

Gépi látás

GKNB\_INTM025

**Biztonsági kamera mozgás felismerését és nyomon követése**

Tóth Norbert

# Bevezetés

**A Projekt célja:**

A projekt célja az volt, hogy felismerje és nyomon kövesse egy videó felvételen történi személy mozgását. A rendszer adaptív mivel video feed inputnak meg lehet adni videot mp.4 formátumban vagy akár webcamera élő képét is, vagy akár hogy praktikusabb legyen raspberry pi kamerán is futtatható lehet.

A kép felismerésénél cél volt, hogy az objektum amit felismer ember legyen ezért azt valahogyan ki kellett küszöbölni hogy kisebb nem emberi objektumokat ne érzékeljen.

# Tervezés

**A feladat megvalósítása:**

A mesterséges inteligencia segítségével sok-sok mód van a mozgásérzékelés, nyomon követés és elemzés elvégzésére az OpenCV-ben. Néhány nagyon egyszerű. Mások pedig nagyon bonyolultak.

A feladat megvalósítására 2 módszert alkalmaztam:

* Gaussian Mixture modell
* Háttér szegmentálás

Háttér szegmentálos a modell a valós idejű árnyékfelismerés nyomon követéshez van használva

Az adaptív Gauss-keverék modell a háttér-kivonáshoz, valamint a háttér-kivonás feladatának hatékony képi pixelenként történő adaptív sűrűség-becslésére szolgál

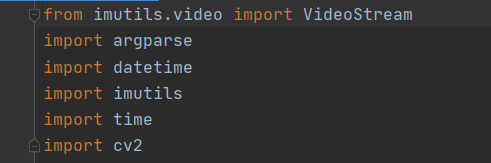
Mivel videófolyamunk háttere nagyrészt statikus és változatlan a videó egymást követő képkockáinál. Ezért, ha modellezni tudjuk a hátteret, figyelemmel kísérjük azt a lényeges változások szempontjából. Ha lényeges változás történik, akkor észlelhetjük - ez a változás általában megfelel a lekövetni kívánt objektum mozgásának.

Ezek a módszer a való világban könnyen kudarcot vallhat. Az árnyékolás, a visszaverődések, a fényviszonyok és a környezet bármilyen egyéb lehetséges változása miatt a hátterünk egészen másképp nézhet ki egy videó különböző képkockáiban.

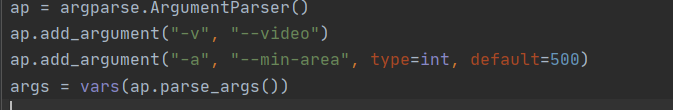
Ha a háttér másnak tűnik, akkor az eldobhatja algoritmusainkat. Éppen ezért a legsikeresebb háttér-kivonási, előtér-észlelő rendszerek rögzített kamerákat és vezérelt fényviszonyokat alkalmaznak.

A cél az, hogy a rendszer akár egy webkamerán vagy egy raspbery pi-n is futtatható legyen ezért a fent említett módszerek egyszerűbb változatát fogjuk alkalmazni, mivel annak ellenére hogy a módszerek hatékonyak számítási szempontból bonyolultak.

**Python és Opencv**



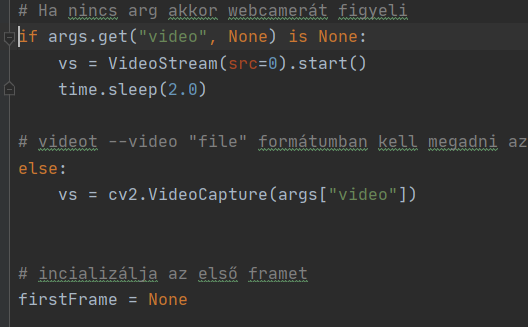
Ezek a sorok a szükséges csomagjainkat importálják.



