

Tartalomjegyzék

A feladat leírása	1
Az elkészített megoldás működése	1
Referencia képek feldolgozása.....	2
A beolvasott képfájlok.....	2
Szürkeárnyalatossá alakítás után	2
A binarizált képek	3
A maszkolt képek.....	3
SIFT eredménye.....	4
A vizsgált kép feldolgozása	4
Az eredeti és a szürkeárnyalatos képek	4
A binarizált és a maszkolt képek	5
SIFT kulcspontok	6
A SIFT-hez használt paraméterek.....	6
A SIFT adatok feldolgozása	7
Perspektíva transzformációk	7
A kapott transzformációk feldolgozása	8
Végleges eredmény képe	9
Tesztelés más háttér előtt.....	10
Tesztelés más paklival.....	11
Felismert képek aránya	11
Hibák.....	12
Kis mértékben átfedő lapok.....	13
Eredeti háttér előtt, amihez a paraméterek igazítva lettek	13
Más háttér előtt.....	14
Más kártyapaklival	14

A feladat leírása

Készítsen egy alkalmazást, amely az asztalra (sötét felületre) kiszórt, egymást lényegileg nem fedő kártyalapot (nevét) tud felismerni (használhat homográfiát, mintaegyezés vizsgálatot, vagy más ismertetett módszert)!

Az elkészített megoldás működése

A feladat megoldásához a SIFT algoritmust használtam. Ehhez egyesével minden az 52 kártyáról készítettem egy képet, majd ezeken a kártyaképeken futtattam a SIFT algoritmust.

A referencia képeken kívül készítettem 20 darab képet kék háttérre kihelyezett kártyákról. Ezekben a képeken több kártya szerepel, különböző módon forgatva és elhelyezve, nemely esetben egymást átfedően. Egy futtatás során egy ilyen képet beolvas a program és erre is lefut a SIFT algoritmus.

A képek feldolgozásához Python programnyelvet és az OpenCV-t használtam. Mind a referencia, mind a vizsgált képen a SIFT futtatása előtt a képeket szürkeárnyalatosra alakítottam, majd bináris küszöbölist végeztem a szürkeárnyalatos képen, végül pedig a binárisan küszöbölt képpel maszkoltam a szürkeárnyalatos képet. Ennek eredményeként a feldolgozandó képen a háttér egyenletesen fekete lett a kártyák képei ezért jelentősen kiemelkednek, de az intenzív részek részletei megmaradtak. A SIFT algoritmus az így előkészített képen futott le.

