disciplina. Acest lucru s-a dovedit însă mai greu decât ne-am așteptat, dar am încercat să obținem rezultate cât mai precise fără să apelăm excesiv la inteligență artificială sau la aparatură performantă. Din păcate, rezultatele nu sunt cele mai bune, astfel că momentan aplicația nu se dovedește a fi practică, dar măcar am reușit să înțelegem mai multe despre domeniul prelucrărilor de imagini prin acest proiect.

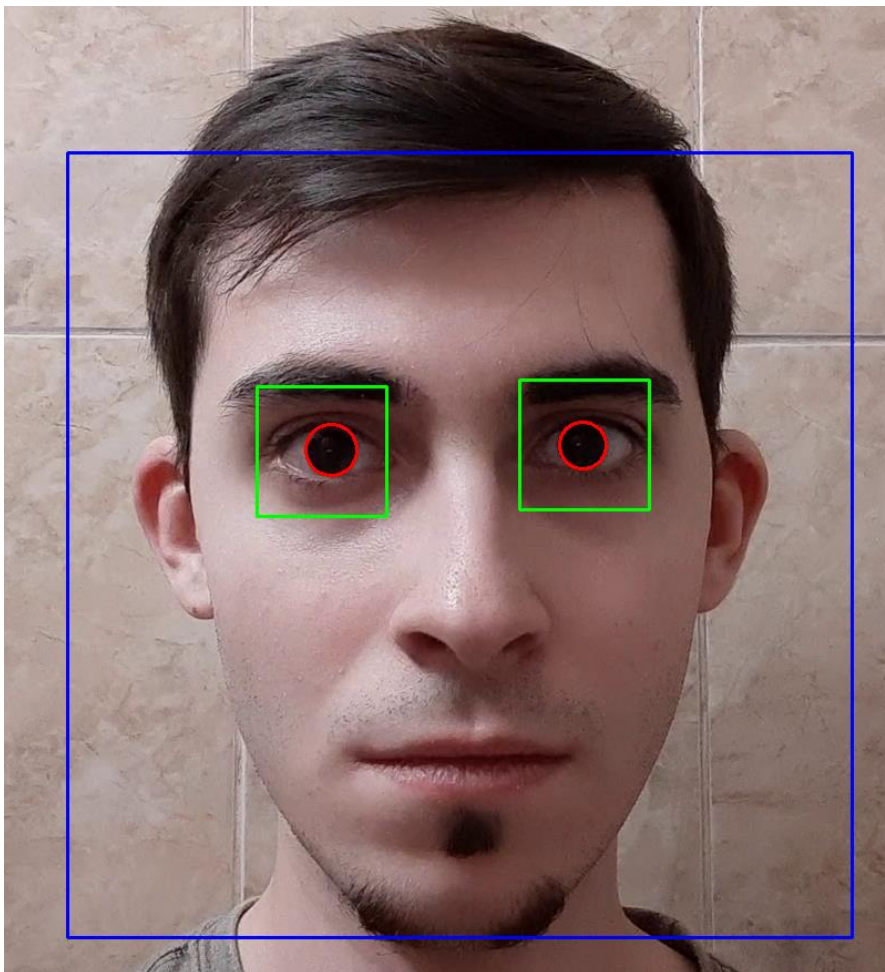
### Metode existente:

Idea de a urmări privirea unei persoane și de a folosi mișcările irisului a fost o problemă abordată de mult timp. Încă de la finalul secolului 19 au fost concepute de “eye tracker” dezvoltate mecanic pe atunci. O astfel de idee presupunea o lentilă cu o gaură conectată cu un băț de aluminiu care se mișca odată cu mișcarea ochiului.

Odată cu dezvoltarea tehnologiei, și mai ales a calculatorului, și detecția mișcării ochiului s-a îmbunătățit considerabil. Astăzi, sunt diferite metode de a observa mișcarea unei persoane. Foarte multe dintre acestea se folosesc de inteligență artificială și rețele neuronale. Printre cele mai celebre aparate care realizează acest lucru este eye tracker-ul de la Tobii care se folosește de una sau mai multe camere speciale și machine learning pentru a mapa ochii utilizatorului pe ecranul acestora.

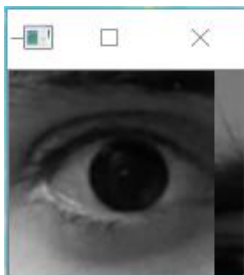
### Metoda implementată:

Metoda implementată de noi folosește machine learning(Haar Cascade) pentru a identifica fața și ochii utilizatorului. Acesta este un clasificator care este antrenat pentru a identifica diverse lucruri, în cazul nostru repere faciale și ochi. După care urmează prelucrarea zonelor de interes, cele care au ochii pentru a identifica irisul.



Chenarul albastru – fața detectată; Chenarele verzi – ochii detectați; Cercurile roșii – irisul detectat de HoughCircles

Prima oară, aplicăm un blur gaussian pe zona de interes și după obținem o valoare optimă de binarizare cu ajutorul algoritmului lui Otsu. Această valoare este necesară pentru a obține contururile ochiului, lucru care se realizează prin algoritmul Canny.

