

Prelucrarea informațiilor video preluate de la o camera WEB



Huza Lisandra

30223

Prelucrarea informațiilor video preluate de la o camera WEB

Cuprins

Introducere

- Context

- Specificații

- Motivație personală

- Limbaje alternative

Studiu Bibliografic

Design și analiză

- Inspirație

- Faza incipientă- familiarizarea

- Implementarea codului

- Testare

Bibliografie

Introducere

1. Context

Obiectivul acestui proiect este de a prelucra informațiilor video preluate de la o camera WEB, implicit de a identifica activ obiectele aflate în mișcare. Imaginile preluate de la camera WEB vor fi procesate pentru a se identifica dacă obiectele surprinse sunt în mișcare activă.

Acest program poate fi folosit de către persoane fizice, în propria locuință, pentru a fi informați despre starea acesteia, în momentele în care aceștia nu sunt la domiciliu și doresc o supraveghere amplă. Dar și în cadrul unor companii unde se dorește o monitorizare eficientă și o înștiințare asupra posibilelor daune.

2. Specificații

Programul va primi imagini cu obiecte în mișcare, iar cu ajutorul unui algoritm de prelucrare, acesta va analiza și decodifica imaginile, reușind astfel să identifice obiectele aflate în mișcare.

3. Motivație personală

Am ales să lucrez la acest proiect din dorința de a învăța analiza datelor primite prin intermediul unui mediu video. Astfel am oportunitatea de a lucra într-un mediu de programare bazat pe limbaj Python, folosind biblioteca open-source -> OpenCV. Consider că, acest proiect mă va ajuta să îmi deschid orizonturile spre câmpul de procesare video și de a experimenta simulări mai ample ale programelor scrise. Am ales să lucrez cu OpenCv-Python datorită surselor accesibile și ajutorul pe partea de implementare pe care acesta le oferă. Dar și de rapiditatea și eficiența codului, codul fiind mai lizibil și mai eficient în acest mod.

4. Limbaje alternative

- JavaScript: Este folosit în principal pentru dezvoltarea front-end a aplicațiilor web, dar poate fi folosit și în back-end. Poți să scrii cod JavaScript direct în browser

- C++: Este un limbaj de programare de nivel înalt cu suport pentru programarea orientată pe obiecte. Este utilizat pe scară largă în dezvoltarea de software, inclusiv în aplicații care necesită performanță ridicată, cum ar fi jocurile video sau motoarele de fizică.

- Java: Este un limbaj de programare orientat pe obiecte, care este utilizat într-o varietate de aplicații, de la dezvoltarea de aplicații mobile la servere web.

- PHP: Este un limbaj de programare folosit în principal pentru dezvoltarea de aplicații web pe partea de server
- C#: Este un limbaj de programare orientat pe obiecte dezvoltat de Microsoft. Este utilizat în principal pentru dezvoltarea de aplicații Windows și jocuri video cu Unity.

Studiu bibliografic

Python este un limbaj de programare interpretat și orientat pe obiecte, creat în 1991 de Guido van Rossum. Acesta a fost conceput pentru a fi un limbaj simplu, clar și elegant. Este un limbaj excelent pentru începători, dar este și puternic și util, fiind folosit pe scară largă în industrie.

Avantajele limbajului Python includ:

- Flexibilitate: Python oferă multă flexibilitate programatorilor. Spre deosebire de C++, care necesită multe linii de cod pentru a executa o singură comandă, Python necesită o singură linie².
- Portabilitate: Programarea în Python se face la fel, indiferent de sistemul de operare (Windows, Linux)².
- Claritate: Codul Python este ușor de înțeles, ceea ce este vital atunci când înveți bazele programării².
- Utilizare largă: Python este folosit într-o gamă largă de aplicații, de la dezvoltarea de aplicații web la știință și divertisment³.

OpenCV (Vedere computerizată cu sursa deschisă) este o bibliotecă de funcții informatice specializată pe vedere computerizată în timp real, fiind cea mai mare bibliotecă de vedere computerizată din lume. Este open source, conține peste 2500 de algoritmi și este operată de fundația non-profit Open Source Vision Foundation. Este utilizat în instrumente pentru detalii 2D și 3D, estimarea ego-mișcării, sisteme de recunoaștere facială, recunoașterea gesturilor, interacțiune om-calculator (HCI), roboți mobili, înțelegerea mișcării.