Referencia képek feldolgozása

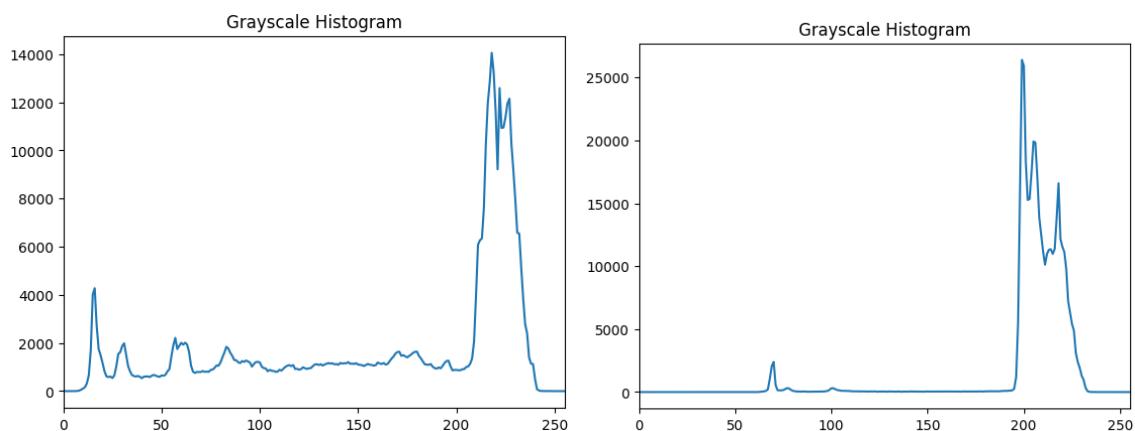
A beolvasott képfájlok



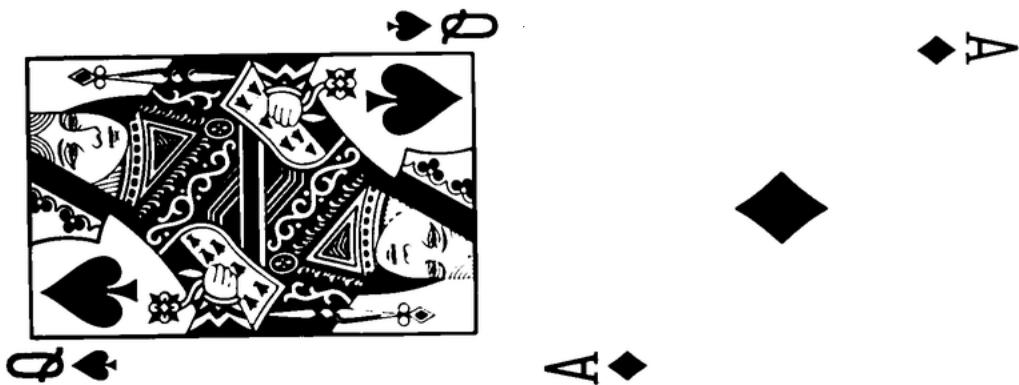
Szürkeárnyalatossá alakítás után



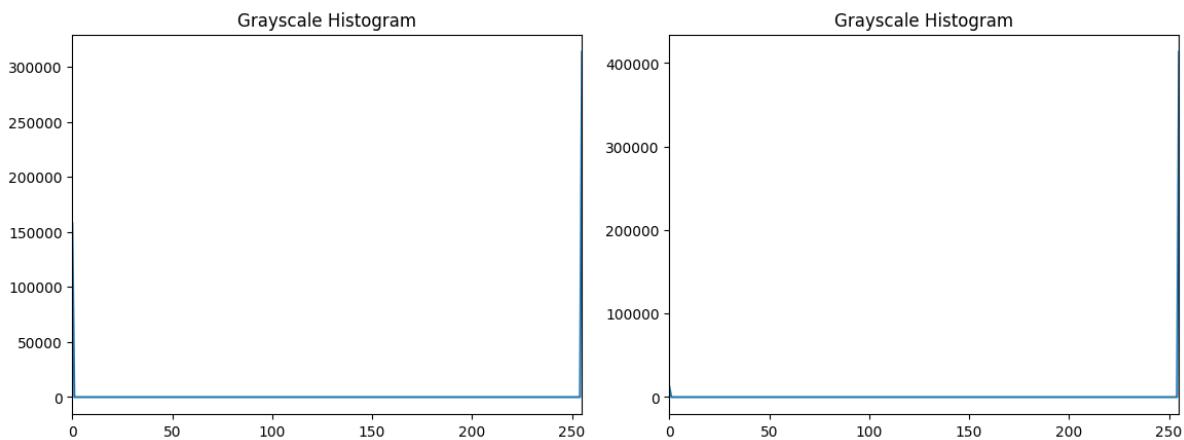
A szürkeárnyalatos képek hisztogramja



A binarizált képek



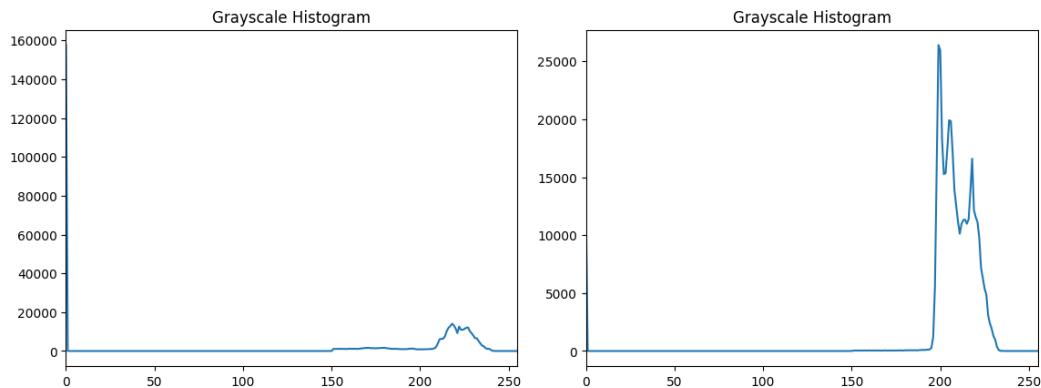
A binarizált képek hisztorogramja



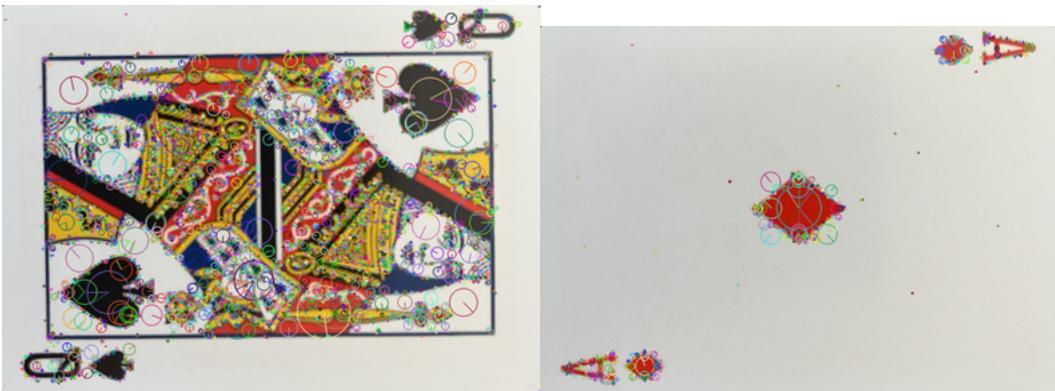
A maszkolt képek



A maszkolt képek hisztorogramja



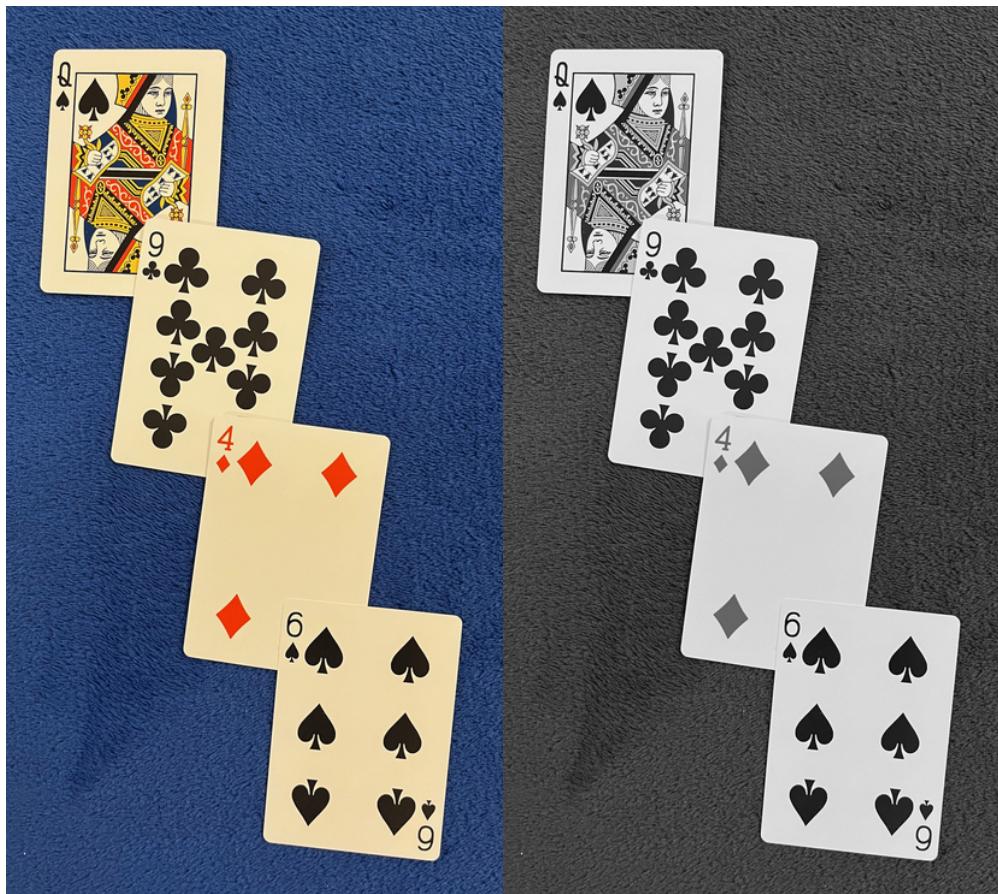
SIFT eredménye



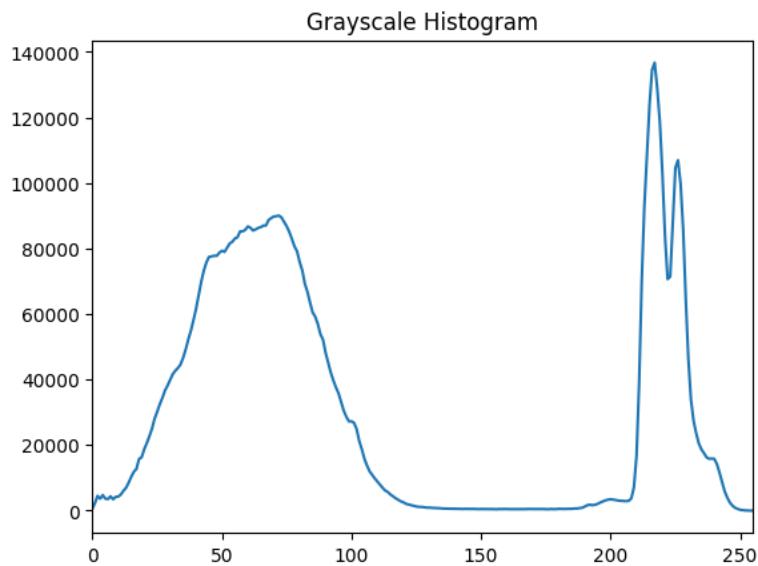
A referencia képek esetében az okozta a problémát, hogy a SIFT algoritmus a J, Q, K lapokon, mivel azok részletgazdagok, vagy nagyon sok kulcspontot talált vagy a kevésbé részletgazdag lapokon, mint az A lapok, nagyon keveset. A túl sok kulcspont a futásidő miatt okozott problémát, mert így több percig tartott egy kép feldolgozása a kulcspontok egyeztetése során. A túl kevés kulcspont azért okozott problémát, mert a kulcspontok egyeztetésekor így nem születtek találatok. A fenti két mintaképen a Q esetében 5218 kulcs pont van használatban a végső paraméterekkel, míg az A esetében 448.

