



M Ű E G Y E T E M 1 7 8 2
W Ū E G Y E T E M 1 7 8 2

TDK dolgozat

Emberi fülszövetekről készült makroszkópikus képek vizsgálata
digitális képfeldolgozással

Konzulens: Dr. Barsi Árpád

Külső konzulens: Dr. Grimm András

Készítette: Medve András

TARTALOMJEGYZÉK

1. BEVEZETÉS, CÉL	1
2. A FELADAT	1
3. A VIZSGÁLT TERÜLET ANATÓMIAI ISMERTETÉSE	1
4. KAPOTT KÉPÁLLOMÁNY ISMERTETÉSE.....	1
4.1. A KÉPÁLLOMÁNY KÉSZÍTÉSÉNEK KÖRÜLMÉNYEI	2
4.2. A KÉPEK KINYERHETŐ METAADATAI	2
4.3. A KÉPEK SPECIÁLIS TERÜLETEI.....	2
5. A FELADAT VÉGREHAJTÁSA.....	4
5.1. ELŐKÉSZÍTÉS	4
5.2. A PROJEKT VÉGHEZVITELÉNEK LEHETSÉGES MÓDJAI	4
5.2.1. <i>Ellenőrzött osztályozás</i>	4
5.2.1.1. Pontdiagramok előállítás.....	4
5.2.1.2. További jellemzők levezetése	11
5.2.1.3. A tulajdonságmátrix létrehozása	11
5.2.1.4. Osztályozás I.	13
5.2.1.5. Osztályozás II.	17
5.2.1.6. Főkomponens analízis (PCA).....	26
5.2.1.6.1. 90%-os információtartalom	28
5.2.1.6.2. 95%-os információtartalom	29
5.2.2. <i>Az összes pixel automatikus osztályozása</i>	31
5.2.2.1. Képek osztályozása az RGB és YCbCr komponensek alapján	31
5.2.2.1.1. Képek osztályozása egységesen	32
5.2.2.1.2. Képek osztályozása egymástól függetlenül.....	34
6. JÖVŐBELI TERVEK	36

1. Bevezetés, cél

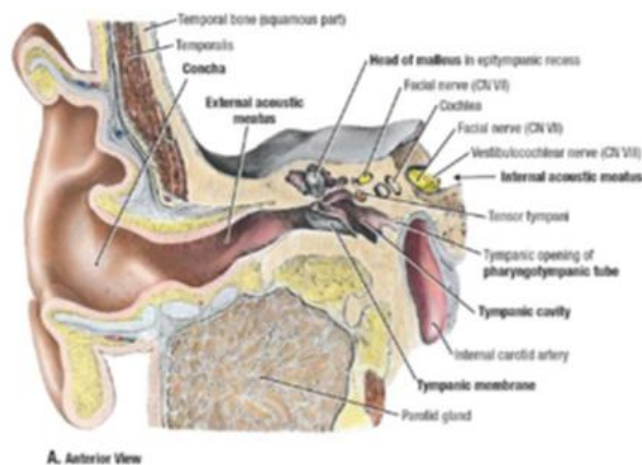
Jelen munka a digitális képfeldolgozás eszközeit, az orvosi képalkotással kívánja ötvözni és ezen témákban kívánja az ismereteket elmélyíteni. A digitális képfeldolgozással kapcsolatos segítséget a Budapesti Műszaki és Gazdaságtudományi Egyetem Fotogrammetria és Térinformatika Tanszéke, míg az orvosi képalkotáshoz kapcsolatos segítséget a Semmelweis Egyetem Fül-Orr-Gégészeti és Fej-Nyaksebészeti Klinikája biztosították.

2. A feladat

A feladat és problémakör a Semmelweis Egyetem Fül-Orr-Gégészeti és Fej-Nyaksebészeti Klinika munkatársainál történt. Megkeresésünkkor a kitűzött céljuk az volt, hogy általuk készített felvételek alapján, valamilyen automatizmus segítségével különítsük el a fül környékén található különböző szövettípusokat, hogy ezt később az oktatásban való szemléltetésre, illetve a szövethatárok pontosítására használják.

3. A vizsgált terület anatómiai ismertetése

Az emberi test fejrészének két oldalán található meg a fül. Ennek látható külső megformálódása a fülkagyló. A feladatban vizsgált szövetek a fülkagylótól a fej belseje felé haladva helyezkednek el.



3.1 ábra: A fül felépítése

Kiindulásként az orvosok 7 különböző szövetet jelöltek meg számunka, amelyeket szerettek volna, ha azonosítanánk. Ezek a szövetek az „elasztikus porc”, a „festett artéria”, a „harántcsíkolt izomszövet A”, a „harántcsíkolt izomszövet B”, a „kötőszövet”, a „nyálmirigy” és a „zsírmirigy” voltak. A szövetek határvonalait az orvosok lehatárolták, ezáltal nekünk a szövetek elhelyezkedésével, kinézetével és tulajdonságaival a továbbiakban nem kellett foglalkoznunk.

4. Kapott képállomány ismertetése

Az elkészített felvételeket kiindulásként kézhez kaptuk. Ezek az állományok digitális fényképek formájában álltak elő, melyek az RGB színcsatornáknak hordozták az intenzitásadatokat. Az

állomány tökéletes vizsgálatához ismerni kell a felvételek pontos körülményeit és a különböző fizikai tulajdonságokat, amelyek a felvétel készítésének pillanatában jelen voltak.