Design si analiză

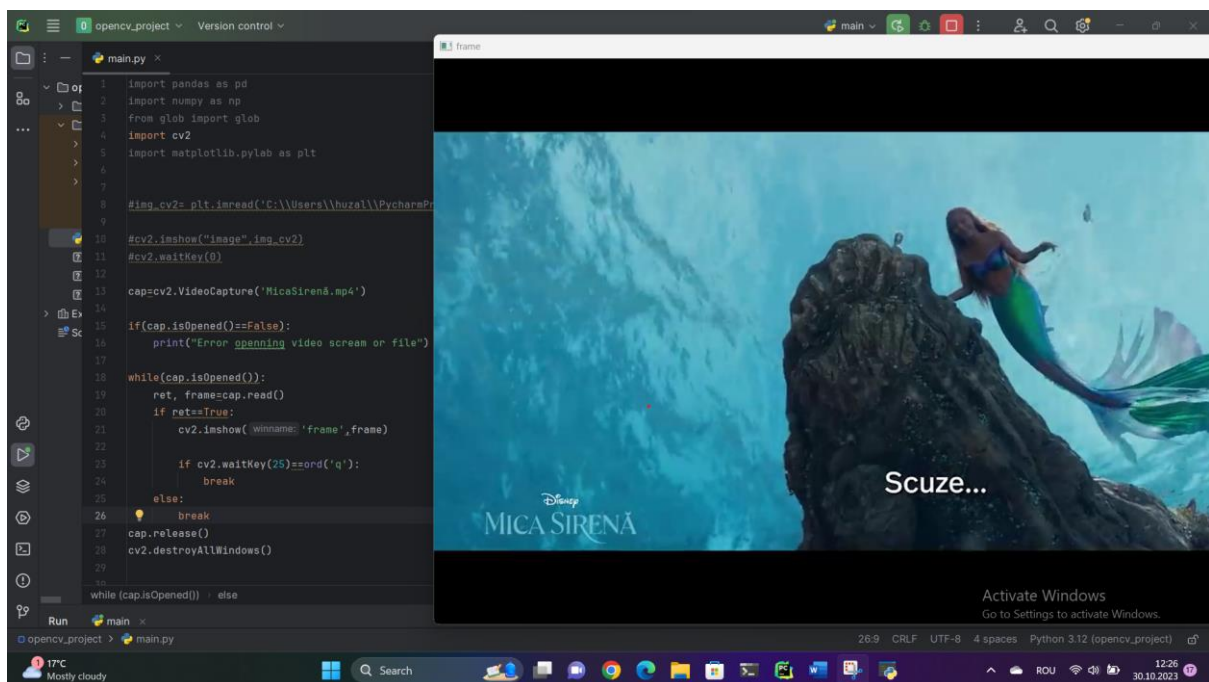
1. Inspirație

Ca și principală sursă de inspirație am ales informațiile furnizate de către browser, precum YouTube, de unde pot înțelege în întregime lucrul cu acest mediu de dezvoltare a aplicațiilor și a programelor. Dar și analiza altor programe care utilizează limbajul Python și biblioteca OpenCv.

2. Faza incipientă- familiarizarea

Pentru început, am avut nevoie de procurarea IDE-ului pentru dezvoltarea programului, astfel m-am ajutat de îndrumările explicate pentru descărcarea acestuia și setarea configurațiilor.

Un prim pas în dezvoltarea programului a lor deschiderea unei imagini, urmată de cea a unui video '.mp4', astfel începând să prindă contur lucrurile de bază.



3. Implementarea codului

```
1 import pandas as pd
2 import numpy as np
3 from glob import glob
4 import cv2
5 from matplotlib import pyplot as plt
```

-“import numpy as np” importă biblioteca numpy, care este folosită pentru lucrul cu vectori și matrici multidimensionale. Aliasul np este folosit pentru a accesa funcțiile și clasele din numpy mai ușor.

-import cv2 importă biblioteca cv2, care este folosită pentru procesarea și vizualizarea imaginilor și a videoclipurilor. Biblioteca cv2 este de fapt o interfață Python pentru biblioteca OpenCV, care este scrisă în C++. Nu este folosit niciun alias pentru cv2.

-“vid=cv2.VideoCapture(0)” creează un obiect de tip VideoCapture, care este folosit pentru a captura cadre dintr-o cameră web. Parametrul 0 înseamnă că se folosește camera web implicită a sistemului.

-“obj=cv2.createBackgroundSubtractorMOG2(history=2)” creează un obiect de tip BackgroundSubtractorMOG2, care este folosit pentru a extrage prim-planul din imaginile capturate, eliminând fundalul. Parametrul history înseamnă că se folosesc ultimele 2 cadre pentru a estima fundalul.

-“kernel=np.ones((3,3),np.uint8)” creează o matrice numpy de dimensiune 3x3, cu toate elementele egale cu 1 și de tip uint8. Această matrice este folosită ca un nucleu pentru operațiile de erodare și dilatare.

-“kernel2=None” inițializează o variabilă kernel2 cu valoarea None. Această variabilă este folosită ca un nucleu pentru operația de dilatare.

