

Simularea funcționalităților mouse-ului folosind repere faciale

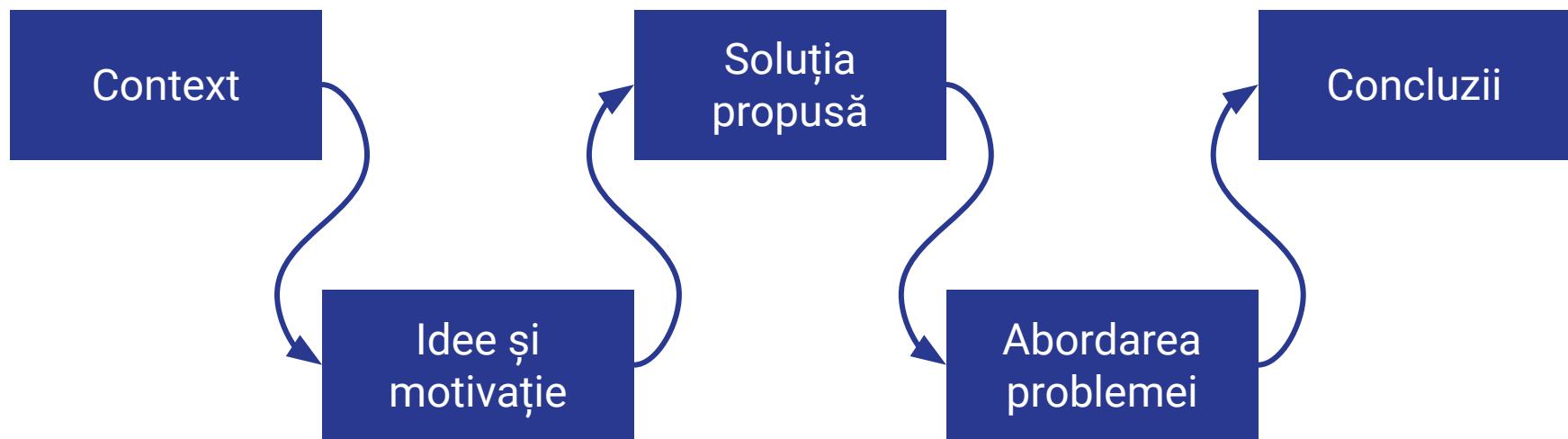
Lucrare de licență – iulie 2020

Student: Iacob Sergiu

Coordonator: Asist. Dr. Eugen Croitoru

Facultatea de Informatică, Universitatea “Alexandru Ioan Cuza” Iași

Cuprins

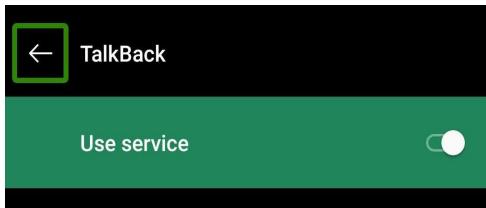


Context

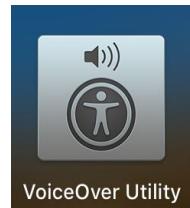
