

# Digitális képfeldolgozás gyakorlat 2023. tavaszi félév

## 2. félévközi beadandó feladat

### Feladatkiírás

Készítsen Python + OpenCV használatával egy programot, amely az alábbi funkciókat látja el.

#### Kártya szegmentálás színek alapján

A program a példatárban található `FCards_02_rs.jpg` képre működjön.

Detektálja a képen a fehér, fekete és vörös területeket színcsatornánkénti küszöböléses technikával és aritmetikai-logikai műveletekkel kombináltan. Ezek a részeredmények bináris képek legyenek.

- Vörös akkor, ha ennek megfelelő B, G és R tartományba esnek a csatornák értékei. [0,5 pont]
- Fehér akkor, ha ennek megfelelő B, G és R tartományba esnek a csatornák értékei és a B, G és R csatornák értékei között a maximális eltérés nem nagyobb 50 értéknél. [0,5 pont]
- Fekete akkor, ha ennek megfelelő B, G és R tartományba esnek a csatornák értékei és a B, G és R csatornák értékei között a maximális eltérés nem nagyobb 25 értéknél. [0,5 pont]

Megjegyzések: A kép alacsony felbontása és a mintavételezési hibái miatt a vörös és fekete számok és a betűk nem lesznek jól szegmentálhatók a kártyalapokon, azokat kihagyhatjuk. Ezzel az egyszerű módszerrel tökéletes eredményt nem lehet elérni, de törekedjenek arra, hogy minél jobb fedést érjenek el a nagyobb területű részekben. Az asztallap barna színéből ne kerüljön szegmentálásra terület.

Hozzon létre az eredeti képpel megegyező méretű új BGR képet, és rajzolja bele a detektált területeket az alábbi módon.

- Ahol fehér terület van, oda fehér szín [255, 255, 255] kerüljön. [0,125 pont]
- Ahol vörös terület van, oda vörös szín [0, 0, 255] kerüljön. [0,125 pont]
- Ahol fekete terület van, oda fekete szín [0, 0, 0] kerüljön. [0,125 pont]
- Minden más képpont legyen sárga színű [0, 255, 255]. [0,125 pont]

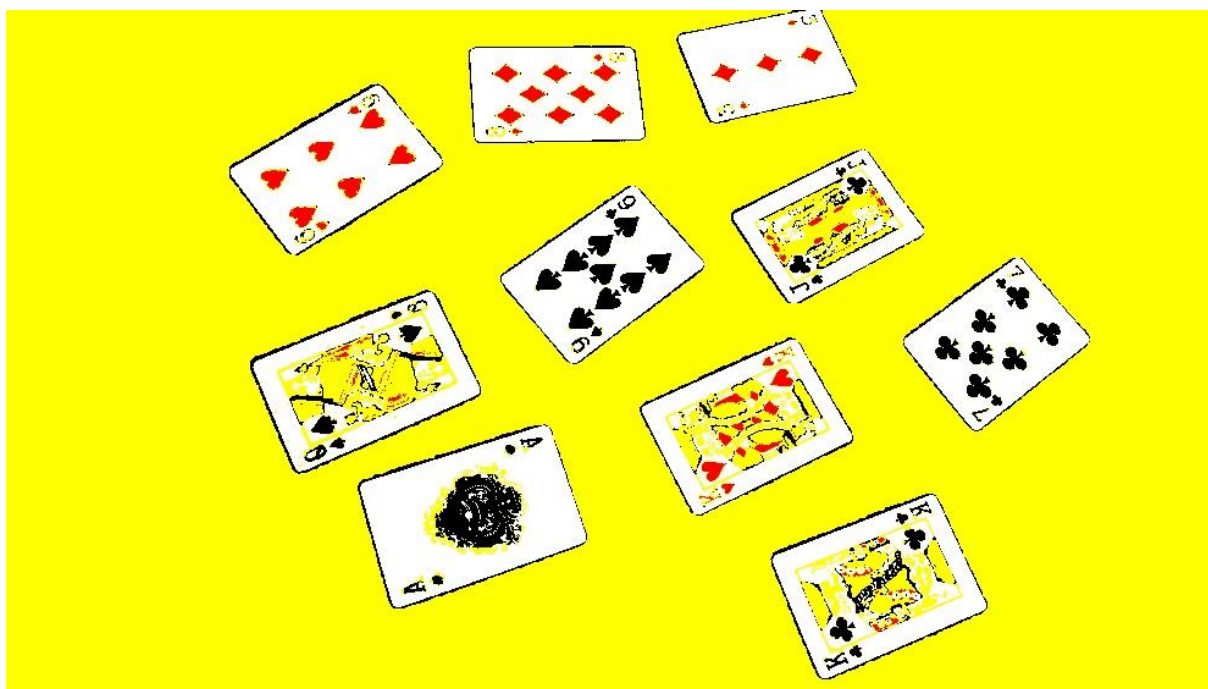
Az eredeti beolvasott és az eredményképet jelenítse meg külön ablakban, ami billentyű lenyomására tűnjön csak el.

Törekedjen hatékony kódot írni OpenCV és Numpy műveletek használatával. Amennyiben egymásba ágyazott `for` ciklusokkal való képbejárást használ, akkor maximum 1,5 pont szerezhető!

#### Ötletek a megvalósításhoz

- Érdemes a bemeneti BGR képet csatornákra bontani (`cv2.split()`).
- A csatornák között aritmetikai/logikai függvényekkel meghatározni a képpontonkénti maximális eltérés mértékét. Pl. [100, 130, 110] képpont érték esetén a maximális eltérés értéke 30.
- A maximális eltérés mértékét küszöbölni a megfelelő értékkel.
- Ezeket a bináris képeket felhasználni a fekete és fehér területek meghatározásához a színcsatorna tartomány szerinti küszöböléssel kombinálva.

## Egy lehetséges kimeneti eredmény



## Beadási információk

Az elkészült programot az alábbi módon adja be.

- Az elkészült Python kód neve legyen a hallgató neve aláhúzás neptunkód pont py (pl. [KaracsonySzilveszter\\_NepKodXX.py](#)).
- Az állományok neve nem tartalmazhat ékezetes betűket, speciális karaktereket és szóközőket, csak az angol ábécé betűit, számokat és az aláhúzás karaktert.
- A két állományt tömörítse be egy ZIP vagy RAR állományba és töltsse fel a megadott határidőig a Coospace-re. (A beküldött megoldásokat szkript csomagolja ki, ezeket találja meg és csomagolja ki.)
- A megoldás többször is beadható, de csak a legutolsó kerül értékelésre! 1 darab ZIP/RAR fájlt töltsön fel. Ha szöveges információt is csatolni szeretne a megoldáshoz, azt egy [txt](#) fájlban helyezze el a tömörített csomagban, vagy a program elején helyezze el kommentár blokkban.

## Értékelés

A beadásra maximum 2 pontot lehet szerezni. A pontok elosztása a részfeladatok leírásánál olvasható.

Ha a program nem működik, akkor automatikusan 0 pontot kap a hallgató. Forráskód nélkül, önmagában csak az elkészült rajz szintén nem értékelhető.

A határidőn túli (határidő + maximum 24 óra) de a végső határidőn belüli leadásért a pontszám fele adható.

A végső határidő után (határidő + 24 óra) és megjegyzésben feltöltött megoldásért nem jár pont.