-în cadrul unei bucle infinite:

- se citește următorul cadru de la camera web și stochează rezultatul în două variabile: ret și frame1. Funcția vid.read returnează două valori: prima este un indicator boolean care arată dacă citirea a reușit sau nu, iar a doua este o matrice numpy care reprezintă imaginea capturată.

- dacă citirea nu a eșuat:

- se inversează imaginea capturată pe axa orizontală, adică se oglindește.

- se aplică obiectul BackgroundSubtractorMOG2 pe imaginea capturată și se returnează o mască binară care indică zonele care aparțin prim-planului. Funcția obj.apply este folosită pentru a extrage prim-planul dintr-o imagine, folosind un algoritm bazat pe modele gaussiene amestecate.

aplică o pragere binară pe masca binară, adică transformă valorile pixelilor în 0 sau 255, în funcție de un prag

- se aplică o operație de erodare pe masca binară, adică micșorează zonele albe și mărește zonele negre

- se aplică o operație de dilatare pe masca binară, adică mărește zonele albe și micșorează zonele negre

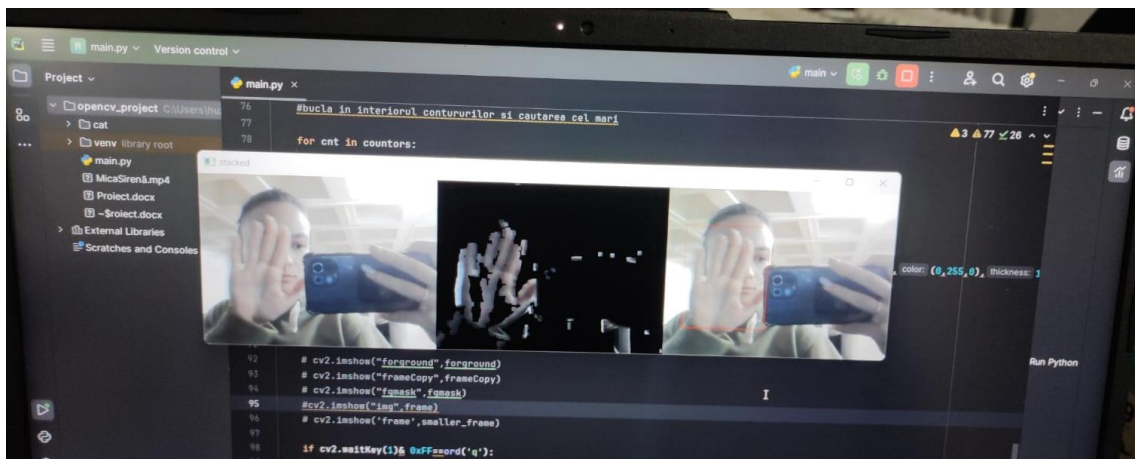
-se găsesc contururile din masca binară, adică zonele albe care au o formă închisă

-se începe o buclă for care parcurge lista de contururi returnată de funcția `cv2.findContours`

- dacă conturul curent are o arie mai mare decât 20000 de pixeli:
se calculează coordonatele și dimensiunile dreptunghiului care încadrează
conturul curent, se desenează un dreptunghi roșu în jurul conturului curent pe
copia imaginii capturate, se scrie un text verde cu mesajul "Obiect detectat"
lângă dreptunghiul roșu pe copia imaginii capturate

-se aplică o operație logică de tip ȘI pe imaginea originală și masca binară, adică păstrează doar pixelii care aparțin prim-planului

- `stacked=np.hstack((frame,foreground,frameCopy))` combină cele trei imagini într-o singură imagine, alăturându-le pe orizontală.



4. Testare

-pentru testarea codului, m-am folosit de obiectele din cameră, pe care le-am mișcat din anumite poziții și cu viteze diferite pentru a putea observa funcționalitatea corectă a programului, dar și acuratețea acestuia.

Bibliografie

Python OpenCV: Capture Video from Camera

<https://www.geeksforgeeks.org/python-opencv-capture-video-from-camera/>

OpenCV Python Tutorial #3 - Cameras and VideoCapture

https://www.youtube.com/watch?v=rKcwcARdg9M&list=PLV9WBSWTU64PFXw2Av9t2Hj6glfKgmtrg&index=16&ab_channel=TechWithTim

How to Detect Moving Objects in Video using OpenCV and Python

https://www.youtube.com/watch?v=YSLVAgclCo&list=PLV9WBSWTU64PFXw2Av9t2Hj6glfKgmtrg&index=17&ab_channel=EranFeit

Motion detection using python | | python opencv project

https://www.youtube.com/watch?v=oxmZ9zczptg&list=PLV9WBSWTU64PFXw2Av9t2Hj6glfKgmtrg&index=18&t=10s&ab_channel=Iknowpython