

**UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**  
**FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI**  
**TEHNOLOGIA INFORMAȚIEI**

# **Face Mask Detection**

## **Proiect de semestru**

**Disciplina:** Sisteme inteligente de suport decizional

**Echipa:** 4 brațe

**Membri:** Mureșan Marius, Pîrlac Sebastian

**2023**

## 1. Descrierea temei:

Obiective: realizarea unei aplicații pentru a detecta dacă un om poartă o mască sau nu.

Performanțe: detectarea fetei și detectarea măștii dacă această este pusă pe față.

Implementare: rulare script `mask_detection.py`, capturare imagine și detectare mască folosind OpenCv, limbaj de programare: python.

Mod de finalizare: rezultatul dacă mască este pe față sau nu.

## 2. Sinteza proiectului (max. 10 rânduri):

Utilitatea unui sistem de detecție a măștilor este legată de menținerea sănătății oamenilor în contextul pandemiei de Covid-19, contribuind la prevenirea bolilor răspândite. Modalitatea de implementare (momentan) este un script de python împreună cu un set date predefinit, care au rolul de a detecta mască de pe față unei persoane cu ajutorul unui camere web. Principalele rezultate obținute sunt reprezentate de detectarea măștii.

## 3. Descrierea aplicației

Răspunde la următoarele întrebări:

- Cui se adresează aplicația?
  - Magazinilor, școlilor, spitalelor și altor unități ce implică interacțiune între oameni.
- Care este profilul utilizatorului?
  - Orice utilizator poate folosi această aplicație, deoarece nu este destinată unui public anume, fiind mai mult o aplicație medicală cu scopul de a ajuta întreaga lume.
- Care este componenta decizională?
  - Determinarea dacă o persoană poartă sau nu o mască facială.
- Care este componenta inteligentă?
  - Antrenarea, învățarea, adaptarea algoritmului de detecție.