4.1. A képállomány készítésének körülményei

A felvételeket olyan testeken végezték, akik haláluk után felajánlották a testüket tudományos célokra. A testek fejrészének eltávolítása után azt folyadékba helyezték, majd lefagyasztották. A megfagyott testrészt egy hűtőkamrában -30 °C -on tartották és ebben az állapotban készítették róla a felvételeket. A szövetekre és azok elhelyezkedésére leginkább mélységében voltunk kíváncsiak, ezért meg kellett oldani, hogy a fülbe és azon szöveibe belelássunk. Mivel a fej teljesen le volt fagyasztva és ezáltal teljesen szilárd halmazállapotú volt, ezért úgy vélték, ha egy marógéppel bizonyos rétegeket lemarnak és azokról elkészítik a felvételeket, akkor az megfelelő minőségű eredményt fog adni. Ennek a lemart rétegnek a vastagsága $0,1\text{ mm}$ volt.

4.2. A képek kinyerhető metaadatai

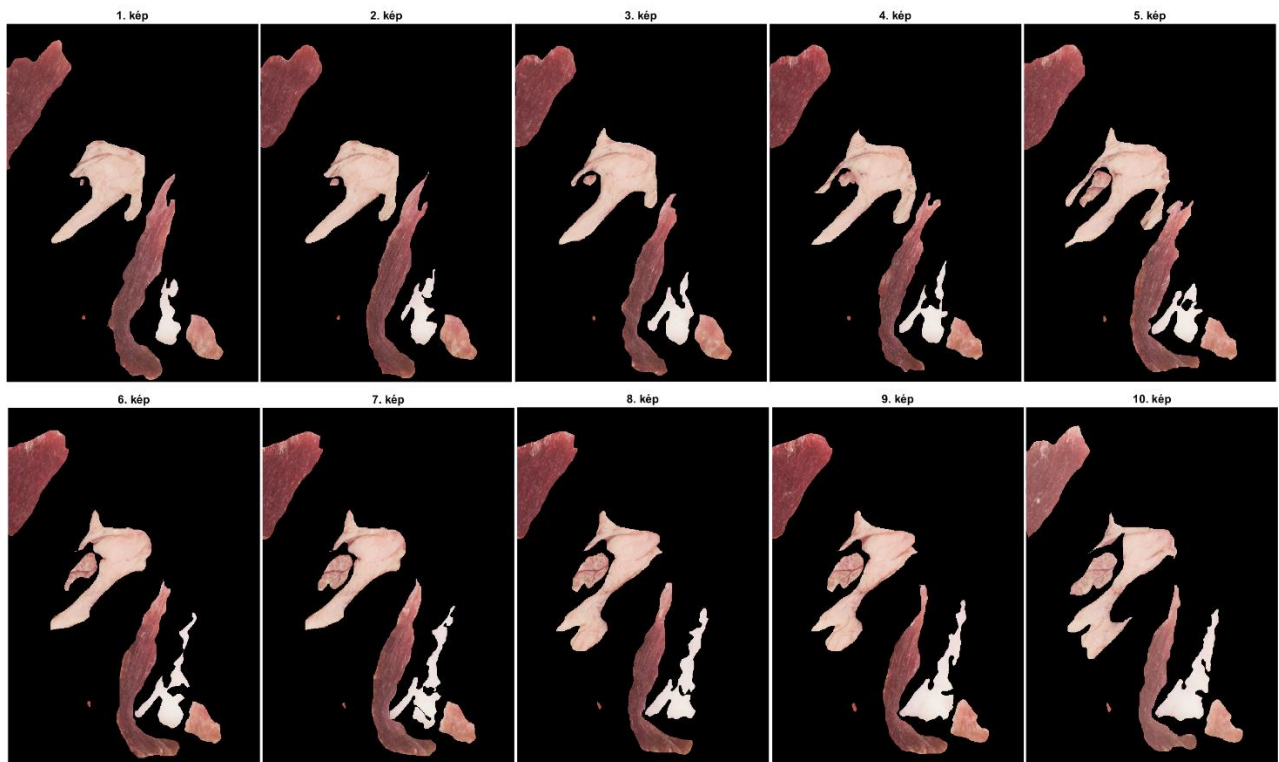
Az elkészült fényképek felbontása 1606×2282 pixel volt, amely $\sim 3,6$ megapixelnek felel meg. A képek színmélysége 24 bit (csatornánként 8 bit = 256 színárnyalat). A fájlok png kiterjesztésben álltak rendelkezésünkre.



