Gépi látás

Biztonsági kamera mozgás felismerése és nyomon követése

Tóth Norbert

1. Bevezetés

A Projekt célja:

A projekt célja az volt, hogy felismerje és nyomon kövesse egy videó felvételen történő személy mozgását. A program lehetőséget ad arra, hogy video feed inputnak meg lehessen adni videot mp.4 formátumban vagy akár webcamera élő képét is, apróbb módosítások után akár elérhető az is hogy raspberry pi kamerán is futtatható legyen.

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.

2. Elmélet háttér

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 videófolyam háttere nagyrészt statikus és változatlan a videó egymást követő képkockáinál, ezért modellezni tudjuk a hátteret és figyelemmel kísérni azt a lényeges változások szempontjából. Ha lényeges változás történik, akkor észlelni lehet, ez a változás általában megfelel a lekövetni kívánt objektum mozgásának.

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

Ha a háttér másnak tűnik, akkor az eldobhatja algoritmusunkat. 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 fogom alkalmazni, mivel annak ellenére, hogy a módszerek hatékonyak számítási szempontból bonyolultak.

Szürkeárnyalati konverzió:

A Kép szürkeárnyalatos konvertálására azért van szükség mivel a szín nincs hatással a mozgásérzékelési algoritmusunkra.

A transzformációhoz csak a három szín átlagát kell választani, mivel RBG kép, ez azt jelenti, hogy R+B+G-t összeadjuk majd elosztjuk 3-mal és megkapjuk a kívánt szürkeárnyalatos képet.(R + G + B / 3). A programunkban ezt cv2.COLOR_BGR2GRAY függvénnyel valósítsuk meg.

Szürkeárnyalatos kép esetén leggyakrabban a [0, 255] intenzitástartomány használt. A 0 érték jelenti a fekete, a 255 a fehér színt. A köztes értékek a feketéből fehérbe tartó szürke átmenetet jelentik.

Gauss simítás

A gaussi simítás széles körben alkalmazott módszer, általában a képzaj és a részletek csökkentése érdekében. Előfeldolgozási szakaszaként használom annak érdekében, hogy javítsam a képszerkezeteket.

Programunkban cv2.GaussianBlur parancsot használjuk ennek megvalósítására.

Az elsimítás során képpontok közelebb kerülnek környezetük átlagához, azaz a kép „simább” lesz, de a szűrt kép intenzitásértékei a kiindulási kép intenzitástartományában maradnak.

Szűrés során minden pixel értéket a szomszédainak lineáris kombinációjával cserélünk le. A kernel vagy maszk adja meg a szomszédság méretét és a súlyokat.

Dilatáció

Dilatáció egy morfológiai művelet, amelyet a kép tulajdonságainak javítására használok. A dilatációhoz mint funkcióhoz két bemenetre van szükség, egy kép amit dilatációra szánok és kétdimenziós strukturáló elemre.

Kontúrok detektálása

A kontúrok egyszerűen egy görbeként magyarázhatók, amelyek összekötik az összes folytonos pontot. kontúrok az azonos intenzitású alak határai. Az alakzat határának (x, y) koordinátáit tárolja. Vonalnak csak két végpontjára van szükségünk. A cv.CHAIN_APPROX_SIMPLE eltávolítja az összes felesleges pontot és összenyomja a kontúrt, ezáltal memóriát takarít meg.

3.Tervezés

A feladat megvalósítása:

Python és Opencv

```
from imutils.video import VideoStream
import argparse
import datetime
import imutils
import time
import cv2
```

Ezek a sorok a szükséges csomagjainkat importálják.

```
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video")
ap.add_argument("-a", "--min-area", type=int, default=500)
args = vars(ap.parse_args())
```

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 van hagyva vagy a video formátum nem támogatott akkor automatikusan az OpenCV 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.

```
# Ha nincs arg akkor webcamerát figyeli
if args.get("video", None) is None:
    vs = VideoStream(src=0).start()
    time.sleep(2.0)

# videot --video "file" formátumban kell megadni az
else:
    vs = cv2.VideoCapture(args["video"])


# incializálja az első framet
firstFrame = None
```

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

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

Ez sok megadott videonál gyakran hibába ütközik, mert nincsen lehetősége lemodellezni egy statikus hátteret.

```
while True:
    frame = vs.read()
    frame = frame if args.get("video", None) is None else frame[1]
    text = "Nem foglalt"
    time.sleep(0.04)
```

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

```
    # újraméretezi a képet, és szürkeárnyalatra átalakítsa
    frame = imutils.resize(frame, width=500)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21, 21), 0)

    # első framet incializálja
    if firstFrame is None:
        firstFrame = gray
        continue
```

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

Végül a képek simításához Gauss-féle elmosódást alkalmazok.