Accesabilitate, tehnologie asistivă



Asistenți digitali

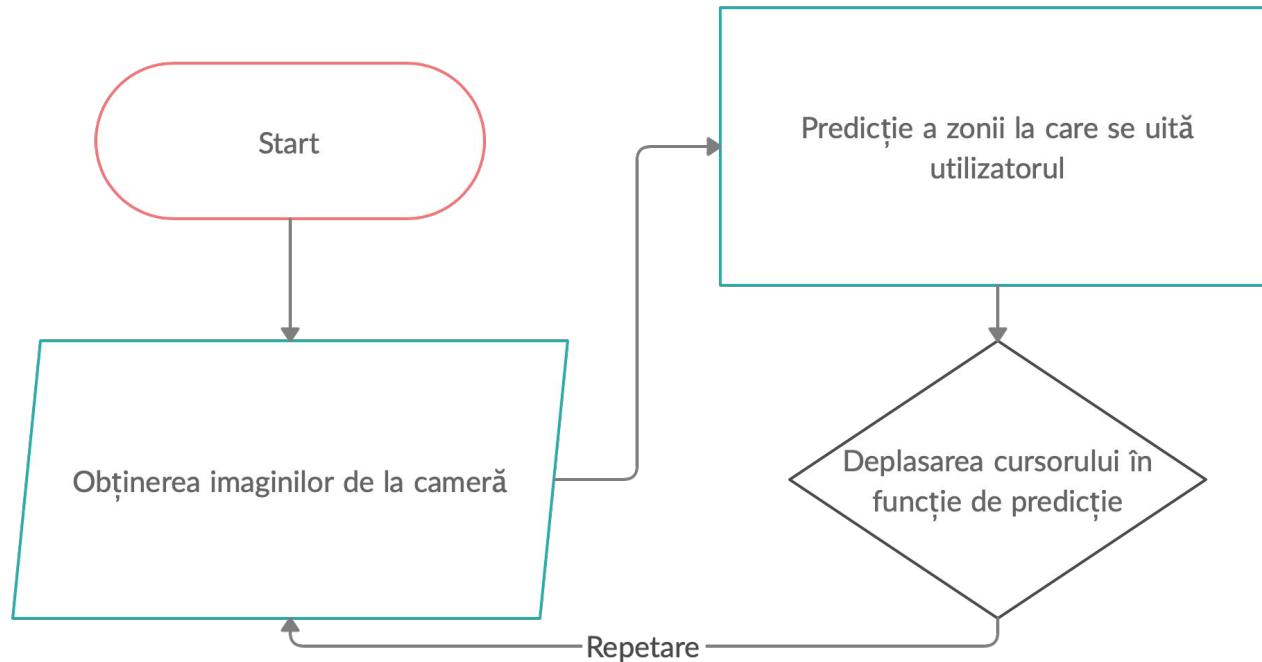


Citioare de ecran



Învățare
automată

Idee



Soluții deja existente

CameraMouse – Windows, mișcări ale capului



Videoclip preluat de pe [site-ul CameraMouse](#)

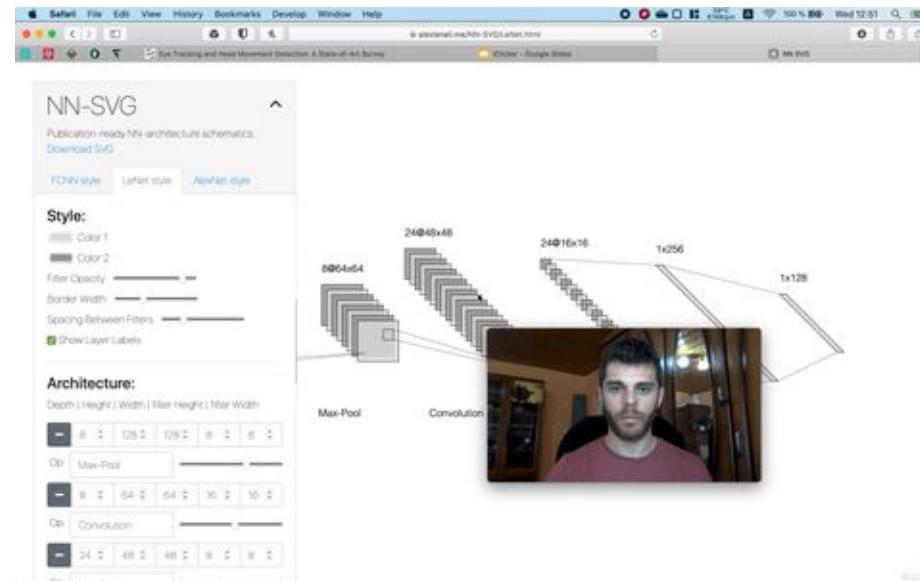
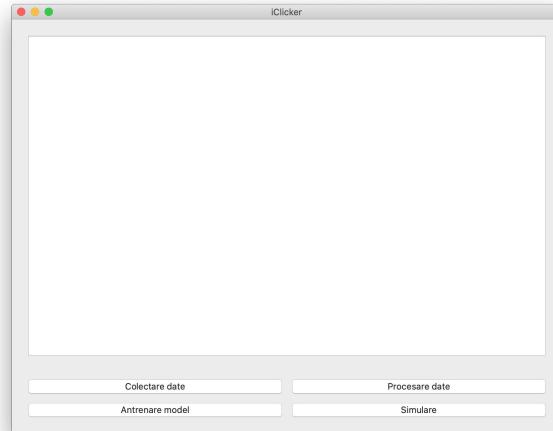
Tobii, IntelliGaze – necesită hardware adițional