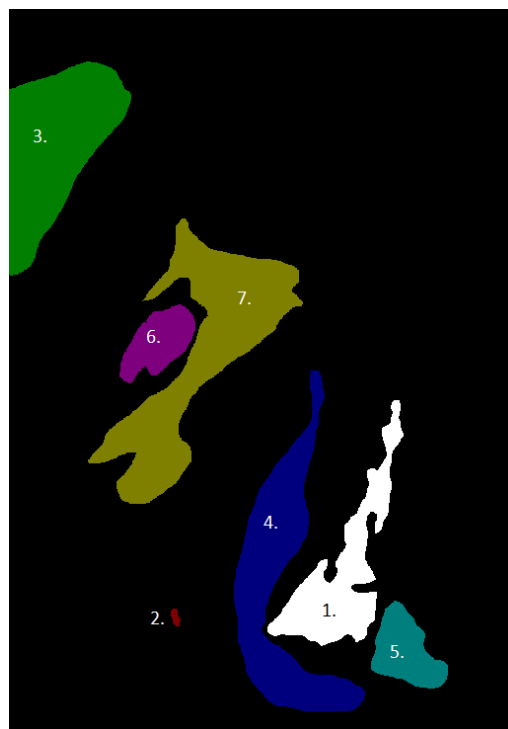
4.1 ábra: Mintakép

4.3. A képek speciális területei

A feladat orvosi jellegéből adódóan mi, mint műszaki beállítottságú emberek, nem tudtunk speciális területeket lehatárolni a képeken. Erre kaptunk orvosi szaktudást és segítséget. A képek mellé mellékelve elküldtük 10 mintaképre a ROI-kat (Region of Interest). Ennek a magyar megfelelője az érdeklődési terület vagy maszk, ami alatt ismert tematika található (esetünkben szövetek). Ezekből az állományokból ki tudtuk nyerni, arra a 10 képre az összes kívánt szövet maszkját, majd ezen maszkolt területekből már elő tudtuk állítani magunknak a tananyagot. Első gondolatainkban azt mondtuk, hogy ezt a 10 képet elemezzük, majd ha előállt valami módszer, akkor ezt a módszert alkalmazzuk az összes többi képen. Az előállt eredményt pedig elküldjük az orvosi csapatnak, akik megállapítják, hogy a módszer milyen megbízhatósággal működik.



4.2 ábra: A maszkok elhelyezkedése az egyes képeken



4.3 ábra: Tematikus ábra a maszkokról (1. Elastikus porc 2. Festett artéria 3. Harántcsíktolt izomszövet A 4. Harántcsíktolt izomszövet B 5. Kötőszövet, 6. Nyálmirigy 7. Zsírmirigy)

5. A feladat végrehajtása

5.1. Előkészítés

A feladatot saját célkitűzésünk szerint Matlab környezetben kívántuk elvégezni. Így a képek elemzése, feldolgozása is ebben a környezetben valósult meg. Ehhez minden rendelkezésünkre álló adatot át kellett alakítani úgy, hogy ezt a Matlabba egyszerűen be lehessen olvasni. Mivel a fényképek egyszerű digitális fényképek voltak, ezért ezeket mindenféle átalakítás nélkül be lehetett hívni a programba. A megkapott maszkok, olyan formátumba voltak, amelyet Matlabbal viszonylag körülményes vagy lehetetlen lett volna beolvasni, ezért áthidaló megoldásként felhasználtuk a Fiji programot, amellyel a maszkokat ki tudtuk exportálni olyan formátumban, amit a Matlab szoftverkörnyezete be tudott olvasni. Miután az adatok numerikusan mátrixokba strukturálva rendelkezésünkre álltak, nekiállhattunk a különböző megoldási lehetőségeken gondolkodni, illetve azokat ki is próbálhattuk.

5.2. A projekt véghezvitelének lehetséges módjai

A fő célunk, hogy a képeink minden képpontját, valamilyen módon osztályokba soroljuk. Ezt az osztályozást két féle módon kezdhethetjük el. Vagy ellenőrzött osztályozást hajtunk végre, vagy automatikusan osztályozzuk a képpontokat. Mindkét módszerhez ugyan azok az adatok szükségesek, de a két módszerben a bemenő adatok másképp kerülnek felhasználásra. Az ellenőrzött osztályozás esetén vannak tanuló tereink (maszkok által lefedett terület) és ezeket tanulmányozva állapítjuk meg az osztályozásunk alapját, majd ezek segítségével határozzuk meg a képek képpontjainak hovatartozását. Ellenben az automatikus osztályozással (klaszterezés) az egész képet alapul vesszük, valamilyen elmélet szerint automatikusan kap minden egyes pixel egy azonosítót, majd utána megnézzük, hogy a maszkok alatti pixelek mennyire homogének illetve, hogy egy klaszter hány maszk alatt található meg. A két megoldás teljesen eltérő területekkel és teljesen más adatmennyiséggel dolgozik, ezért érdemes mindkét módszerrel megvizsgálni a képeket.

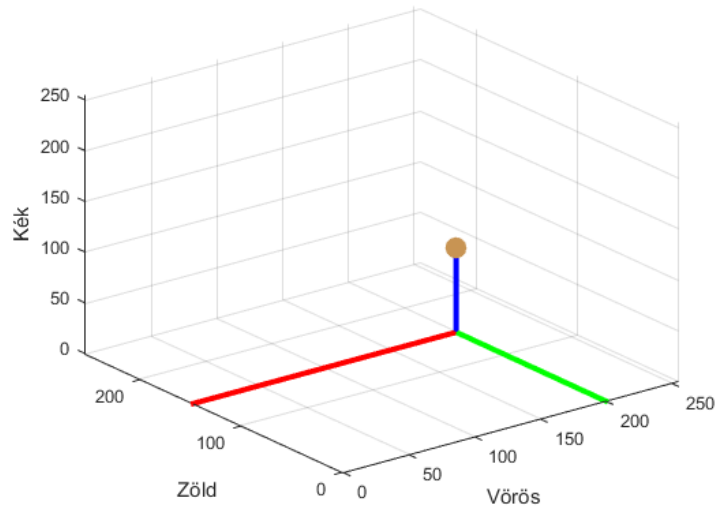
5.2.1. Ellenőrzött osztályozás

Ezen rész alatt gyűjtjük egybe azokat az eljárásokat, amelyek abból indulnak ki, hogy ismerjük a meghatározott szövegeinket és azok helyzetét a képeken, majd ezekből a pixelekből vezetünk le további jellemzőket és osztályozzuk a fennmaradó képeket.

