

### 3. 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.

Egy kedves ismerősünknek elszöktek a cicái. Ez gyakran előfordul, ezért automatizáljuk a keresést. Írjunk programot, ami a környező házaknál elhelyezett webcamok képein megtalálja őket! Elsőként az egyik elveszett cicáról adunk egy képet:



lost-cat.gif

Mivel 4 egyforma macskáról van szó, egy cica képe elegendő lesz. Az egyik szomszéd verandáját mutató képen több olyan cicát is felfedezünk, amely nagyon hasonlít a mi elveszett macskáinkra:



cat-search.gif

Azt kell kiderítenünk, hogy valóban a mi macskáink vannak-e ezen a képen. Az ilyen típusú feladatokat pl. neurális hálózatokkal 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 két dolgot kell megtanulni:**

## 1) Mátrix tárolása egyindexes tömbben

Mátrixok tárolása c-ben a leghatékonyabb sorfolytonos (v. esetleg oszlopfolytonos) alakban. 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). Bővebben ld. az előadás anyagban: Mátrixok és mutatók 10000\_AK.pdf

E feladatban képeket fogunk mátrixban tárolni, amelyeket a 2-es beadandó feladathoz hasonló formátumban, .dat kiterjesztésű fájlokban is megadunk. A méretük (szélesség x magasság) kiolvasható a fájlnevből. Ezekben (a text fájlok és a python képezelési hagyományát követve) a kezdő adat a kép bal felső sarkának felel meg. (Érdemes 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 az átlagtól való eltéréseiket  $(x - \bar{x})(y - \bar{y})$  alakban összeszorozva tudjuk megállapítani. (Itt  $\bar{x}$  az átlagolás jele.) Ezt a szorzatot azután átlagoljuk és viszonyítjuk ahhoz, hogy a két adatsor mennyire ingadozik (populáció szórásaik szorzatához):

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

Így az  $x$  vagy  $y$  adatsort magával korreláltatva 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  $i, j$  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. Codeblocks esetén ekkor a projekt direktorin kívülre tanácsos áttenni. Mivel az egyes pontok a korábbi bővítésével adódnak, csak az utolsó működő változatot kell beadni, azt is az előbbiek szerint nevezve el.

A program a következő parancssori argumentumokat olvassa be:

első\_fájl\_neve szélessége magassága második\_fájl\_neve szélessége magassága

**a)**

Írjunk függvényt, ami beolvas és a fentiek szerint eltárol egy adatfájlt. Ezt futtassuk le a két megadott adatfájllal, és utána ellenőrzésül írassuk is ki külön ciklusban, sorokra tördelve! (Ehhez kapcsolódó példa szerepel az előadás fóliái között). Javasolt a programot úgy megírni, hogy a kiírás akkor történjen meg, ha a kép méretei kisebbek egy határnál (pl. 20), hiszen a nagy mátrixok már amúgy sem áttekinthetőek. Tesztelésre használjuk ezeket a fájlokat: smiley10x11.dat, sad10x11.dat, tree10x11.dat

Megtekintésre mindegyik elérhető png formátumban. (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 végezni el. Írassuk is ki a korreláció értékét és teszteljük a smiley-smiley, smiley-sad és smiley-tree párosításban! 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 második 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.)

Ezt úgy tegyük, hogy képzeletben a nagyobb képre rárajzolunk a kicsivel azonos méretű téglalapot, és ezt a téglalapot a nagy képen néhány (javasolt:5) pixel lépésekben vízszintesen és függőlegesen is végigléptetjük. (Mivel a cicák nem tudnak magasra ugrani, tulajdonképpen elég a kép alsó felében keresni.) Minden pozícióban hasonlítsuk össze a kijelölt képrészletet a kis képpel! Ezt kétféleképp próbálhatjuk megvalósítani:

- Módosítjuk az c) pontban írt függvényt úgy, hogy megadhassunk neki egy eltolást is, és azt a számolásban figyelembe vesszük.
- Vagy a függvény módosítása nélkül, a függvény meghívásakor a kiválasztott részlet pozíciójának megfelelően módosítjuk a függvénynek átadott pointert.

Semmiképp 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 az 1. feladat minimumkereséséhez hasonlóan 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 d) pontban egy cicát találtunk, 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 függvényünkől 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!

### **g) Szorgalmi:**

Egy cica még mindig nincs meg: építsünk be forgatást is, pl. 8-10 fokos lépésekben,  $\pm 45$  fokos szögig! Gyorsítás kedvéért próbálhatjuk a következőket:

- a korreláció kiszámolása során is kihagyunk indexeket, pl. csak 2-esével lépegetve összegzünk
- eltárolhatjuk az eredményeket és utólag megnézzük a néhány legjobb egyezést.