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

```
if firstFrame is None:
    firstFrame = gray
    continue

# első frame és aktuális framet hasonlitsa össze
frameDelta = cv2.absdiff(firstFrame, gray)
thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH_BINARY)[1]
```

hátteret a firstFrame változón keresztül modellezem majd arra használom, hogy kiszámítsam 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.

feltárja a kép azon területeit, amelyek csak jelentősen változnak a pixelintenzitás értékében. Ha a delta kevesebb, mint 25, akkor elveti a pixelt, és feketére állítja. Ha a delta nagyobb, mint 25, akkor fehérre állítja.

```
    thresh = cv2.dilate(thresh, None, iterations=2)
    cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
```

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

```
for c in cnts:
    # ha a kontur kicsi ignorálja
    if cv2.contourArea(c) < args["min_area"]:
        continue

    # kiszamitsa a kontúrokat és felrajzolja

    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    text = "foglalt"
```

Elkezdi a loopot az egyes kontúrjain, ahol leszűri a kicsi és a lényegtelen kontúrokat.

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

Továbbá frissíti a text mezőt foglaltra.

```
# kiírja a textet az ablakokon
cv2.putText(frame, "allapot: {}".format(text), (10, 20),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
cv2.putText(frame, datetime.datetime.now().strftime("%A %d %B %Y %I:%M:%S%p"),
    (10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0, 255), 1)

# kamera ablakokat megnyitja
cv2.imshow("Security Feed", frame)
cv2.imshow("Thresh", thresh)
cv2.imshow("Frame Delta", frameDelta)
key = cv2.waitKey(1) & 0xFF

# x gombbal bezárja az ablakokat
if key == ord("x"):
    break
```

Ezek a sorok láthatóva teszik a 3 mozgás követéshez szükséges ablakot.

4. Felhasználói Leírás

A program működéséhez szükséges feltelepíteni az **Opencv 2**-es az verzióját, valamint az **imutilis** és a **numpy** nevű packageket.

parancsok: pip install imutils

pip install numpy

pip install opencv-python==2.4.9

pip install argparse

Ahhoz, hogy videófájlokat tudjunk megadni argumentumnak **–video „fajlnév”** formában kell hivatkozni, ha az argumentumot üresen hagyjuk vagy nem támogatott formátumú a videó, akkor a program a webkamera feedet fogja alapul venni.

Lehetőség van megadni **–a „”** formátumban azt, hogy mennyi kép azon területének minimális mérete (pixelben), amelyet tényleges „mozgásnak” kell tekinteni, ezzel kiküszöbölhető, hogy tévesen észleljen objektumokat, az előre beállított pixel 500, de ha a program tévesen érzékel objektumokat tanácsos ezt megnövelni.

Ha a felhasználó szeretné bezárni a program által megjelenített ablakokat, akkor erre az „x” gombbal van lehetősége.

A program ideális működéséhez szükség van hogy indításkor a kamera képen a háttér jelenjen meg mozgás nélkül ahhoz, hogy a firstframe letudja modellezni a hátteret, valamint hogy a kamera rögzített legyen.

5. Tesztelés

A teszteléshez olyan videokat kerestem, amik lehetőleg egy stabil álló háttérrel kezdődnek, hogy a programnak legyen lehetősége lemodellezni a hátteret.

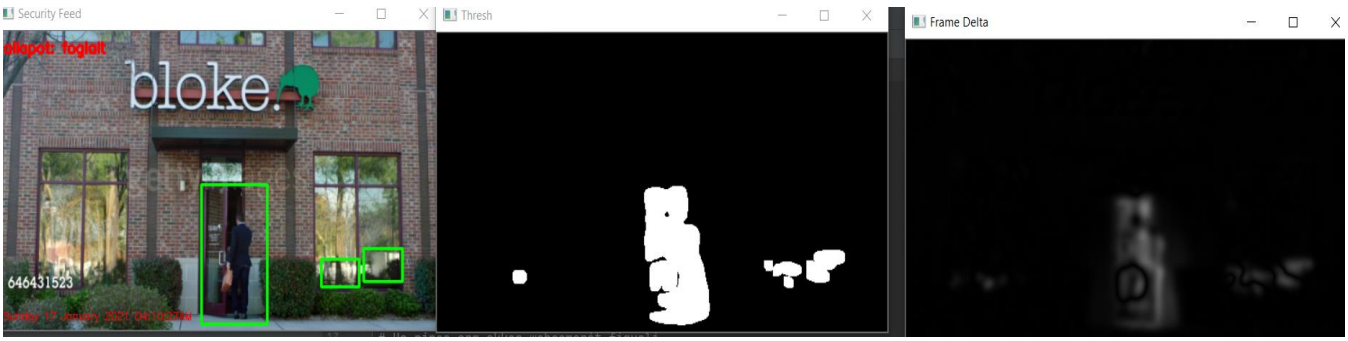
Ahhoz, hogy a program pontosan tudja érzékelni a mozgó testeket –min-area értéket egyes videokhoz külön kell be állitani, mert eszerint dönt a program, hogy milyen méretet kell mozgásnak tekinteni.