5.2.1.1. Pontdiagramok előállítás

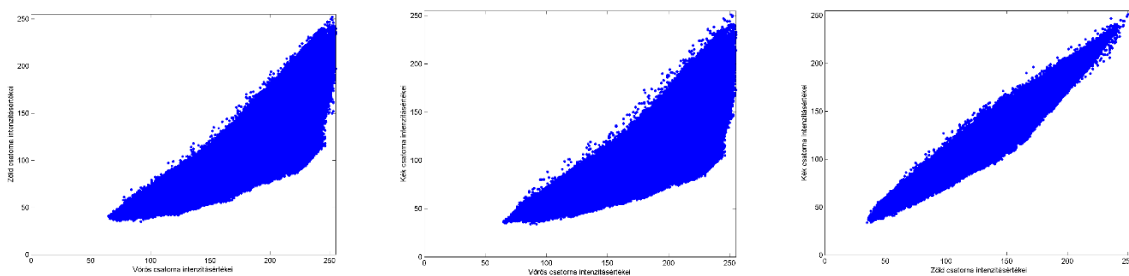
A legalapvetőbb és legegyszerűbb lehetőségnek az tűnt, ha egyszerűen felhasználjuk a három csatorna (R-G-B) intenzitásadatait és ezekből pontdiagramokat készítünk és ezekre ránézve el tudjuk-e különíteni az adott maszkokat.

Pontdiagram: a pontdiagram egy olyan ponthalmaz, amely a tengelyein egy adott pixel adott tulajdonságait mérjük fel. Pl.: egy matematikai koordináta rendszert alapul véve az X tengelyen mérjük föl az adott pixel vörös (R) intenzitásértékét, az Y tengelyen mérjük fel ugyan azon pixel zöld (G) intenzitásértékét és a Z tengelyre mérjük föl a pixel kék (B) intenzitásértékét.



5.1 ábra: Pontdiagram egy pixelhez

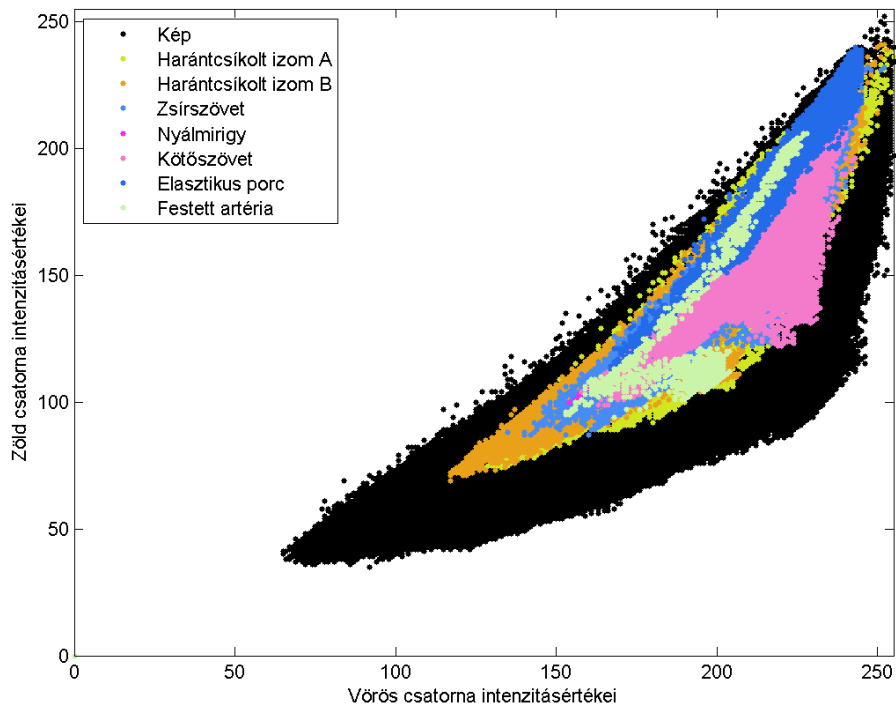
Mi a nagy adatmennyiség miatt ($1606 * 2282 * 10$ pixel = 36648920 pixel) inkább szétbontottuk a feladatot és a 3D-s megjelenítés helyett inkább ennek a 2D-s nézeteit alkottuk meg. Első körben feltételeztük, hogy a képek egyenlő minőségűre sikerültek, így egybevetjük mind a 10 képet, amelyeken fel voltak tüntetve a szövetek helyei, majd ebből elkészítettük az összes kép egységes pontdiagramját.



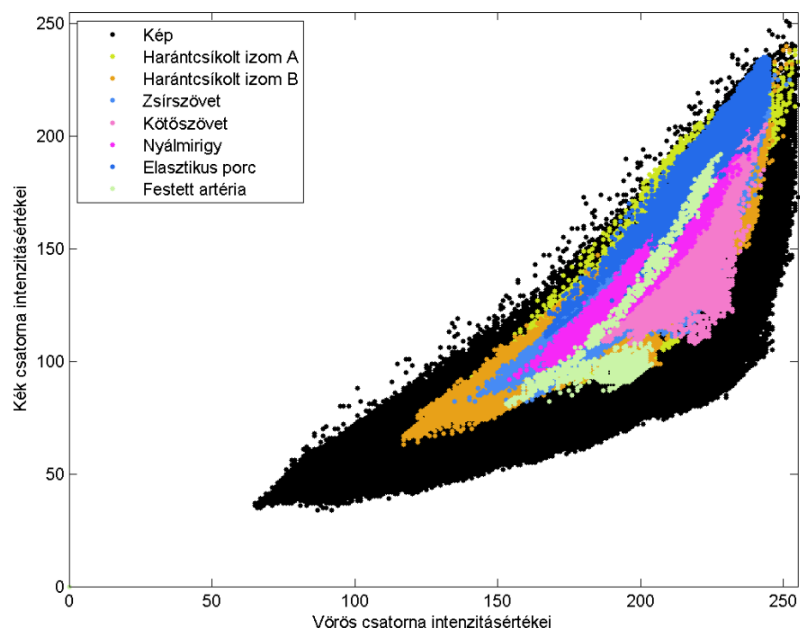
5.2 ábra: A 2D-s pontdiagramok (balról jobbra: Vörös-Zöld, Vörös-Kék, Zöld-Kék)

