**Jegyzet**

**Miért szükséges a szürkeárnyalatos konverzió?**

A Haar Cascade kaszkád osztályozó szürkeárnyalatos képeket igényel az arcfelismeréshez. Ennek oka a következő:

* A szürkeárnyalatos képek kevesebb információt tartalmaznak, mint a színes képek, így a feldolgozásuk gyorsabb.
* A szürkeárnyalatos képek kevésbé érzékenyek a fényviszonyokra és a zajra, így az arcfelismerés pontosabb.

**Hogyan működik az arcfelismerés az OpenCV-ben a detectMultiScale függvénnyel?**

A detectMultiScale az arc kaszkád (Haar Cascade) osztályozó kulcsfontosságú függvénye. Lényegében a következőket végzi:

1. **Szürkeárnyalatos kép bemenete:** A függvény a szürkeárnyalatos képet használja bemenetként, melyet az előző lépés során alakítottunk át.
2. **Méretezi a képet:** A kaszkád különböző méretű keresőablakokat használ, hogy különböző méretű arcokat is megtaláljon. A detectMultiScale függvény a képet többször méretezi le, így képes kisebb és nagyobb arcok megtalálására is.
3. **A keresőablak mozgatása:** A keresőablak a szürkeárnyalatos képen lépésről lépésre végigmegy. Minden egyes pozíciónál a Haar-jellemzők kiszámításra kerülnek.
4. **Haar-jellemzők kiértékelése:** Minden egyes régióban kiszámításra kerülnek a Haar-jellemzők, ezek a téglalap alakú minták, amelyek sötét és világos pixelek kontrasztján alapulnak. A kaszkád ezeket a jellemzőket vizsgálja, hogy megtalálja az arcnak megfelelő részeket (pl. szem, orr, száj).
5. **A kaszkád szintjei:** A kaszkád egy többlépcsős rendszer. Minden egyes szint egy egyre összetettebb osztályozót tartalmaz. Ha egy régió eléri az egyik szintet, akkor átadásra kerül a következő, ennél is nehezebb osztályozó szintnek.
6. **Arc észlelése:** Ha a régió átjut az összes kaszkádszinten, akkor arcként kerül azonosításra. A detectMultiScale függvény egy listát ad vissza, melyben koordináták találhatók (x, y, w, h), melyek az arcok körüli téglalapot írják le.

**Paraméterek:**

A detectMultiScale függvénynek néhány fontos paramétere van:

* **scaleFactor:** Ez a paraméter meghatározza, hogy mennyire szeretnénk méretezni a képet az egyes kaszkádszintek között. Egy nagyobb scaleFactor gyorsabb felismerést, de kevésbé pontos eredményt jelent.
* **minNeighbors:** Ez a paraméter meghatározza, hányszor kell a kaszkádnak egy régiót arcként azonosítania, mielőtt ténylegesen arcként detektálja. Egy nagyobb minNeighbors szigorúbb kritériumokat állít fel, és segít csökkenteni a hamis pozitív eredményeket.

**Kódban:**

A scaleFactor 1.1 értékre van állítva, az azt jelenti, hogy minden szint között a kép 10%-kal lesz lekicsinyítve. A minNeighbors 5-re van állítva, ez azt jelenti, hogy egy régiót legalább ötször kell arcként azonosítani, mielőtt ténylegesen arcként kerül észlelésre.

ez itt a szürkeárnyalatos konverzió és az arcfelismerés


**Rajzolás:**

* **cv2.rectangle:** Ez a függvény rajzol egy téglalapot a képre. A paraméterei a következők:
  + **kep:** A bemeneti kép, amire rajzolni fogunk.
  + **(x, y):** A téglalap bal felső sarkának koordinátái.
  + **(x+w, y+h):** A téglalap jobb alsó sarkának koordinátái.
  + **(255, 0, 0):** A téglalap színe (kék).
  + **2:** A téglalap vastagsága.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

**II. verzió:  
  
Kép beolvasása a kameráról:**

* A ret, kep = cap.read() sor egy képet kap a kamerafolyamból. A ret egy logikai változó, amely azt jelzi, hogy sikeresen beolvastunk-e egy képet vagy sem a .read() művelettel. A kep változó tartalmazza magát a kép adatait.

Ablak bezárása:

* A cv2.waitKey(1) parancs 1 milliszekundumra vár billentyűleütésre. Ha ezen idő alatt nem történik billentyűleütés, a parancs -1 értéket ad vissza. Ha billentyűleütés történik, a parancs a lenyomott billentyű ASCII kódját adja vissza.
* A & 0xFF bitenkénti ÉS művelet a billentyűkód legfelső 8 bitjét 0-ra állítja. Ez biztosítja, hogy a billentyűkód ASCII érték legyen, még akkor is, ha a billentyűzet nem angol elrendezésű.
* Az ord('q') parancs a 'q' betű ASCII kódját adja vissza, ami 113.

A képen szöveg, Betűtípus, képernyőkép látható

Automatikusan generált leírás

**Kép beolvasása a kameráról:**

A cap.read() parancs beolvassa a következő képet a kamerafolyamból. A parancs két értéket ad vissza:

* ret: Egy logikai változó, amely jelzi, hogy a beolvasás sikeres volt-e. Ha True, akkor a beolvasás sikeres volt, és a kep változó tartalmazza a beolvasott képet. Ha False, akkor a beolvasás sikertelen volt.
* kep: A beolvasott kép adatait tartalmazó tömb.

A képen szöveg, Betűtípus, képernyőkép, tervezés látható

Automatikusan generált leírás

**Memóriafelszabadítás és bezárás:**

**1. cap.release():**

* Felszabadítja a kamerát a program által történő használat alól.
* Biztosítja, hogy más programok is hozzáférhessenek a kamerához.
* Memóriát szabadít fel, amit a kamera foglalt.

**2. cv2.destroyAllWindows():**

* Bezárja az összes OpenCV ablakot, amelyet a program nyitott.
* Felszabadítja a memóriát, amit az ablakok foglaltak.
* Megakadályozza, hogy a program ablakai nyitva maradjanak a háttérben, miután a program befejezte a futást.