Imagine preluată de pe [Digital Trends](#)

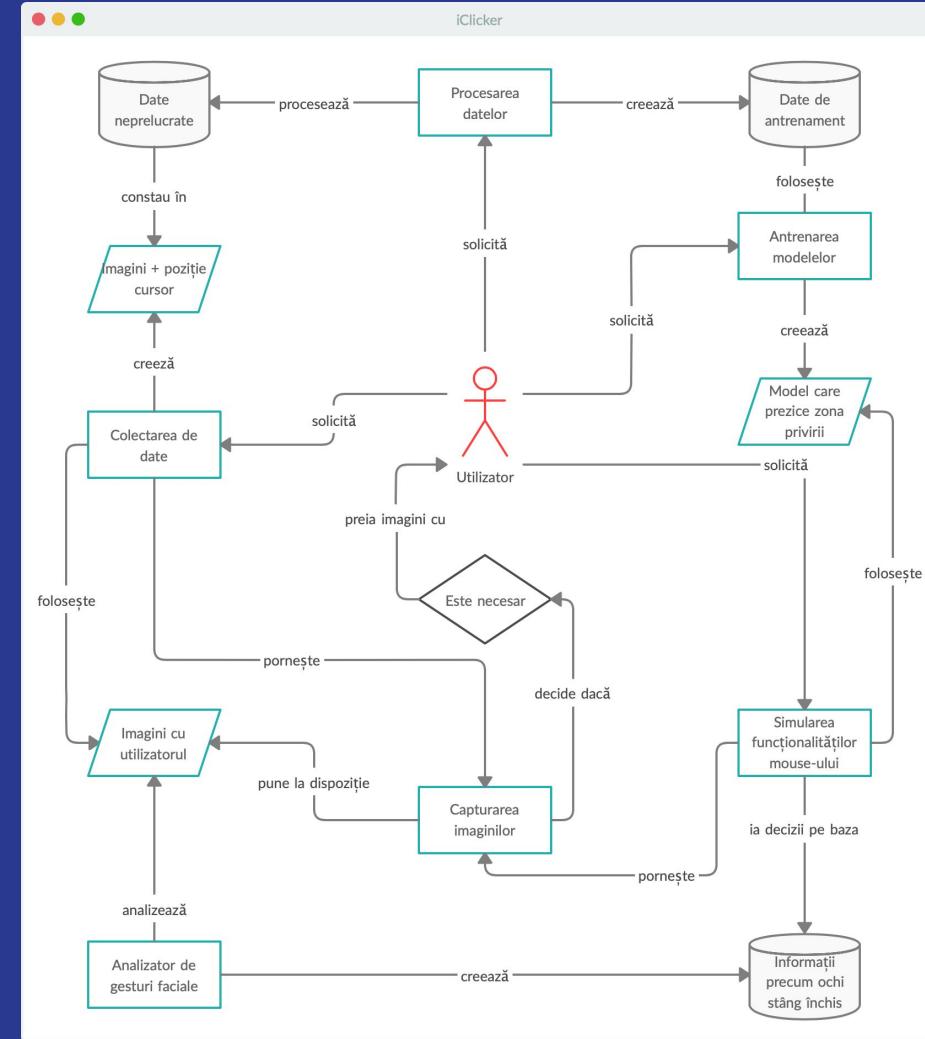
Soluția propusă

- Aplicație cross-platform
- Folosirea webcam-ului
- Gesturi faciale



iClicker

Organigramă



Abordarea problemei

Obținerea datelor
de antrenament

Procesarea
datelor

Antrenarea
unui model

Folosirea
modelului

Importanța datelor

Metoda pasivă

Metoda activă

Avantajele procesării

Extragerea ochilor

Extragerea feței

Formarea “bandei
oculare”