Az argumentum parserbe kettő lehetőség van video input megadására, ha argumentumnak –video „előre felvett video” akkor lehetőség van egy tetszőleges video fájlon lefuttatni a programot, ha argumentumot üresen hagyjuk vagy a video formátum nem támogatott akkor automatikusan az OpenCV automatikusan a webkamera feedjét veszi alapul.

A –min-area érték megadja, a kép azon területének minimális mérete (pixelben), amelyet tényleges „mozgásnak” kell tekinteni.

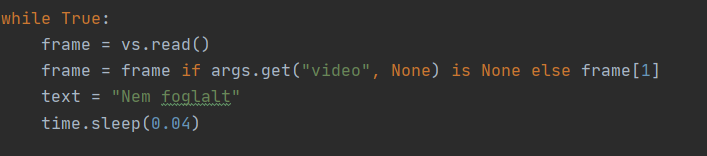
Ezt helyzethez viszonyulva változtatni kell, gyakran találunk egy kép kis régióit, amelyek jelentősen megváltoztak, valószínűleg zaj vagy a fényviszonyok változása miatt.



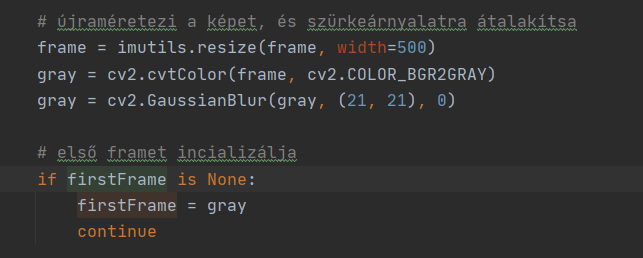
Ez az if-else függvény a video input beaolvasására szolgál.

Valamint beállítsuk a firstFramet none-ra az elképzelés az, hogy a videó input nem fog tartalmazni mozgást szóval lehetőség van lemodellezni a hátterét a megadott videostreamnek.

Ez konkrétan megadott videoknál gyakran hibába ütközik, mert nincsen lehetősége lemodellezni egy statikus hátteret.



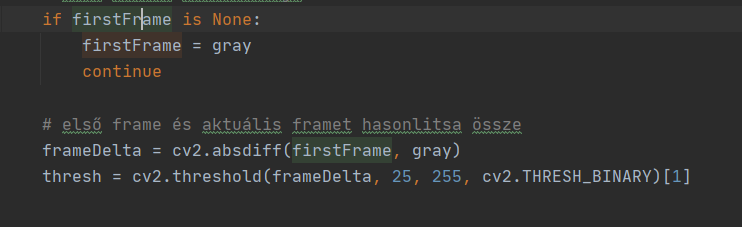
A framebe elmentsük a beaolvasott aktuális framet és kialakítunk egy textet ami arra fog szolgálni, hogy kiírja a képen történik-e mozgás.



Átméretezzük a framet hogy 500 pixel szélességű legyen - nem kell a nagy képeket közvetlenül a videofolyamból feldolgozni.

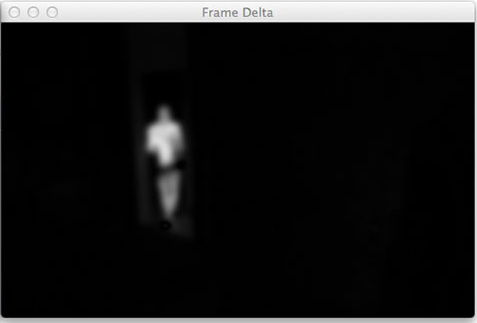
A képet szürkeárnyalatosra is konvertáljuk, mivel a szín nincs hatással a mozgásérzékelési algoritmusunkra. Végül a képek simításához Gauss-féle elmosódást alkalmazunk.

Gauss-simítást kell alkalmaznunk az átlagos pixelintenzitásokra egy 21 x 21 régióban. Ez segít elsimítani a magas frekvenciájú zajt, amely eldobhatja mozgásérzékelési algoritmusunkat.