A vizsgált kép feldolgozása

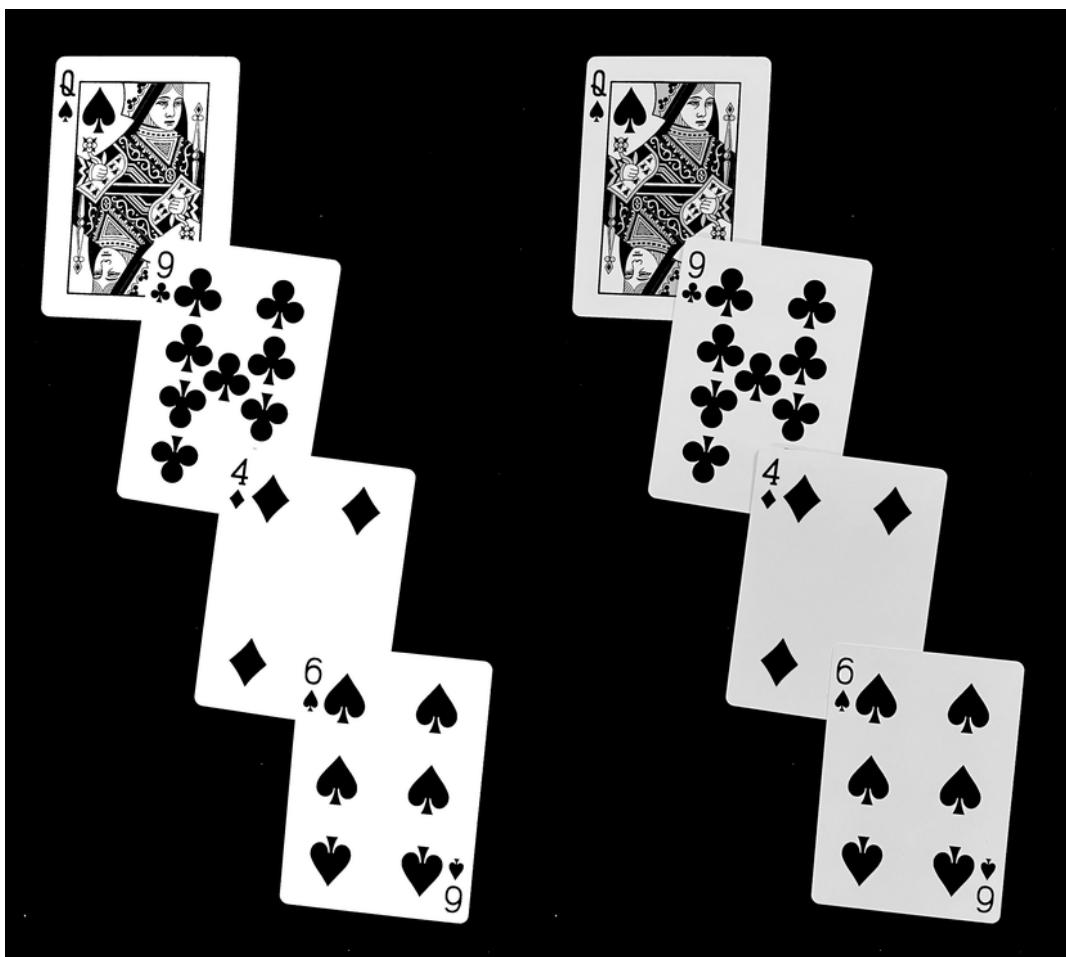
Az eredeti és a szürkeárnyalatos képek



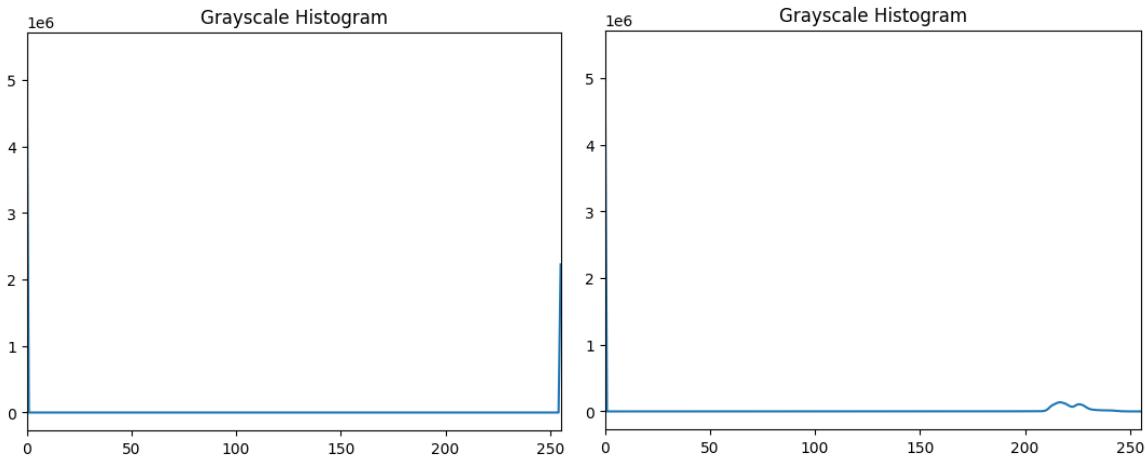
A szürkeárnyalatos kép hisztogramja



A binarizált és a maszkolt képek



A binarizált és a maszkolt képek hisztogramja



SIFT kulcspontok



Ezen a képen a SIFT algoritmus a végleges paraméterekkel 22602 kulcspontot talált.

A SIFT-hez használt paraméterek

Az alábbi paramétereket használtam a SIFT algoritmust:

- contrastThreshold=0.01 – ez jelentősen növelte a detektált pontok számát, de érzékeny lett a zajra
- edgeThreshold=15 – több előre eső kulcspontot talált így meg, ami a kevés mintázattal rendelkező lapok miatt volt szükséges
- nOctaveLayers=11 – segített több ép pontot találni, de számítási igényt nagyon megnövelte
- sigma=1.75 – ez gyengítheti a nagyon finom részletek megtalálását, de az csak a J, Q, K lapoknál van, cserébe csökkent a zajos részeken a pontok száma

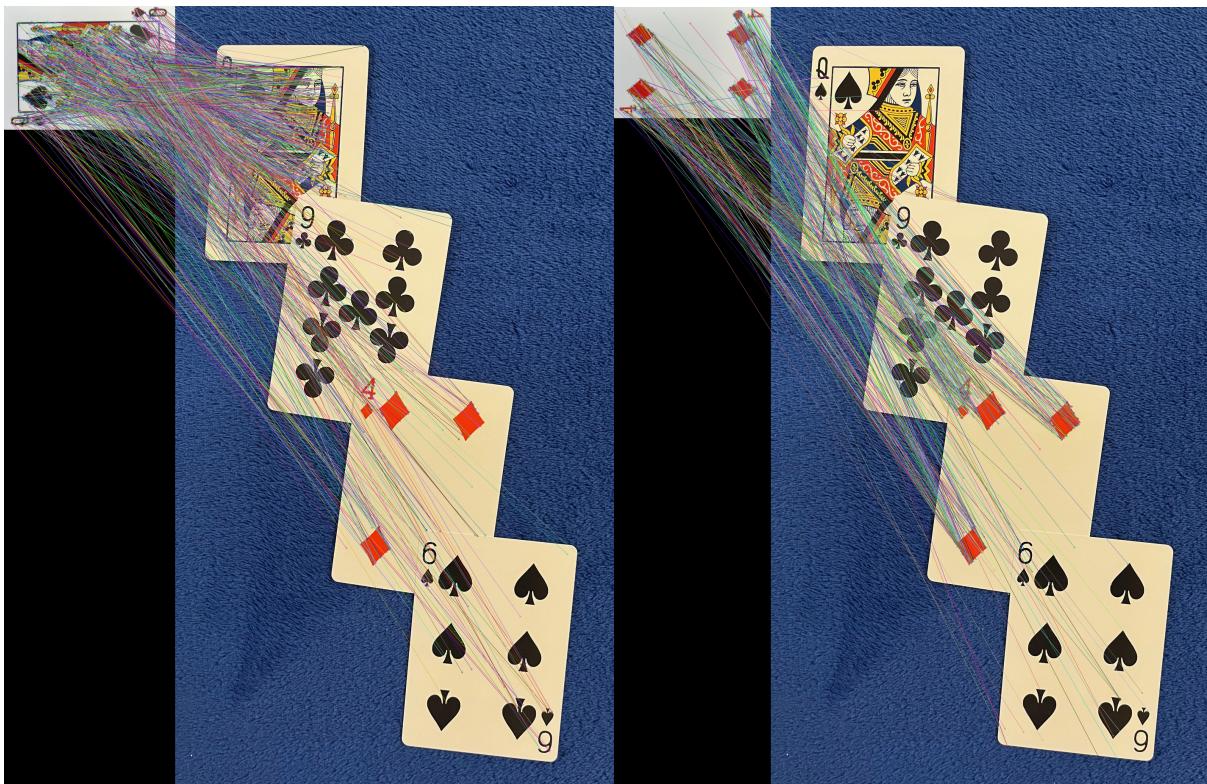
Ezeket a paramétereket főleg próbálkozásos alapon állítottam be, egy ciklusban futtattam a lehetséges értékeket és ahol a legtöbb lapot megtalálta azt a paraméterezeit hagytam meg.

A SIFT adatok feldolgozása

Miután kinyertem az adatokat a referencia képekből és a feldolgozandó képekből is, minden egyes referencia képet megpróbálok megkeresni a feldolgozandó képen a leíró adatok alapján. A program először létrehoz egy brute force illesztőt, amely az euklideszi távolságot használja a leíróvektorok összehasonlítására és nem követeli meg, hogy a párosítás minden két irányban kölcsönösen a legjobb legyen. Ezután a matcher minden egyes referencia-leíróhoz megkeresi a keresett képen belüli 2 legközelebbi szomszédot, vagyis két legkisebb távolságú párt rendel hozzá. Az eredmény egy lista, amely minden referencia-leíróhoz két lehetséges egyezést ad vissza, így előkészítve az adatot a későbbi Lowe-arány-teszthez és geometriai szűréshez. Az arány tesztnél a 0.9-es arányt választottam, mert ez jobban működött a tesztek alapján, mint az alacsonyabbak. Ez természetesen több kevésbé jó találatot adott vissza, de a későbbiekben ezek szűrésre kerültek egy saját módszerrel illetve a homográfia során is jelentősen szűrve lettek.