MLP, CNN

Metrica performanței

Versionarea modelelor

Parametrizare

Rezultate

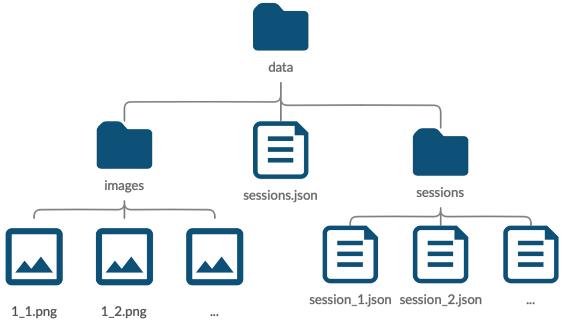
Urmărirea ochilor

Deplasarea cursorului

Repere faciale – click
stânga/dreapta

1	2	3
4	5	6
7	8	9

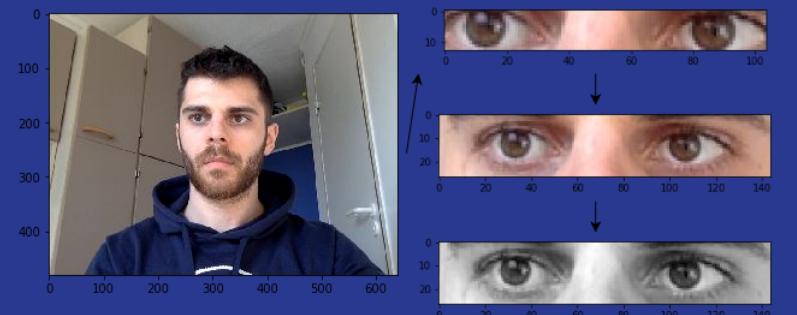
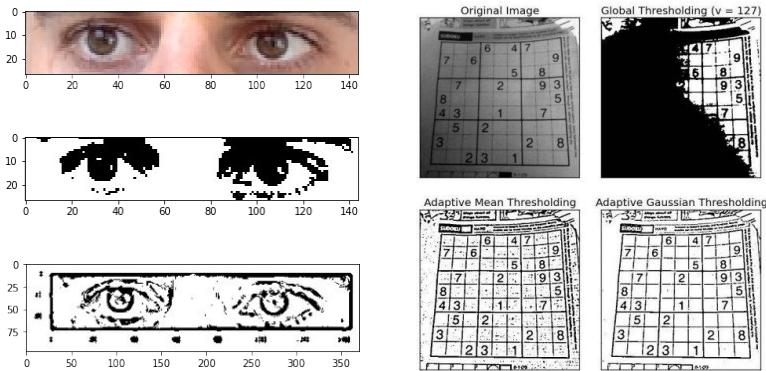
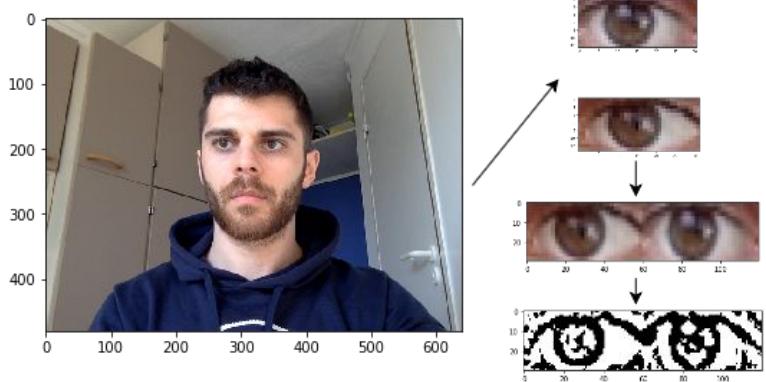
Obținerea datelor de antrenament



