**CAMSEES**

**Student 1: Darius-Ovidiu Firan**

**Student 2: Casian-Gabriel Șuli**

**1. Introducere**

Camsees se folosește de orice dispozitiv video, conectat sau integrat în calculatorul pe care rulează, pentru a detecta și evidenția mișcarea care apare în imagine. De asemenea, în momentul în care detectează mișcare, va porni și înregistrarea care va fi stocată local. Proiectul nostru nu este ceva inovativ din punct de vedere al tehnologiilor folosite sau al caracteristicilor, dar este foarte ușor de folosit și tot odată, gratuit.

**2. State of the art**

**Motion Detector Pro**

Motion Detector Pro este o aplicație pentru camere de securitate pentru acasă, care utilizează camera încorporată pentru a detecta mișcările din zona înconjurătoare folosind un algoritm avansat de detectare a mișcării. Dacă detectează mișcare, trimite un e-mail cu un link către un alt telefon mobil, permițându-vă să monitorizați de la distanță o zonă folosind telefonul Android. Folosiți-l pentru a urmări animalul dvs. de companie, afacerea/biroul sau acasă. Cu Motion Detector Pro puteți să:

✓ Faceți o fotografie când camera vede o mișcare și trimiteți-o la un e-mail sau la alt telefon. Excelent pentru a prinde hoți, pentru a-ți urmări afacerea/casa sau pentru a fi cu ochii pe animalul tău de companie

✓ Indicați în timp real pe afișajul telefonului unde este detectată mișcarea

✓ Stocați imaginile pe cloud sau local pe cardul SD al telefonului

**Alfred Home Security Camera**

⏩ LIVE STREAM 24/7: Urmărește videoclipul live de înaltă calitate de oriunde

⏩ ALERTĂ DE INTRUS INTELIGENT: Primiți alerte instantanee când camera este declanșată de Detectarea mișcării

⏩ DEPOZITARE NELIMITATĂ în CLOUD: Urmăriți, descărcați și partajați înregistrările dvs. video în orice moment

⏩ FILTRU DE LUMINĂ SLABĂ: Întărește securitatea casei pe timp de noapte sau când împrejurimile sunt întunecate

⏩ WALKIE-TALKIE: descurajează hoții, interacționează cu vizitatorii sau animalele de companie și liniștește bebelușii

⏩ ZOOM, PROGRAMARE, REMINDER, TRUST CIRCLE, SIRENA

**Avantajele și dezavantajele utilizării aplicației noastre**

|  |  |  |  |
| --- | --- | --- | --- |
| **Caracteristici** | **Motion Detector Pro** | **Alfred Home Security Camera** | **Camsees** |
| Link store | [Google Play](https://play.google.com/store/apps/details?id=dk.mvainformatics.android.motiondetectorpro.activity&hl=en&gl=US) | [Google Play](https://play.google.com/store/apps/details?id=com.ivuu&referrer=utm_source%3Dlandingpage%26utm_medium%3Dreferral%26utm_campaign%3Dtop) | - |
| Notă store | 3.9 / 5 | 4.8 / 5 | - |
| Nr. instalări | 1M+ | 10M+ | - |
| Nr. ratinguri | 12.6K | 595K | - |
| Ads/in-app purchases | x | x | ? |
| Login/user management | x | x | - |
| Update în timp real | x | x | x |
| Notificări | x | x | - |
| Zoom | - | x | - |
| Alarmă | - | x | - |
| Înregistrare | - | x | x |
| Încadrare adaptivă a obiectelor în mișcare | x | - | x |

**Tab. 1**

**3. Design și implementare**

Unul din conceptele de bază folosite în aplicația noastră este cel de Background Subtraction(BS). Metodele de Background Subtraction rezolvă problema extragerii primului plan prin crearea unui model de fundal. Pipeline-ul BS-ului poate conține următoarele faze:

* generarea fundalului – procesarea a N cadre pentru a oferi imaginea de fundal
* modelarea fundalului – definirea modelului pentru reprezentarea fundalului
* actualizarea modelului în fundal – introducerea algoritmului de actualizare a modelului pentru gestionarea modificărilor, care apar în timp
* detectarea prim-planului – împărțirea pixelilor în seturi de fundal sau prim-plan

Diagram

Description automatically generated

**Fig. 1. BS basic pipeline**

Ieșirea BS-ului constă într-o mască binară, care separă pixelii cadrului în două seturi: pixeli din prim-plan și de fundal.