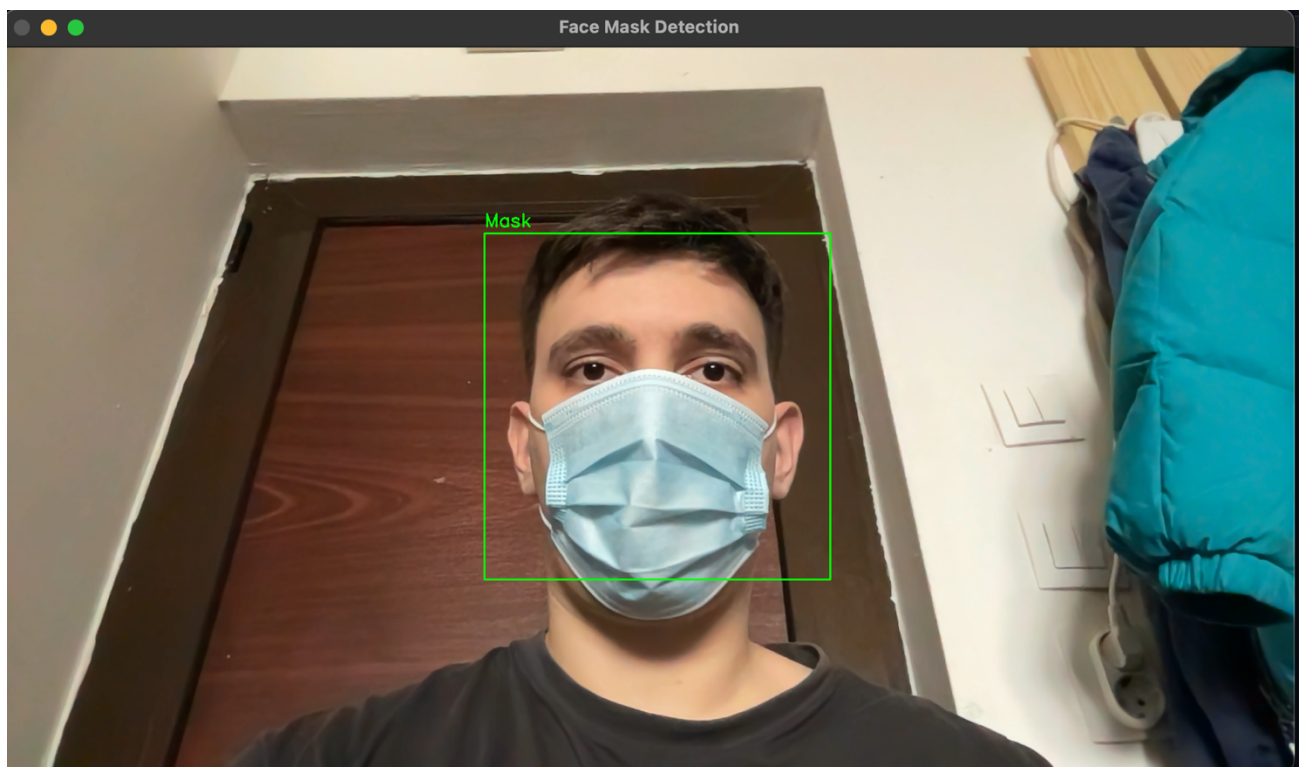
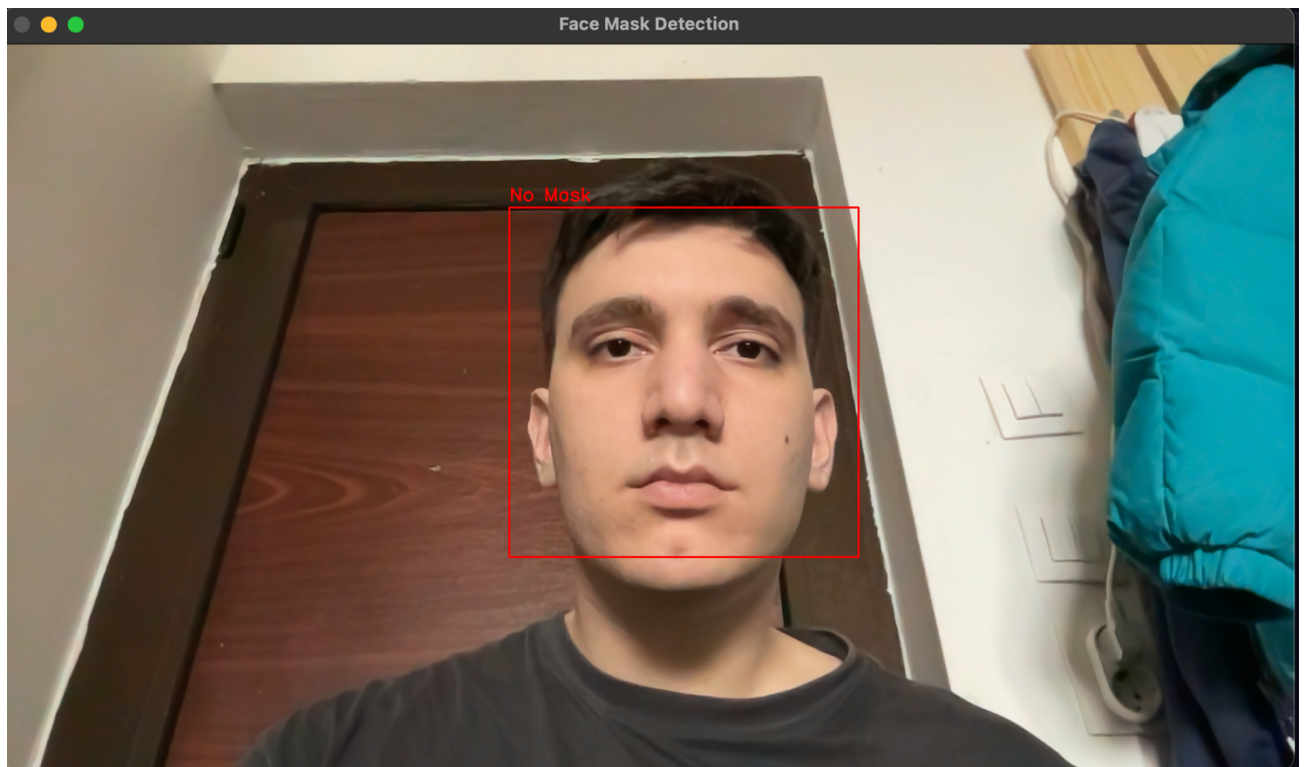
## 4. Implementare

Căutare, găsim, preluarea și adăugarea setului de date.