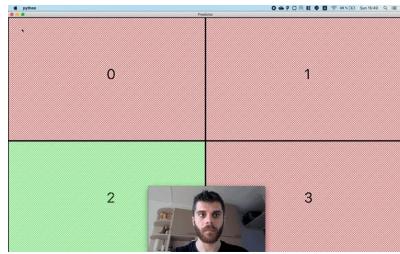
A screenshot of a Python IDE (PyCharm) showing the code for a "DataCollectorGUI" application. The code includes imports for "PyQt5", "PyQt5.QtWidgets", "PyQt5.QtCore", and "PyQt5.QtGui". It defines several classes: "BaseEvent", "BaseWindow", "iClicker", "DataCollector", and "DataCollectorGUI". The "DataCollectorGUI" class inherits from "QApplication" and contains methods for starting and stopping data collection, choosing a model, and displaying collected data. A configuration file "config.py" is also shown at the top of the project structure.

A screenshot of a terminal window titled "python" showing the execution of the "DataCollectorGUI" application. The command "python -m main" is run, and the application's graphical interface is displayed. The interface has four input fields: "Collectane date", "Processare date", "Antrenare model", and "Simulate". Below the interface, the terminal shows log output from the application's logger, including messages about capturing from the webcam and saving data.

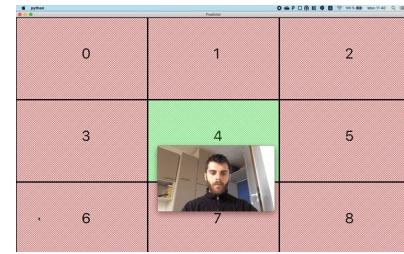
Procesarea datelor



Antrenarea modelelor predictive



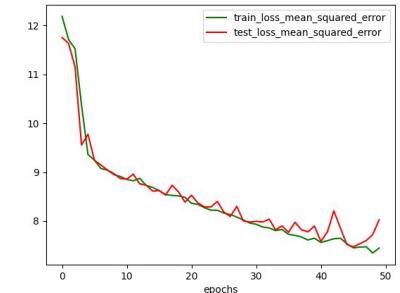
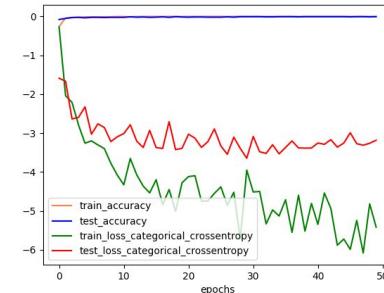
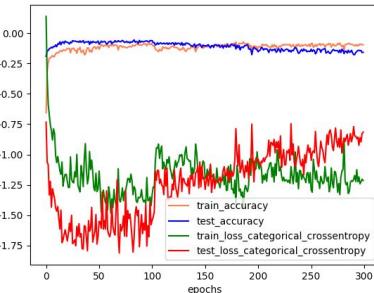
MLP, ochi + adaptive gaussian thresholding, clasificare



CNN folosind față, clasificare

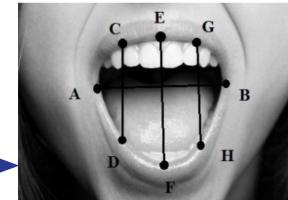
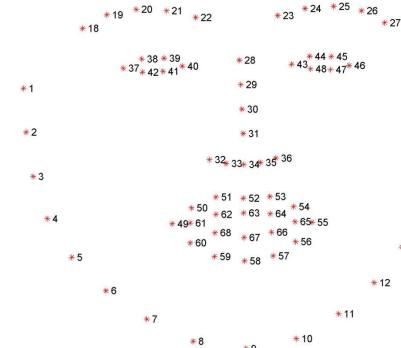