Trebuie menționat că frecvent în abordările BS-ului accentul este mutat pe implementarea modelelor avansate de fundal și aspectul de reprezentare robust a caracteristicilor.

Algoritmul folosit de Camsees este GSoC. În timpul Google Summer of Code (GSoC) 2017, a fost furnizată avansarea LSPB: BackgroundSubtractorGSOC. Metoda GSoC BS a fost introdusă pentru a face LSBP mai rapid și mai robust. Metoda se bazează pe valorile de culoare RGB în loc de descriptori LSBP și atinge performanțe ridicate pe CDnet-2012.

Implementarea GSoC BS nu se referă la niciun articol, prin urmare, să vedem punctele de bază, explorând sursa [bgfg\_gsoc.cpp](https://github.com/opencv/opencv_contrib/blob/master/modules/bgsegm/src/bgfg_gsoc.cpp)[1]. În primul rând, trebuie să acordăm atenție parametrilor de instanțiere BackgroundSubtractorGSOC:

Text

Description automatically generated

**Fig. 2. BackgroundSubtractorGSOC instantiation parameters**

În spatele lor există următoarele semnificații:

* mc: steag de compensare a mișcării camerei
* nSamples: numărul de mostre de menținut în fiecare punct al cadrului
* replaceRate: probabilitatea înlocuirii vechiului eșantion – cât de repede va fi actualizat modelul
* propagationRate: probabilitatea de propagare la vecini
* hitsThreshold: câte rezultate pozitive trebuie să obțină eșantionul înainte de a fi considerat un posibil înlocuitor
* alfa: coeficient de scară pentru prag
* beta: coeficient de părtinire pentru prag
* blinkingSupressionDecay: factor de dezintegrare a suprimării intermitente
* blinkingSupressionMultiplier: multiplicator de suprimare intermitent
* noiseRemovalThresholdFacBG: puterea de eliminare a zgomotului pentru fundal
* noiseRemovalThresholdFacFG: puterea de eliminare a zgomotului pentru prim-plan

Noi am implementat acest algoritm folosind OpenCV. Mai departe voi enumera etapele parcurse:

**Video Processing**

Încărcați și procesați datele video cu OpenCV [VideoCapture](https://docs.opencv.org/4.3.0/d8/dfe/classcv_1_1VideoCapture.html#ac4107fb146a762454a8a87715d9b7c96)[2]:

Text

Description automatically generated

**Fig. 3. Video Processing**

**Model Initialization**

Inițializați modelul:



**Fig. 4. Model Initialization**

**Obtaining results**

Obțineți rezultatele (dimensiunea inițială a frame-urilor va fi 1920x1080):

Text

Description automatically generated

**Fig. 5. Obtaining results**

**Visualizing results**

Vizualizați rezultatele folosind [imshow](https://docs.opencv.org/4.3.0/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563)[3]:

Text

Description automatically generated

**Fig. 6. Visualizing results**

Pe ecran vor apărea frame-ul inițial (un fragment din input-ul video), masca prim-planului obținută cu OpenCV BS-GSoC și rezultatul obținut prin subtraction.

Am ales acest model și această implementare deoarece, conform unor teste relizate pe seturile de date [CDnet-2012](http://jacarini.dinf.usherbrooke.ca/dataset2012)[4], este o variantă îmbunătățită a [LSBP](https://www.semanticscholar.org/paper/Subtraction-using-Local-SVD-Binary-Pattern-Guo-Xu/b0985612b033854763080a9d42adddbe10b2be72?p2df)[5].

