

1. feladat

Alakfelismerés macskákra

Közismert, hogy a macskás gif-ek milyen fontos szerepet játszanak/játszottak az Internet fejlődésében. Egy másik mostanában nagyon fontos kérdéskör pedig a számítógépes alak- vagy arcfelismerés. Ez a két megatrend inspirálta a következő feladatot.

Gyakran előfordul, hogy egy kedves ismerősünknek elszöknek a macskái. Azért, hogy gyorsan megtalálja őket, az összes környékbeli ház webkamera felvételein szeretne gyorsan, automatizáltan keresni. Írjunk programot, ami egy ilyen képen megtalálja a macskákat! Elsőként az egyik elveszett cicáról adunk egy képet:



lost-cat.gif

Mivel 7 egyforma macskáról van szó, egy cica képe elegendő lesz. Az egyik szomszéd verandáját mutató képen több olyan cica is felfedezhető, amelyek nagyon hasonlítanak az elveszett macskákra:



cat-search.png

Azt kell kiderítenünk, hogy valóban a keresett macskák vannak-e ezen a képen. Az ilyen típusú feladatokat pl. neurális hálózatokon alapuló algoritmusokkal szokták megoldani, de mi most egy egyszerűbb megközelítést alkalmazunk.

A kereséshez és összehasonlításhoz az elveszett cica képét egy mátrixban tároljuk. Ezután kiszámoljuk ennek a képnek és a verandát mutató kép egyes részleteinek a korrelációját és feljegyezzük, hogy a nagy kép mely részein lesz a korreláció egy bizonyos határértéknél nagyobb.

A feladat megoldásához három dolgot kell megtanulni:

1) Mátrix tárolása egyindexes tömbben

Mátrixok tárolására az a leghatékonyabb, ha sorfolytonos alakban tároljuk az adatokat. Ez azt jelenti, hogy ha $cols$ a mátrix oszlopainak száma (azaz a mátrix szélessége), akkor a hagyományosan a_{ij} elemet $a[i * cols + j]$ módon tudjuk elérni (akár kiolvasás, akár értékadás céljából). Ehhez $rows * cols$ hosszúságú tömböt kell létrehozni ($rows$ a sorok száma). (Lásd az előadás anyagában).

E feladatban képeket fogunk mátrixban tárolni, amelyeket .dat kiterjesztésű fájlokban adunk meg. A méretük (szélesség x magasság) kiolvasható a fájlnevből. Ezekben (a text fájlok és a python képrekezelési hagyományát követve) a kezdő adat a kép bal felső sarkának felel meg. (Érdeemes megjegyezni, hogy ezzel ellentétben pl. a bmp formátumok a bal alsó sarokkal kezdenek.)

2) Két adatsor korrelációja

Korreláció számolásakor azt vizsgáljuk, hogy a két adatsor mennyire váltakozik együtt. Ezt úgy tudjuk megállapítani, hogy az átlagtól való eltéréseiket $(x - \bar{x})(y - \bar{y})$ alakban összeszorozzuk. (Itt \bar{x} és \bar{y} az átlagot jelöli.) Ezt a szorzatot azután átlagoljuk és viszonyítjuk ahhoz, hogy a két adatsor mennyire ingadozik (leosztjuk a szórásokkal):

$$\text{corr}(x, y) = \overline{(x - \bar{x})(y - \bar{y})} / (\sigma_x \sigma_y)$$

Így az x vagy y adatsort magával korreláltatva éppen 1-et kapunk.

A mennyiségek indexes alakját felhasználva néhány átalakítással olyan alakhoz jutunk, amit technikailag könnyebb kiszámolni:

$$\text{corr}(x, y) = \frac{\overline{xy} - \bar{x} \cdot \bar{y}}{\sqrt{(\overline{x^2} - \bar{x}^2)} \sqrt{(\overline{y^2} - \bar{y}^2)}} ,$$

ahol az átlagokat egymáshoz hasonlóan az egyszerű összegekből kapjuk:

$$\bar{x} = (\sum_{i=1}^n x_i) / n , \quad \bar{y} = (\sum_{i=1}^n y_i) / n , \quad \overline{x^2} = (\sum_{i=1}^n x_i^2) / n , \quad \overline{y^2} = (\sum_{i=1}^n y_i^2) / n , \\ \overline{xy} = (\sum_{i=1}^n x_i y_i) / n ,$$

és n az adatsorok hossza.

Ugyanez kétindexes adatkészletünk esetében úgy számolható, hogy mindkét indexben összegzünk, azaz $\sum_{i=1}^n$ helyett $\sum_{i=1}^n \sum_{j=1}^m$ írandó, ahol n és m a mátrix sorainak ill. oszlopainak száma, valamint a mennyiségek i indexe helyett ij indexpár használandó.

Ezek után

Már elvileg minden adott ahhoz, hogy egy macskák alakfelismerésére alkalmas programot írjunk.

Az alábbi a)--g) pontok szerinti lépéseket ajánlott követni, és a működő változatot main-a.c, main-b.c stb. néven elmenteni, mielőtt továbblépünk a következő pontra. Mivel az egyes pontok a korábbi bővítésével adódnak, csak az utolsó működő változatot kell beadni, azt is az előbbieik szerint nevezve el.

A program a következő parancssori argumentumokat olvassa be, pontosan ebben a sorrendben:

első_fájl_neve szélessége magassága második_fájl_neve szélessége magassága

Az adatfájlok megtalálhatóak a halnum.public/feladat1 könyvtárban.