CNN folosind "banda oculară", regresie



Urmărirea ochilor și deplasarea cursorului

Model
CNN, "bandă
oculară"



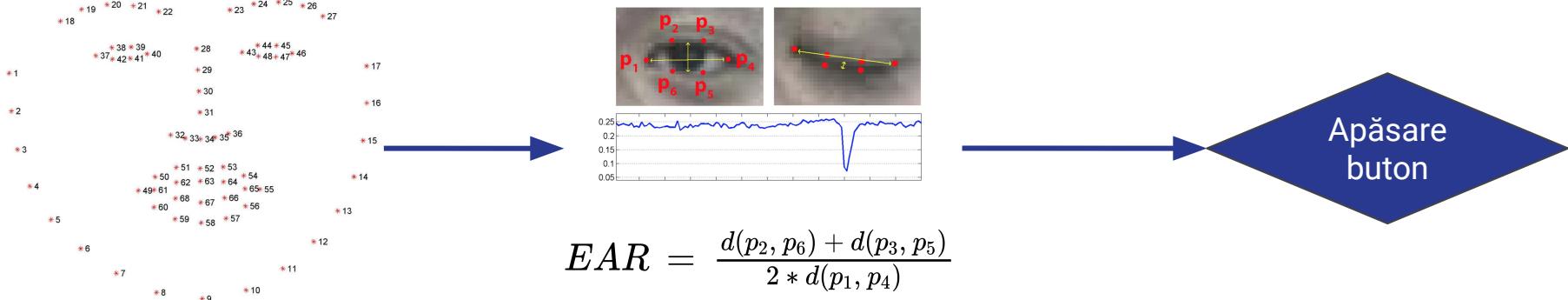
$$MAR = \frac{d(C, D) + d(E, F) + d(G, H)}{3 * d(A, B)}$$

0	1	2
3	4	5
6	7	8

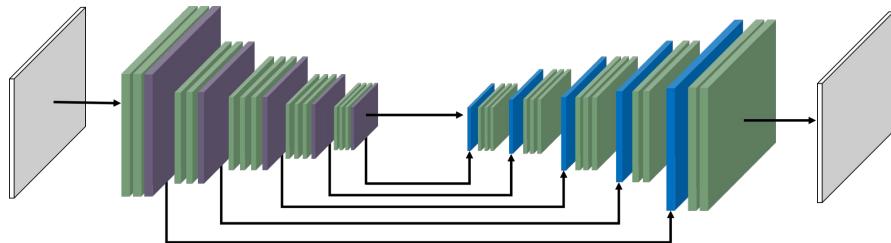
0	1	2
3	4	5
6	7	8

Deplasare
cursor

Simularea apăsării butoanelor



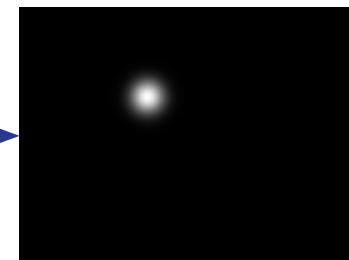
Identificarea reperelor faciale



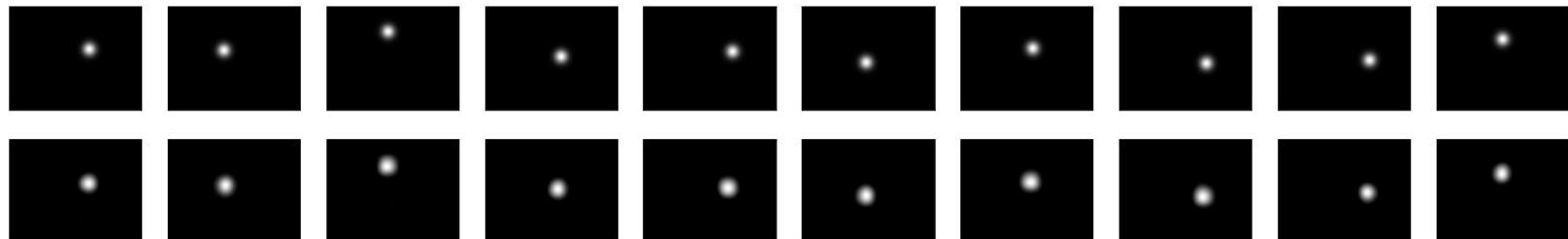
Arhitectură de tip Hourglass. Imagine preluată de pe [Medium](#)