Látható hogy a pontdiagram egy tömör egységet alkot, nincsenek elkülönülő részek, amelyek esetleg arra utalhatnának, hogy a keresett szövetünk épp ott helyezkedne el.

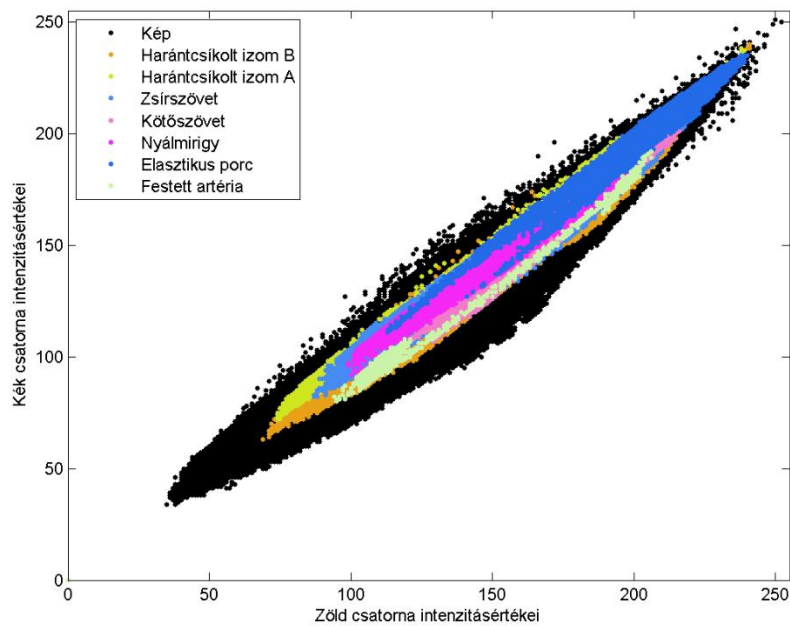
A következőkben ezeken a pontdiagramokon kiemeljük azokat az intenzitáspárokat, amelyek egy-egy szövethez tartoznak, majd ezeket egyszerre ábrázoljuk. Amennyiben azt az eredményt kapjuk, hogy a pontdiagramon megfelelően elkülönülnek a különböző szövetekhez tartozó pontok, akkor az adott szövet alatti intenzitáspárokat rákeresve nagy valószínűséggel visszkapjuk azokat a helyeket, ahol az a szövet található. Ha ezek a rétegek fedik egymást, akkor további elemzésekre van szükség.