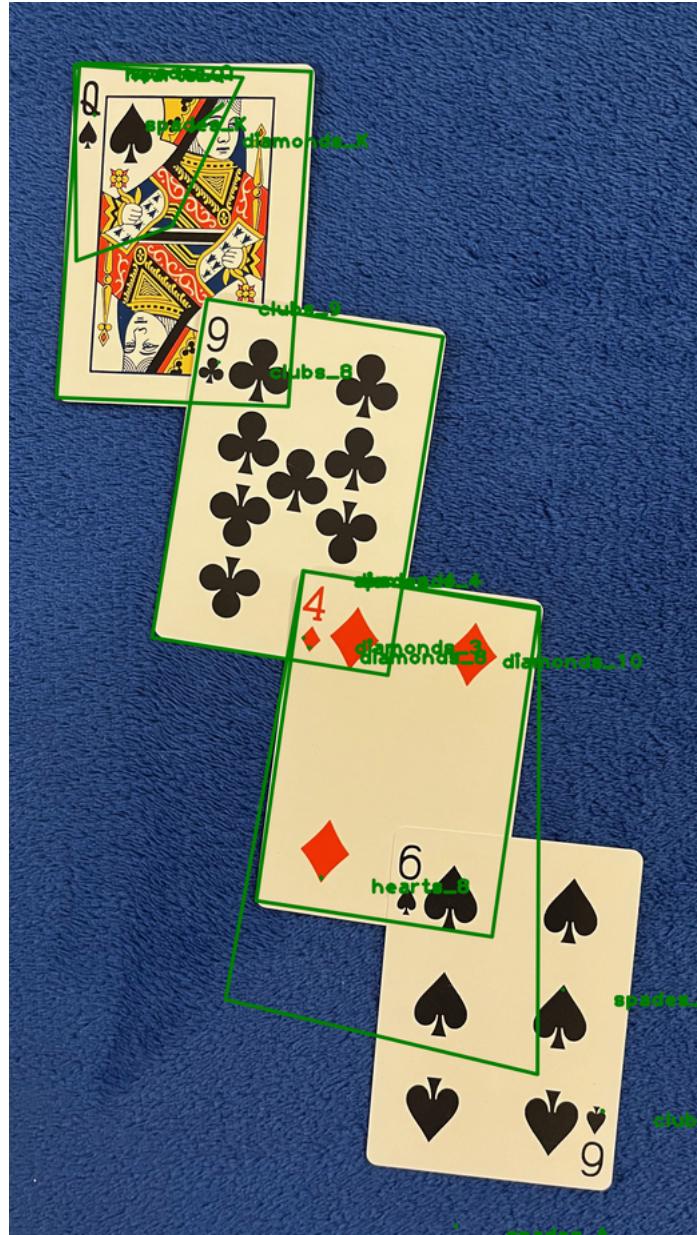
Ezt követően homográfiával megpróbáltam megtalálni a megfelelő transzformációt. Itt három féle módszerrel kerestem a legmegfelelőbbet (cv2.estimateAffine2D és cv2.findHomography RANSAC majd USAC_ACCURATE metódussal) és a legjobb eredményt tartottam meg. Az így kapott mátrix-al végeztem el a perspektíva transzformációt.

Perspektíva transzformációk



A kapott transzformációk feldolgozása

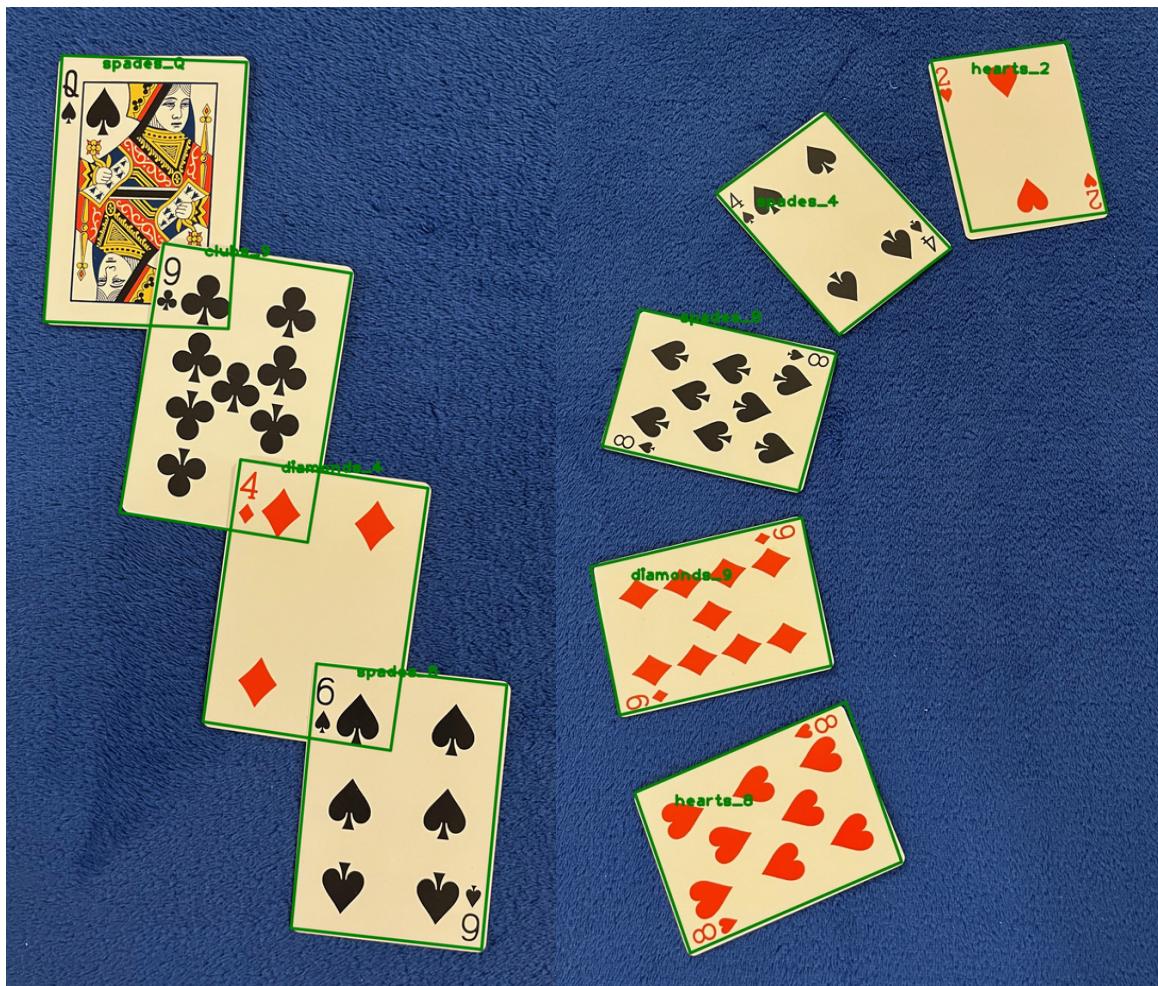
Az így kapott eredmények még sok hibás adatot tartalmaztak amint a az lenti képen látható.