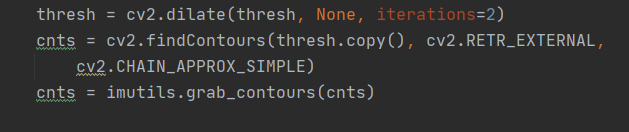
háttérünket az firstFrame változón keresztül modelleztük, felhasználhatjuk arra, hogy kiszámítsuk a különbséget a kezdeti képkocka és a későbbi új képkockák között a videó adatfolyamból.

Két képkocka közötti különbség kiszámítása egyszerű kivonás, ahol a megfelelő pixelintenzitás-különbségek abszolút értékét vesszük delta = |background\_model – current\_frame|

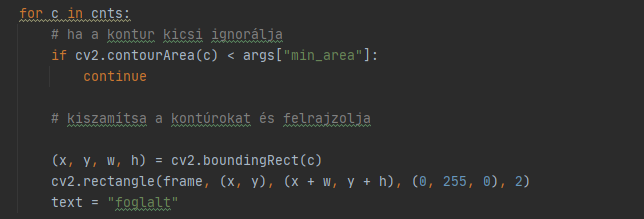


Igy néz ki ha változás történik a first framehez képest.

feltárjuk a kép azon területeit, amelyek csak jelentősen változnak a pixelintenzitás értékében. Ha a delta kevesebb, mint 25, akkor elvetjük a pixelt, és feketére állítjuk. Ha a delta nagyobb, mint 25, akkor fehérre állítjuk.



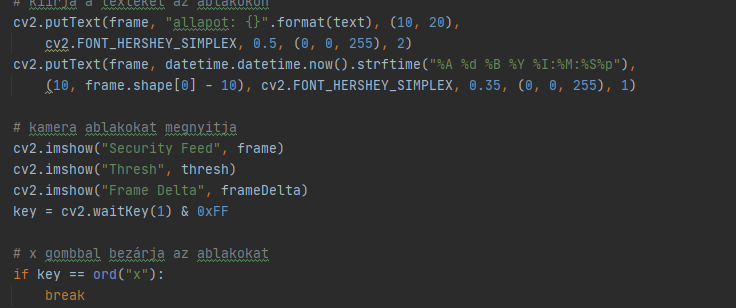
Tekintettel erre a threshold képre, egyszerű a kontúrészlelés alkalmazása e fehér régiók körvonalainak megtalálásához..



Elkezdjük a loopot az egyes kontúrjain, ahol leszűrjük a kicsi, lényegtelen kontúrokat

Ha a kontúrterület nagyobb, mint a megadott --min-area, akkor az előtér és a mozgás területét körülvevő határoló mezőt rajzoljuk körbe.

Továbbá frissítsük a text mezőt foglaltra.



Ezek a sorok lehetővé teszik számunkra a 3 mozgás követéshez szükséges ablakot.

**Program leírása**

A program ideális működéséhez szükség van hogy indításkor a kamera képen az aktuális háttér jelenjen meg ahhoz, hogy a firstframe letudja modellezni a hátteret.

A rendszer meglehetősen jól teljesít, annak ellenére, hogy mennyire leegyszerűsített, reálisak legyünk, az eredmények korántsem tökéletesek. Több határoló dobozt kapunk, annak ellenére, hogy a helyiségben csak egy ember mozog - ez korántsem ideális. És világosan láthatjuk, hogy a világítás apró változásai, mint például az árnyékok és a falra való visszaverődések, hamis pozitív mozgásérzékelést váltanak ki.

OpenCV erőteljesebb háttér-kivonási módszereire, amelyek tulajdonképpen az árnyékolással és a kis reflexióval tudnak számolni jobb eredményt adnának, de ehhez nagyob teljesítményű gépek kellenének.