5.3 ábra: Vörös-zöld pontdiagram az összes maszkkal

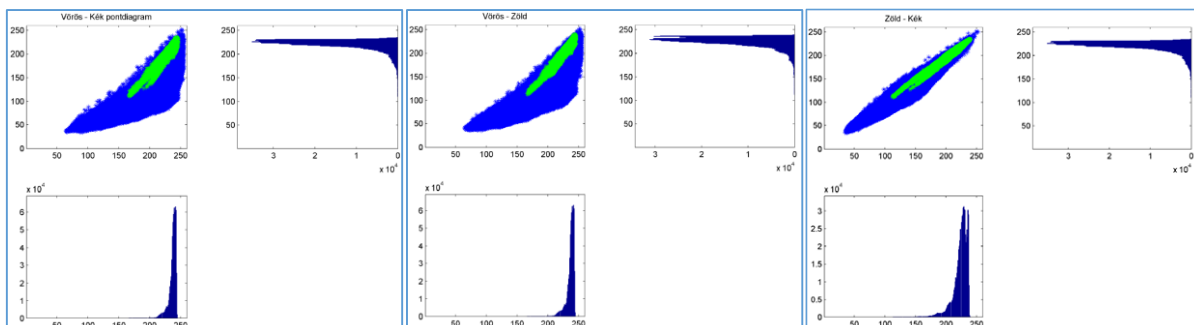


5.4 ábra: Vörös-kék pontdiagram az összes maszkkal

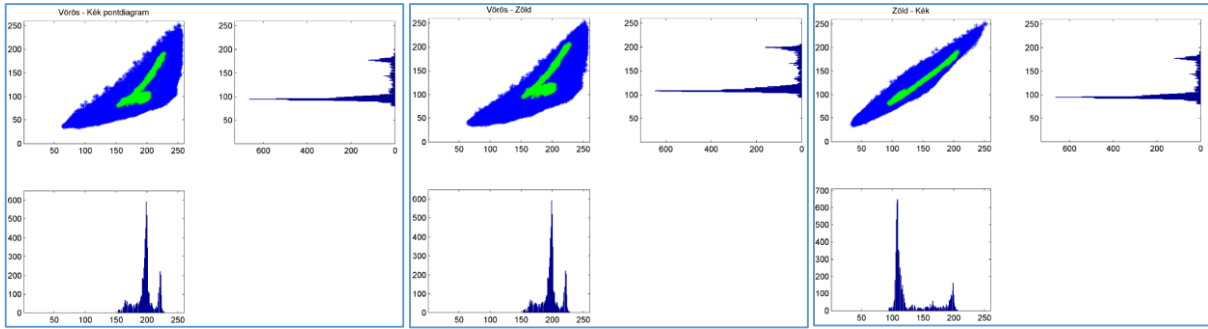


5.5 ábra: Zöld-kék pontdiagram az összes maszkkal

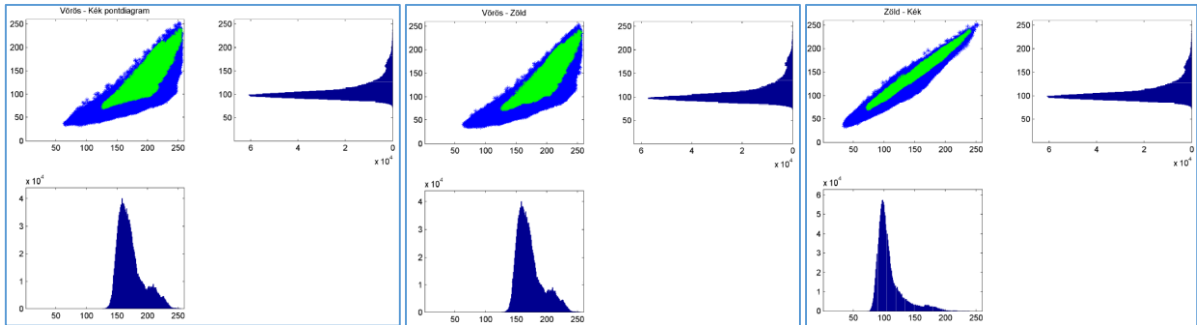
Ezek a képek észrevehető, hogy majdnem mindegyik szövet fed egymást. Szinte majdhogynem egymáson helyezkednek el. Ez az egyértelmű azonosítást nem teszi lehetővé. Ezután átgondoltuk, hogy ezen a módszeren lehet-e még finomítani, valamilyen módon felhasználni az eddigieket, vagy esetleg mi lehet a probléma? Könnyen belátható, hogy a fenti ábrák nem tökéletesen tükrözik a valóságot. Korábban említettem, hogy mekkora adatmennyiséggel dolgozunk, ez több millió pont, az ábrákon elhelyeztük az összes képpont párt úgy, hogy minden egyes kombinációt tartalmaz. Viszont a kép minden kombinációból mindössze csak egyet reprezentál. Így érdemesnek láttam megnézni, hogy a pontdiagramon az egyes maszkok milyen eloszlásban vannak jelen a pontdiagramon. Ha ez az intenzitásterben jól lehatárolható környezetben koncentrálódik, akkor bizonyos veszteséggel, de a pixelek nagy része azonosítható lenne.



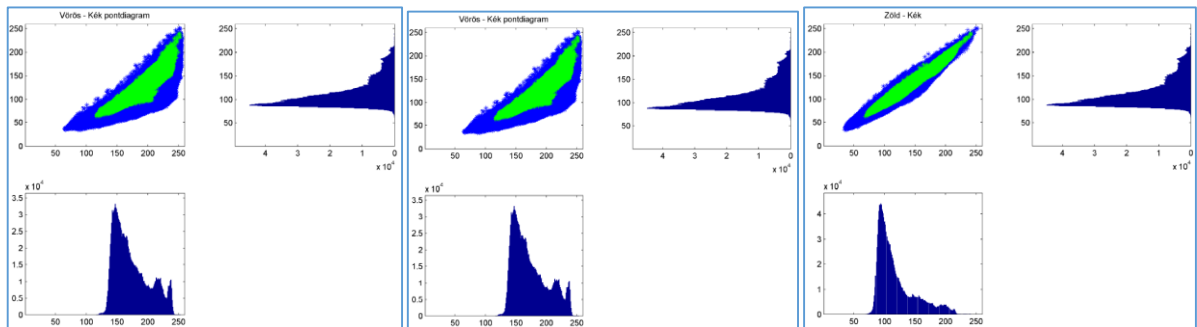
5.6 ábra: Elasztikus porc pontdiagramja, hisztogrammal minden sávkombinációban



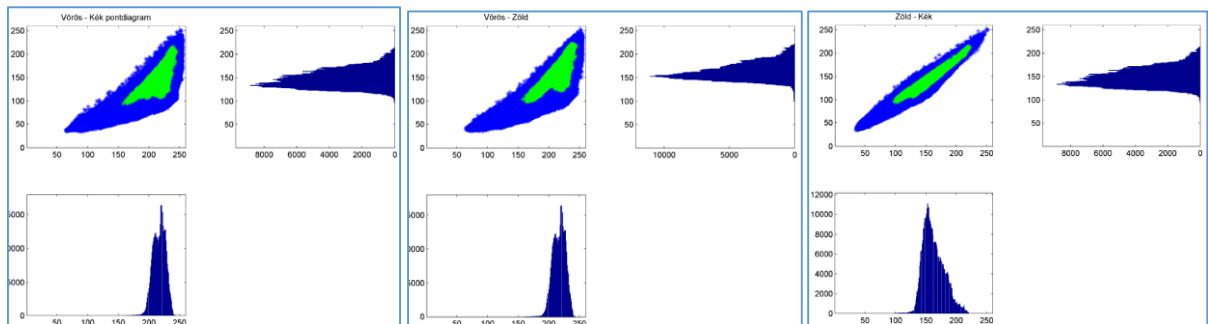
5.7 ábra: Festett artéria pontdiagramja, hisztogrammal