Ezeket a találatokat egyesével feldolgoztam és eltávolítottam azokat, amik általában hibás adatokra utaltak, mint például a nem konvex találatokat, a nagyon kis területet kijelölő találatokat, a képernyőn kívülre eső találatokat.

Ezek után a találatok középpontja alapján megpróbáltam meghatározni, hogy mely találatok azok, amelyek ugyan arra a lapra vonatkoznak. Ezek közül azokat tartottam csak meg, amelyek a legtöbb inline-rel rendelkeztek. Ezek alapján alakult ki egy végleges eredmény.

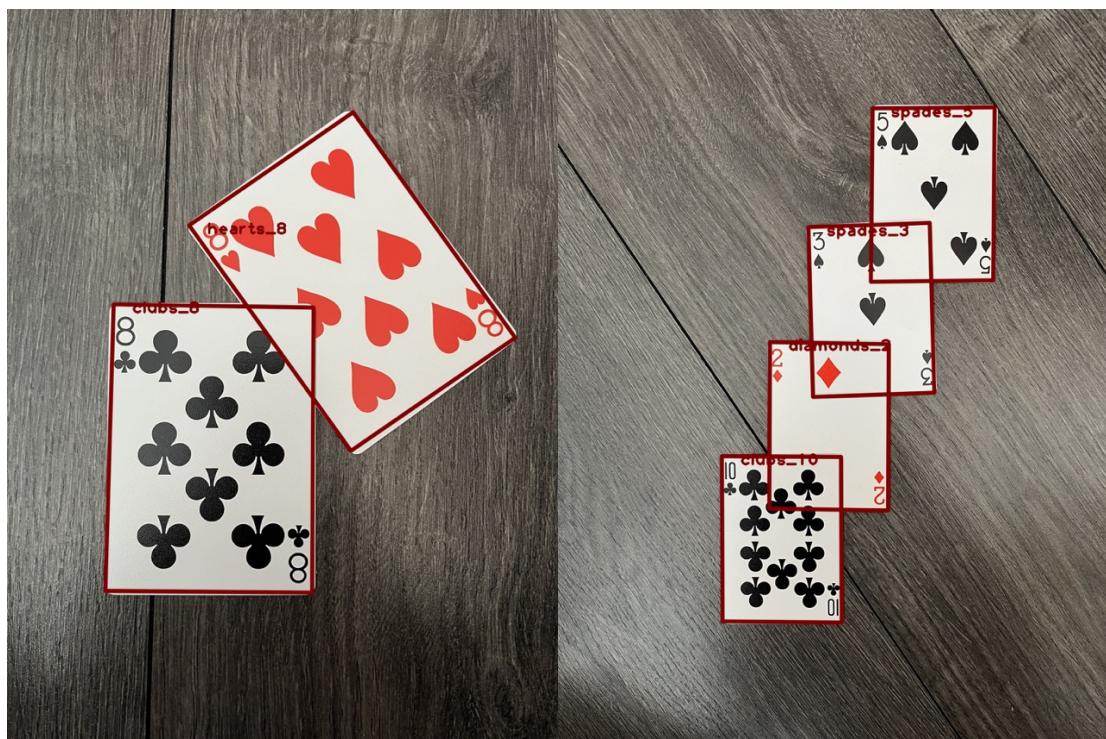
Végleges eredmény képe



A kapott eredmény a fenti képen is látszik, hogy nem tökéletes, mivel a 6-os lapot nem sikerült felismernie, azt eltalálta, hogy pikk, de a 6-os helyett 8-as lappal azonosította. Azonban a kis mértékű átfedést kezeli, meg tudja egymástól különböztetni a lapokat és a különböző módon forgatott lapokat is kezeli.