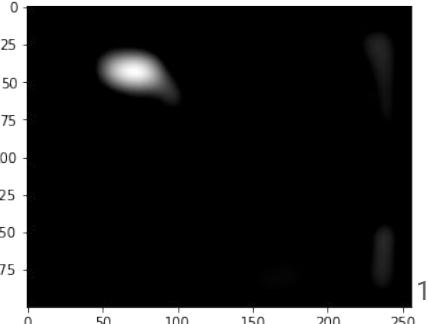
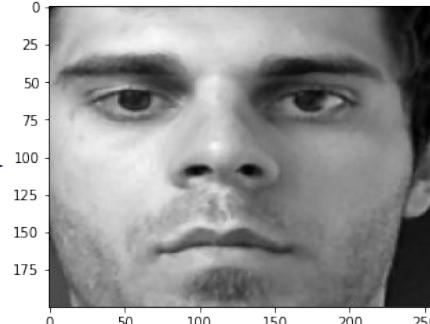
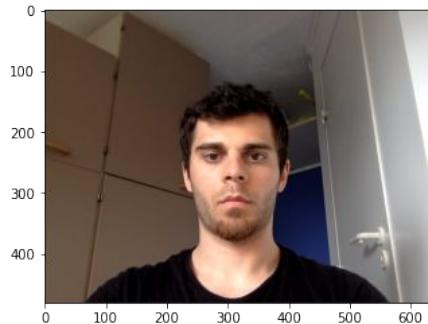
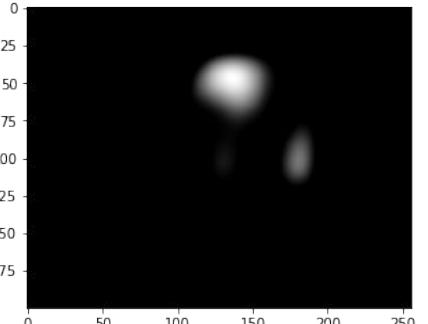
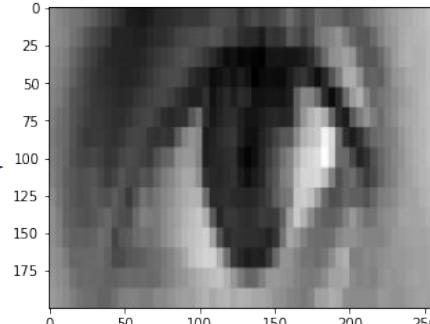
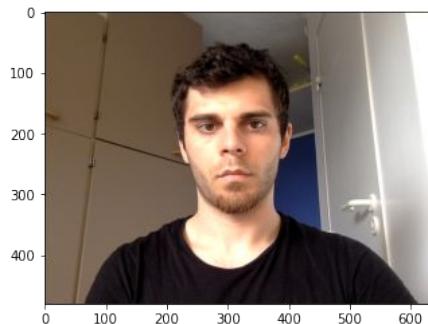
Mulțimea de date
[crowdpupil](#)



Codificare prin hărți
termografice



Identificarea reperelor faciale



Live Demo

Concluzii

Atingerea setului de obiective

- Urmărirea ochilor
- Simularea deplasării cursorului
- Simularea apăsării butoanelor

Îmbunătățirea soluției

- Apăsare dublă a butoanelor
- Glisare (scroll)
- Modele mai performante