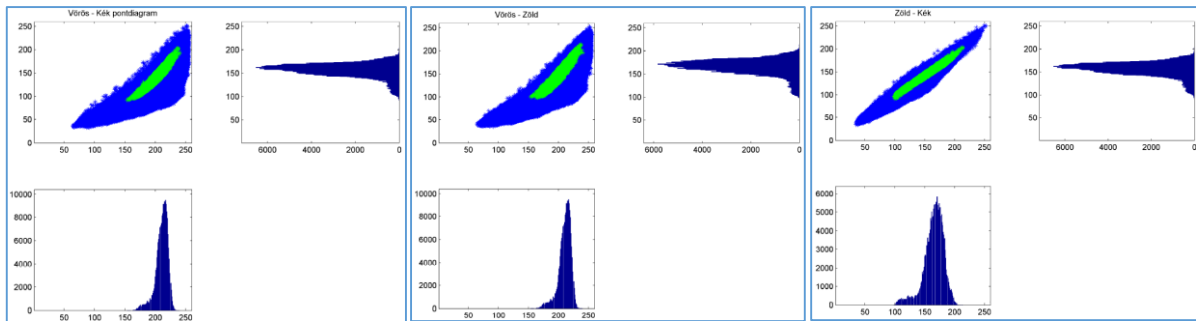
5.8 ábra: Harántcsíkolt izom A pontdiagramja, hisztogrammal



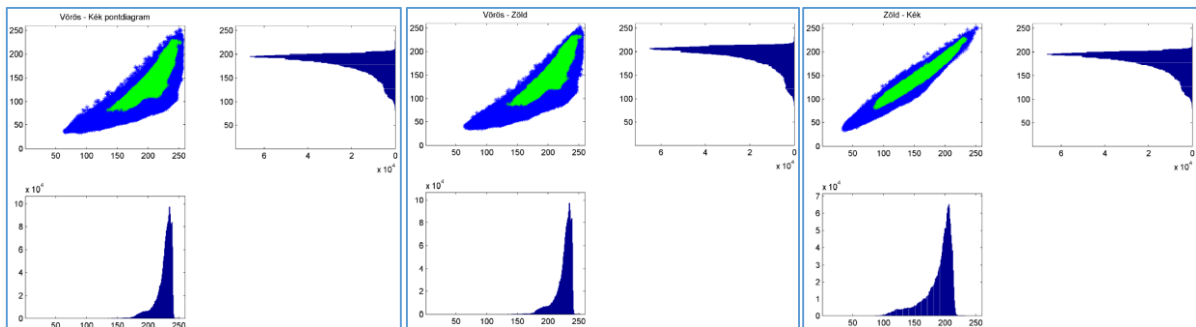
5.9 ábra: Harántcsíkolt izom B pontdiagramja, hisztogrammal



5.10 ábra: Kötőszövet pontdiagramja, hisztogrammal



5.11 ábra: Nyálmirigy pontdiagramja, hisztogrammal

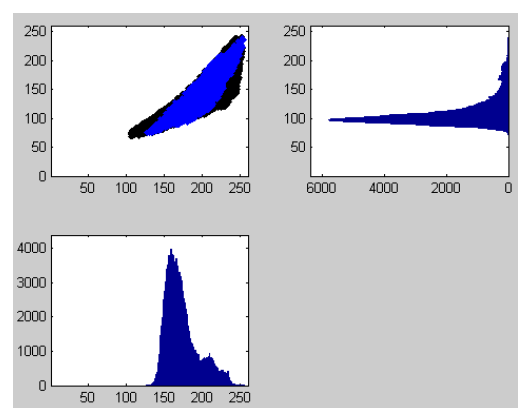
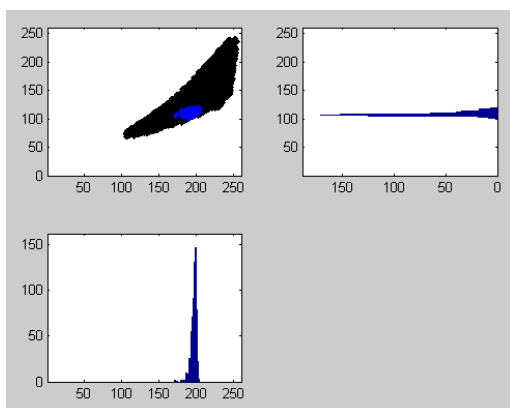
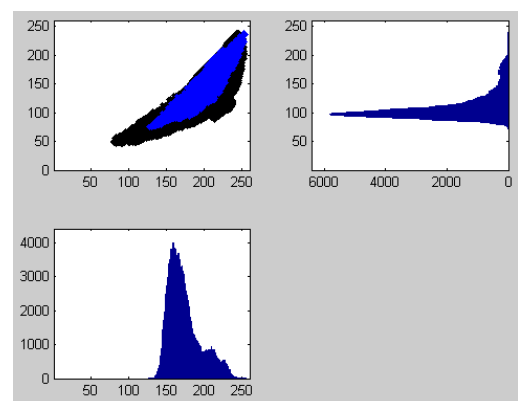
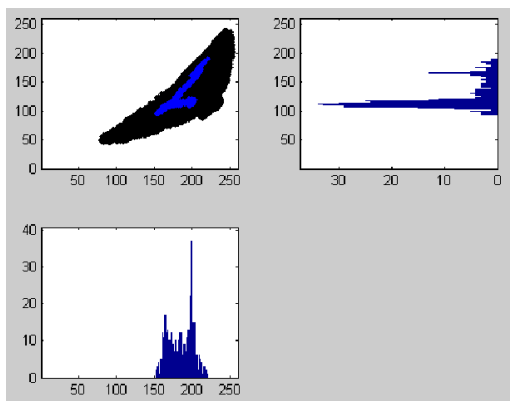
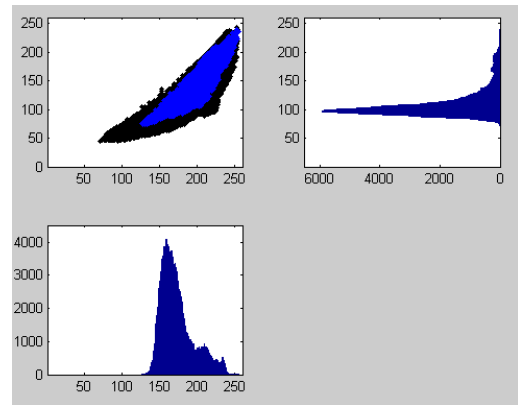
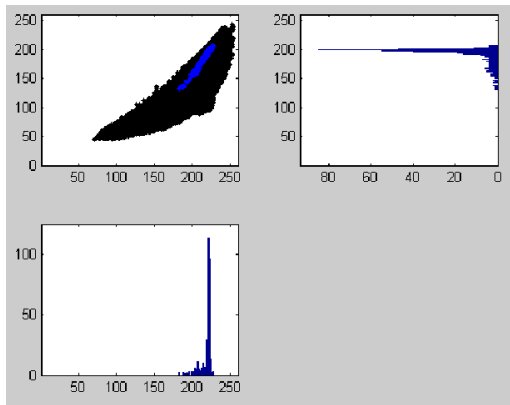