Tesztelés más háttér előtt

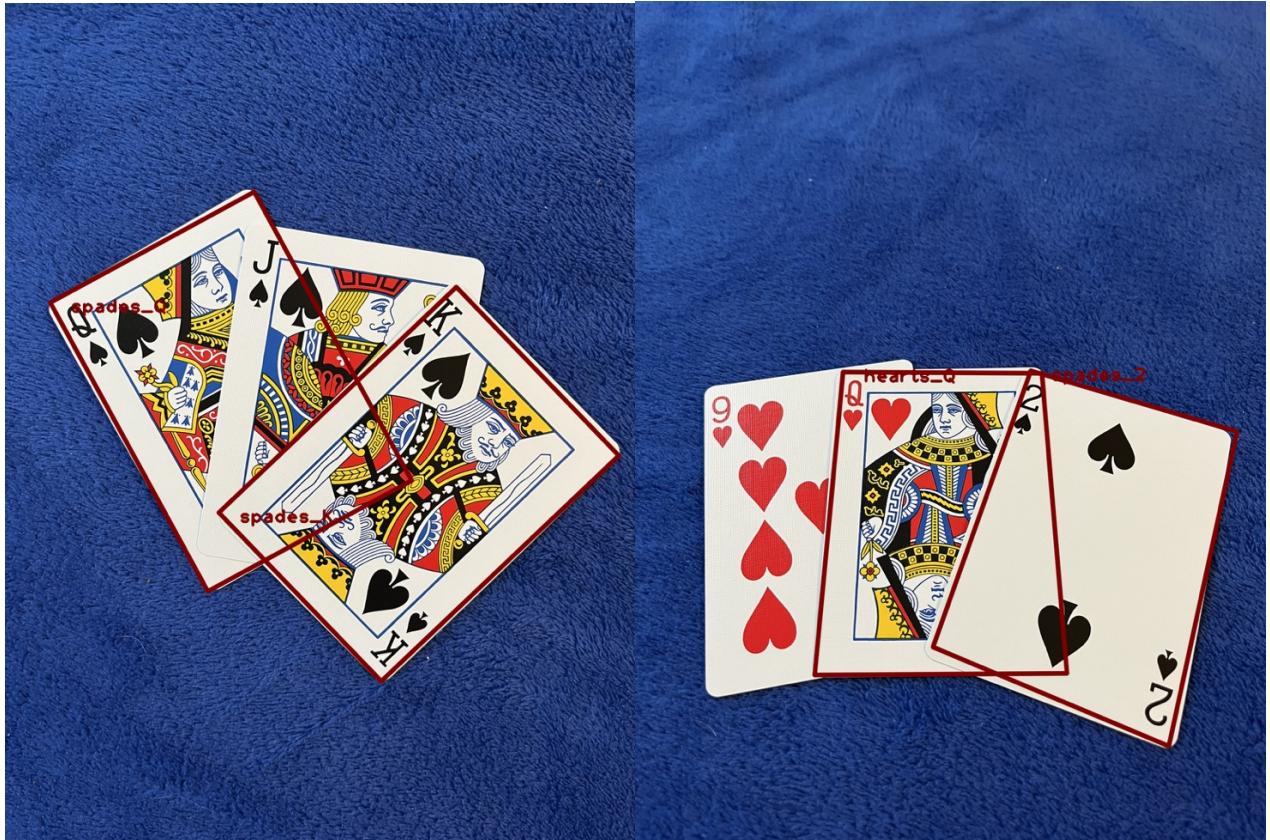
Mivel az algoritmus paramétereit a kék háttérrel készült képekkel kísérleteztem ki és állítottam be, ezért készítettem újabb képeket más háttér előtt. Ezek abban jelentősen különböznek a korábbi képektől, hogy a háttér mintázata miatt ott is sok kulcspontot talált a SIFT algoritmus.



Tesztelés más paklival

Készítettem további képeket egy másik kártya paklival, amelyen a mintázatok kis mértékben természetesen eltérnek az eredeti, referenciaként szolgáló kártyacsomaghoz képest.

Ennek eredménye sokkal rosszabb lett, ahogy az várható is volt, kevesebb lapot tudott jól felismerni.



Felismert képek aránya

A kék háttér előtt készült 20 képen összesen 117 lap szerepelt. Ebből a program 10-et egyáltalán nem talált meg, 6-ot megtalált, de rossz lapnak ismert fel, és talált egy olyan lapot, ahol valójában nem volt lap.

A teszteket elvégeztem mintás szürke háttér előtt is, ott 20 képen szintén 117 lap szerepel összesen, amiből 13-at nem ismert fel, és 6-ot hibásan ismert fel.

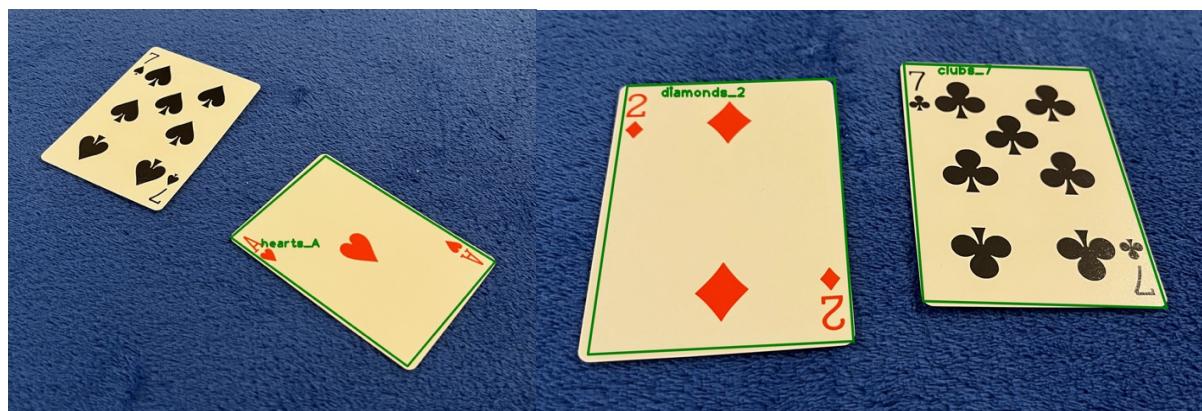
A más paklival végzett testelés során a 15 képen megjelenő 52 labból 17-et nem talált meg, 1-et megtalált, de rosszul ismert fel és 1 helyen lapot jelzett, amely valójában nem volt kártyalap.