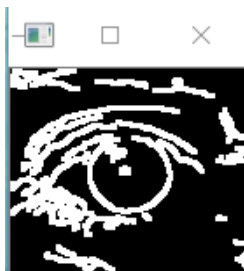


Blur pe zona ochiului

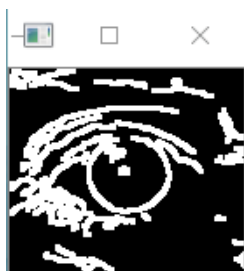


Muchiile obținute după aplicarea algoritmului Canny

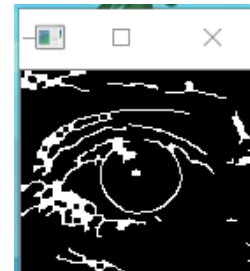
Pentru a mai tria din muchii și a le face mai clare și distincte, aplicăm operații morfologice, care sunt în ordine: dilatare(pentru a umple multe goluri lasate de denivelările ochiului), deschidere(pentru a scăpa de pete și puncte nedorite), eroziune(pentru a subția muchiile și a putea diferenția clar între conturul irisului și alte muchii).



Dilatare



Deschidere



Eroziune

Acum urmează detecția propriu-zisă a irisului, folosind HoughCircles pentru a detecta cercuri. Unul din parametrii acestei funcții este cel care reprezintă “încrederea” în posibilitatea ca un pixel să reprezinte centrul unui cerc. Pentru acest parametru am creat o buclă de reglare, astfel că valoarea lui se modifică cât timp nu se detectează niciun cerc, dar poate și să realizeze și să iasă din buclă dacă chiar nu există niciun cerc în cadrul respectiv, de exemplu dacă persoana are ochii închiși.

Algoritmul HoughCircles poate totuși să returneze și mai multe cercuri sau cercuri eronate, așa că urmează o secțiune de detecție a erorilor și corectare. Inițial, este o fază de calibrare în care toate cercurile sunt acceptate ca fiind corecte, urmând a se face o medie între pozițiile acestora. Acest cerc este folosit și ca cercul din centru după care se va compara direcția irisului. După faza de calibrare fiecare cerc se compară cu cercul medie și este acceptat doar dacă se află într-o marjă de eroare față de cercul de referință. Dacă mai multe cercuri sunt acceptate, atunci se ia cercul cel mai apropiat de medie. Dacă niciun cerc nu este bun, atunci pentru acest cadru va fi folosit cercul medie. Media este

reactualizată cu fiecare cerc acceptat pentru a ține cont de mișcarea irisului. Tot aici sunt luate în calcul și alte probleme ce pot apărea, precum eșuarea prelungită de a detecta ochii sau irisul, fapt ce va duce la recalibrare, sau cazul în care ochii nu sunt detectați deoarece utilizatorul îi are închiși.

La final, se compară cercul obținut corect cu punctul central al ochiului pentru a obține direcția în care se uită persoana din fața camerei. Această direcție este folosită în a mișca mouse-ul constant. Dacă unul din ochi este detectat ca fiind închis, atunci se va apăsa butonul corespunzător pe mouse, iar dacă ambii sunt închiși, acest lucru oprește funcționalitatea de control a mouse-ului.

### **Rezultate experimentale:**

Rezultate obținute la acest moment nu sunt spectaculoase, deoarece mediul în care se află utilizatorul (caracteristici precum lumina sau fundalul) afectează considerabil algoritmul de detecție. Atât algoritmul Haar dar și HoughCircles oferă rezultate imprevizibile și inconstante.

### **Tehnologii utilizate:**

Pentru a dezvolta aplicația propusă, vom folosi limbajul de programare Python și IDE-ul PyCharm. Biblioteca principală pe care o vom utiliza va fi OpenCV, pe care o vom folosi pentru majoritatea funcțiilor de procesare de imagine. Pentru mișcarea mouse-ului vom folosi librăria PyAutoGUI. O bibliotecă ajutătoare pe care am folosit-o pentru calculul mediei de corecție este statistics.

### **Planificare și alocare a task-urilor:**

- Enache Ștefan:
  - Detecție față și ochi folosind Haar Cascade
  - Corectarea erorilor de detecție prin calculul medie
  - Pornirea/oprirea controlului mouse-ului prin închiderea ochilor
  - Acționare click stânga/dreapta prin închiderea ochiului corespunzător
- Murariu-Tănăsache Iulian:
  - Prelucrarea imaginii pentru detecția irisului
  - Ajustare dinamică a parametrilor pentru detecție
  - Calibrare poziție inițială a irisului
  - Determinare direcție de mișcare a irisului

### **Concluzii:**

În concluzie, deși programul nu funcționează la așteptările noastre, am aflat despre multe tehnici și algoritmi pentru a extrage informații din imagini. A fost o încercare admirabilă de a replica o idee interesantă care devine tot mai cercetată.

## Referințe:

- <https://www.tobii.com/group/about/this-is-eye-tracking/>
- <https://docs.opencv.org/>
- **Digital Image Processing - R.C.Gonzales, R.E.Woods**
- [https://docs.opencv.org/4.x/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html)
- [https://en.wikipedia.org/wiki/Eye\\_tracking](https://en.wikipedia.org/wiki/Eye_tracking)
- <https://pyautogui.readthedocs.io/en/latest/>