A korrelációt számoló függvény tesztelésére használható kisméretű adatfájlok: smiley10x11.dat, sad10x11.dat, tree10x11.dat, smileybig24x20.dat.

Megtekintésre gif formátumban: smiley, sad, tree, smileybig

A feladathoz tartozó adatfájlok: Elveszett cica (lost-cat80x57.dat) , Hol van a mi macskánk? (cat-search640x360.dat).

a)

Írjunk függvényt, ami beolvas és az 1)-es pont szerint eltárol egy adatfájlt!

Ezután ellenőrzésül írassuk is ki az adatokat egy másik függvénnyel olymódon, hogy az adatok sorokra vannak tördelve! De ez a kiírás csak akkor történjen meg, ha a kép méretei kisebbek egy határnál (pl. 20), hiszen a nagy mátrixok már amúgy is nehezen áttekinthetőek.

Tesztelésre használjuk ezeket a fájlokat: smiley10x11.dat, sad10x11.dat, tree10x11.dat, smiley10x11.dat

Megtekintésre mindegyik elérhető gif formátumban is. (A)

b)

Írjunk függvényt, ami kiszámolja a két megadott kép közötti korrelációt, megadva neki a két beolvasott tömböt, és azok méretét! Egyelőre feltételezhetjük, hogy a tömbök egyforma méretűek. A bevezetőben szereplő képleteket használva érdemes az összes összegzést egyetlen kettős ciklusban elvégezni. Írassuk is ki a korreláció értékét! De a kis képek esetére vonatkozó kiírás is maradjon meg!

Teszteljük programunkat a smiley—smiley, smiley—sad és smiley—tree párosításban! Ez úgy értendő, hogy a parancsargumentumokban e fájlok neveit és méreteit adjuk meg. Azt várjuk, hogy azonos képeknél 1, hasonló képeknél 1-hez közeli, a nagyon eltérőknél 0-hoz közeli értéket kapunk. (A)

c)

Bővítsük a korreláció függvényt úgy, hogy a két kép mérete különböző is lehessen! A **másodikként megadott kép** vízszintesen is és függőlegesen is nagyobb vagy egyenlő méretű lehet, mint az első.

A korrelációs számolás a közös részen fusson le, a nagyobb kép kilógó részét figyelmen kívül hagyva! Gondoljuk meg, hogy ehhez a két kép méretei közül melyeket kell átadni a függvénynek! Ezt tesztelhetjük a smiley figurát tartalmazó megnövelt kép segítségével: a sad—smileybig és sad—smiley összehasonlításoknak azonos eredményt kell adniuk. (A)

d)

Az egyik keresett cica képe a lost-cat80x57.dat fájlban található. Keressük meg legalább az egyik ilyen cicát a cat-search640x360.dat képen! (E képek megnézhetőek gif formátumban.)

Megnyitva az adatfájlokat, észrevehetjük, hogy a számértékek nem pontosan olyan elrendezésben szerepelnek bennük, mint a korábban használt smiley, sad, stb. fájlokban. Gondoljuk át, hogy ez miatt át kell-e írni a programban a beolvasó függvényt?

A keresés előtt hagyjuk lefutni az eddigi programrészt, ami kiírja a képadatokat (kis kép esetén) és a korrelációt! Így nem kell külön beadni a c) résznek megfelelő programot.

A keresést úgy tehetjük meg, hogy a nagyobb képre képzeletben rárajzolunk a kicsivel azonos méretű téglalapot, és ezt a téglalapot a nagy képben néhány (javasolt:4-5) pixel lépésekben vízszintesen és függőlegesen is végigléptetjük. Minden pozícióban hasonlítsuk össze a kijelölt képrészletet a kis képpel! Ezt pl megvalósíthatjuk úgy, hogy c) pontban megírt függvénynek megadhatunk egy eltolás értéket is, amit a számolásban figyelembe vessz.

Semmikép se pakoljuk át a kiválasztott rész értékeit egy másik tömbbe az összehasonlítás kedvéért, mert az időigényes!

Vigyázzunk, hogy csak olyan eltolást adjunk meg, hogy a függvény érvényes indexeket címezzen meg a két mátrixban!

Amennyiben a két adattömb közötti korreláció értéke egy határ feletti (javasolt határ: 0.4), akkor írjuk ki a téglalap pozícióját és a korreláció értékét! Reméljük, hogy ha az egyik cicát sikerül megtalálni a nagy képen és azután hazavisszük, a többi úgyis jön vele! (T)

e)

Pontosítsuk a keresést! Ehhez a d) pont szerinti léptetés során tároljuk el a korreláció maximumának pozícióját és értékét, a végén pedig írjuk ki azt! Utána az így talált maximum környékén már 1 pixeles lépéseket végezve keressük meg a maximumot pontosabban, és írjuk ki! (T)

f) Szorgalmi:

A e) pontban egy cicát találtunk, haza vittük, de mégse jött vele a többi. Azokat is meg kell keresni! A cicák a nagy képen lehetnek messzebb, mint a portrén, így kisebbnek látszhatnak. Ezért terjesszük ki a keresést olyan módon, hogy a korreláció függvényünkéből csinálunk egy összehasonlító függvényt, aminek kicsinyítési arányt is meg lehet adni. A kicsinyítési arányt lépésenként (pl. 10%) csökkentve kereshetjük a többi cicát. Építsük be a függőleges tengelyre való tükrözés lehetőségét is!