Hibák

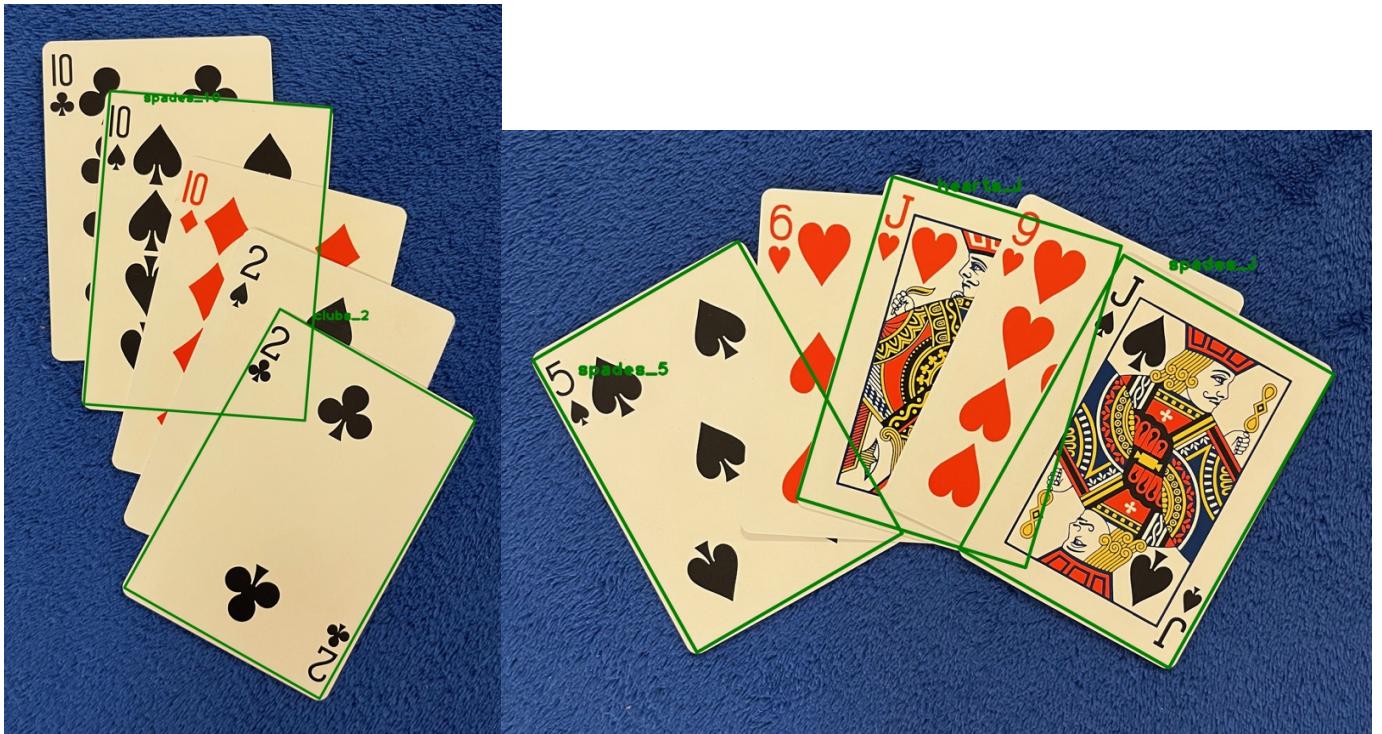
Itt a 6-os és a 9-es lapot nem sikerült megtalálnia a programnak. Más SIFT paraméterekkel ezeket is megtalálta, de akkor a többi képen született rosszabb eredmény.



Itt a perspektíva bizonyos esetben okozott problémát. Például itt az első képen nem találta meg a 7-est.



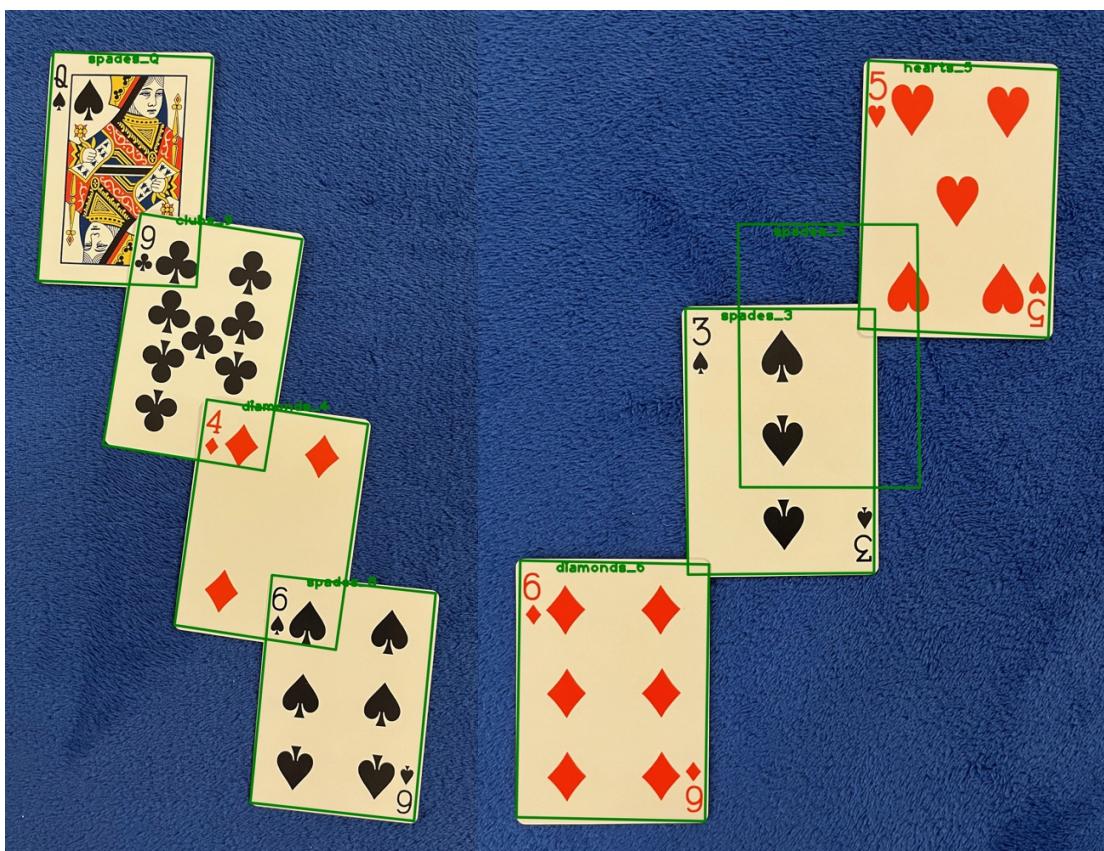
A nagy mértékű átfedésnél már nem ismerte fel a lapokat, csak alacsony arányban, ez a bal oldali képen látszik. A jobb oldali képen azonban felismerte annak ellenére, hogy nem teljesen szemből történt a fotózás.



Kis mértékben átfedő lapok

Az alábbiakban néhány mintát mutatok az egymást csak kis mértékben átfedő lapok felismeréséről. Itt a hibák ellenére a felismerés a legtöbb esetben jól működött, de előfordult, hogy talált nem létező lapokat, vagy rosszul ismert fel lapokat.

Eredeti háttér előtt, amihez a paraméterek igazítva lettek



Más háttér előtt



Más kártyapaklival