**4. Utilizare**

Pentru a putea rula aplicația trebuie este nevoie de Python versiunea 3.9 , plus două importuri: [numpy](https://numpy.org/doc/1.21/)[6] (versiunea 1.21.5) și cv2([opencv](https://docs.opencv.org/4.6.0/)[7] 4.6.0.66). De asemenea este nevoie de o cameră video conectată la calculatorul pe care rulează aplicația în cazul în care se dorește monitorizarea live a mișcării de pe aceasta. Camsees poate fi rulat standard sau cu câteva argumente cum ar fi “-v” sau “--video” urmat de path-ul către un videoclip pe care doriți să reperați obiecte în mișcare sau “-r” sau “--record” dacă doriți să se înregistreze.

Odată pornită, aplicația va afișa pe ecran 3 ferestre: “Initial Frames” (cadrul inițial preluat de pe camera web sau din videoclipul pe care a fost rulată aplicația), “Foreground Masks” (masca obținută cu BS-GSoC), “Subtraction Result” (rezultatul background-ului live obtinut).

**5. Rezultate și concluzii**

În urma testării aplicației au fost necesare mai multe modificări fine asupra variabilelor. Printre cea mai importantă a fost valoarea ariei obiectelor.

Graphical user interface, text, application

Description automatically generated

**Fig. 1. Aria obiectelor**

În funcție de această valoare apar box-urile în jurul obiectelor care se mișcă. Astfel, la o valoare prea mare am descoperit că nu detectează mișcările mici. În caz contrar, la o valoare mică, în loc să detecteze un obiecte întreg (de exemplu o mână) pe care să o încadreze, detectează finele mișcări ale mâinii pe care le încadrează cu zeci de box-uri de dimensiune mică.

Cel mai greu pas a fost adăugarea funcționalității de a înregistra ecranul atunci când este detectată mișcare. În momentul prezentării, acest feature nu este complet. Presupunerea noastră este că e o problemă de codare a frame-ului pentru a fi scris în format .mp4, însă ar fi necesare mai multe teste pentru a rezolva problema.

Concluzionând, putem spune că am aflat cât de mult a avansat tehnologia în domeniul detectării obiectelor folosind chiar și cea mai simplă cameră. Am presupus înainte să începem că va fi greu procesul de conectare în timp real a unei camere, însă modulele din python ne-au ușurat munca. În continuare aplicația se mai poate extinde, o idee fiind adăugarea unui notificator care să trimită direct pe telefonul mobil un mesaj când apare mișcare în cadru.

**Bibliografie**

[1] - <https://github.com/opencv/opencv_contrib/blob/master/modules/bgsegm/src/bgfg_gsoc.cpp>

[2] - <https://docs.opencv.org/4.3.0/d8/dfe/classcv_1_1VideoCapture.html#ac4107fb146a762454a8a87715d9b7c96>

[3] - <https://docs.opencv.org/4.3.0/d7/dfc/group__highgui.html#ga453d42fe4cb60e5723281a89973ee563>

[4] - <http://jacarini.dinf.usherbrooke.ca/dataset2012>

[5] - <https://www.semanticscholar.org/paper/Subtraction-using-Local-SVD-Binary-Pattern-Guo-Xu/b0985612b033854763080a9d42adddbe10b2be72?p2df>

[6] - <https://numpy.org/doc/1.21/>

[7] - <https://docs.opencv.org/4.6.0/>

<https://daniel-schwalm.medium.com/recording-video-for-object-detection-with-python-and-opencv-32e34bb02772>

<https://www.codespeedy.com/save-webcam-video-in-python-using-opencv/>

<https://learnopencv.com/background-subtraction-with-opencv-and-bgs-libraries/>