Realizarea script-ului după cum urmează:

1. Preprocesarea datelor:
  - Folosim o funcție pentru a încărca și preprocesa setul de date.
  - Setul de date este organizat în subdirectoare (cu mască, fără mască).
  - Funcția citește imaginile din subdirectoare, le redimensionează (64x64) și normalizează valorile pixelilor între 0 și 1.
2. Configurare set de date:
  - Setare folder rădăcină al setului de date.
  - Apelare funcție pentru a încărca și preprocesa datele.
3. Construirea unei rețele neuronale convoluționale (CNN)
  - Folosim TensorFlow (bibliotecă de calcul, antrenare Machine learning) pentru a construi un CNN simplu pentru clasificare binară (cu mască, fără mască).
4. Împărțire set de date
  - Setul de date se împarte în seturi de antrenament și testare
5. Convertire etichete la valori numerice
  - Convertim etichetele setului la etichete numerice (0,1).
6. Antrenare model
  - Modelul CNN este antrenat pe setul de antrenament și validat pe setul de testare
7. Detectarea măștii
  - Folosim OpenCv pentru a captura cadre video de la camera web
  - Folosim OpenCv pentru detectarea măștii folosind modelul antrenat
  - Folosim o funcție, care preia un cadru de imagine, detectează fața folosind detectorul Haar pentru recunoaștere facială și clasifică fiecare față (cu mască sau fără).
8. Rulare aplicație
  - Captare cadre video, aplicare detectare, afișare rezultat.

## 5. Rezultate experimentale



În prima imagine aplicația analizează fața persoanei, își dă seama că aceasta nu poartă o mască, iar prin urmare fața acesteia este marcată cu roșu indicând un rezultat negativ.

În a doua imagine datorită faptului că persoana poartă o mască, fața acesteia este marcată cu verde, prin urmare avem un rezultat pozitiv.

## 6. Concluzii

Obiectivele propuse au fost realizate în procent de 80%, aplicația funcționează în parametrii, însă mai este loc de dezvoltare: prin eficientizarea detectării măștii și prin antrenarea aplicației cu un set de date mai variat.

Dificultățile întâmpinate au fost că nu am găsit un tutorial complet, au fost câteva probleme cu librăriile implementate și faptul că detecția măștii având un fundal alb în spate se făcea mai greu din cauza setului de date.

Dezvoltări ulterioare:

- Arătarea procentului de corectitudine a măștii;
- Verificare dacă masca purtată este una medicală sau nu;

## 7. Contribuții

2 autori: Muresan Marius (MM), Pirlac Sebastian (PS).

Contribuțiile pot fi enumerate astfel:

alegere temă – MM, PS;  
documentare bibliografică – MM,PS ;  
stabilire obiective și mod de implementare – MM;  
implementare Python – MM;  
identificare scenarii de utilizare – MM,PS;  
metodologie testare – MM,PS;  
testare și colectare rezultate – MM,PS;  
interpretare rezultate – MM,PS;  
redactare material scris, prima variantă – MM;  
verificare și corectare material scris – MM,PS;  
realizare prezentare – MM,PS.

## 8. Bibliografie

<https://chat.openai.com/>

<https://github.com/chandrikadeb7/Face-Mask-Detection>

<https://www.youtube.com/watch?v=krTHVKkUEYw&list=WL&index=177>

<https://www.youtube.com/watch?v=IOI0o3C xv9Q&list=WL&index=176&t=422s>