allapot: foglalt
Thresh
Frame Delta


Security Feed
allapot: foglalt
Sunday 17 January 2021 03:58:47AM
Thresh
Frame Delta
Ha nincs arg akkor webcamerát figyeli

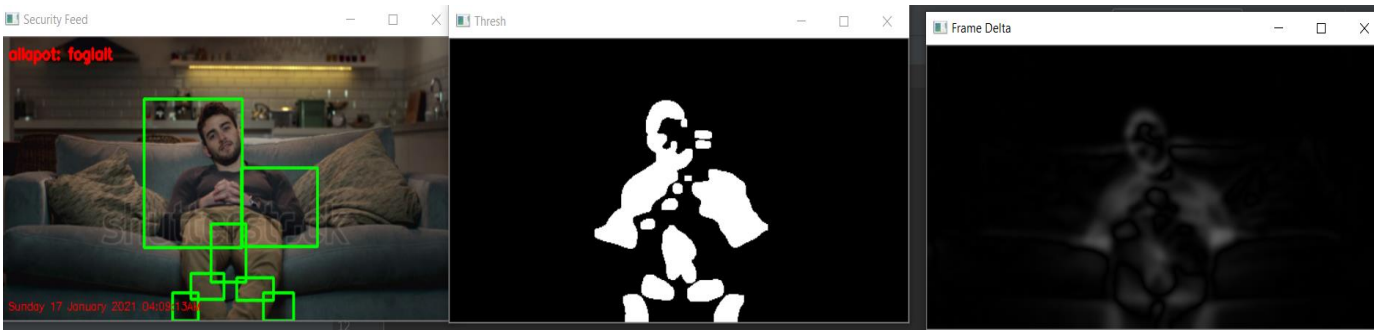

Security Feed
allapot: foglalt
Thresh
Frame Delta

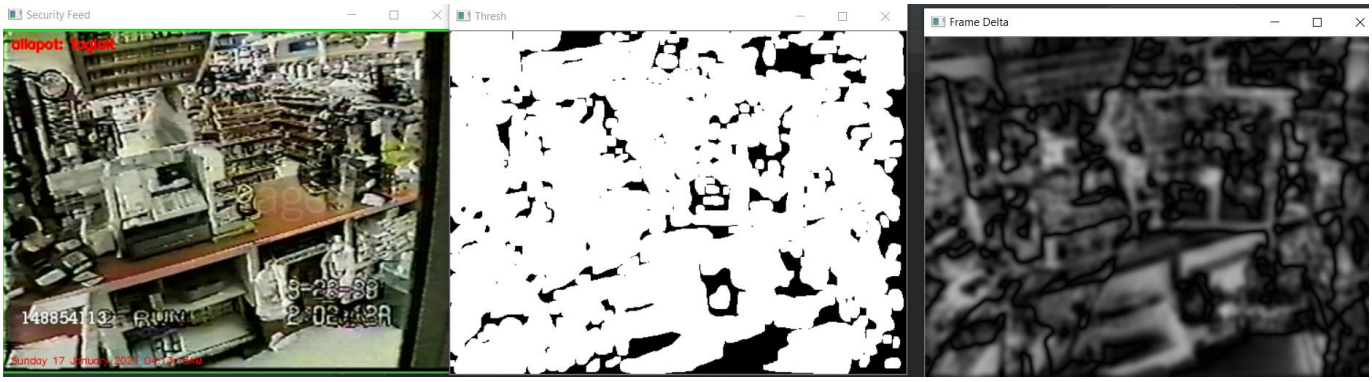

Security Feed
allapot: foglalt
Thresh
Frame Delta



Hibás próbálkozás:

Ha a video nem stabil helyzetben lévő kamerával van rögzítve eredményképp a képen látható fals pozitív visszajelzést kapunk.




Security Feed
Thresh
Frame Delta