5.12 ábra: Zsírmirigy pontdiagramja, hisztogrammal

Látható, hogy van egy-két olyan szövet, melynek hisztogramja igen kecsegtetőnek tűnik, egy csúccsal rendelkeznek és a szórása is megfelelően kicsi. De sajnos a legtöbb szövetnél annyira kiszélesedik ez a hisztogram, hogy ha a várható érték környezetében meghúznánk egy határt, akkor ez azt eredményezné, hogy nagyon sok, ahhoz a szövethez tartozó pixelt nem azonosítanánk. Ez egyenlő lenne azzal, hogy a szövetek csak töredékét azonosítjuk, ami nekünk nem elegendő. Emellett a hisztogramcsúcsok átfedésben vannak, vagy legalábbis annyira közel állnak egymáshoz, hogy nem lehet megfelelő megbízhatósági szint mellett szeparálni a két kategóriát.

Mindezek után nagyon úgy fest, hogy ez a módszer számunkra nem fog semmiféleképpen megfelelő eredményt biztosítani, de még ide tartozóan felmerült egy olyan eshetőség, miszerint az egyik kiindulási feltételünk nem valósul meg. Ez pedig a fényképek készítésénél a fényviszonyok változása. Könnyen elképzelhető, hogy készítenek egy fényképet, majd lecsiszolják a felületet és a lecsiszolt felület másképp veri vissza a fényt, ezáltal az ugyanazon szövetek más és más intenzitásértékekkel kerülnek rögzítésre a kamerában. Utólagosan még az a problémakör is ehhez az elemzéshez sorolható, ha egy adott szövet színe a test belseje felé haladva változik.

Ennek a módszernek a megoldása a következő volt. Azt mondtuk, hogy nem az összes képet tekintjük egyszerre, hanem vesszük a képeket egyesével és az ahhoz a képhez tartozó szöveteket rakjuk föl, majd ezeket a képeket sorba rendezzük.



5.13 ábra: Festett artéria pontdiagramja, hisztogrammal az 1. 5. és 10. képen

5.14 ábra: A harántcsíkolt izomszövet A pontdiagramja, hisztogrammal az 1. 5. és 10. képen

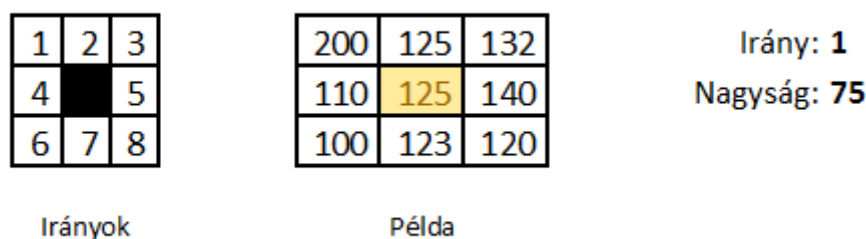
Ebből meglepődve lehet tapasztalni, hogy például korábban láttuk, hogy a festett artéria nevezetű szövetnek a hisztogramja igen biztató volt azonosítás szempontjából, viszont ezt a folyamatot megvizsgálva azt tapasztaljuk, hogy ennek a szövetnek a pontdiagramja képről képre változtatja a helyzetét. Ellenpéldaként ott van a harántcsíkolt izomszövet A, amely szinte semmire nem változtatja a helyzetét egyik képen sem, viszont erről az előző lépésben kellett lemondanunk, mert túlságosan széles volt a hisztogramja. Természetesen ezzel a módszerrel végigmentünk az összes szövet összes sávján, de mivel az előálló adatmennyiség és az információ tartalom között nagyon kicsi volt a korreláció, ezért hoztuk fel csak ezt a két fenti példát.

Összességében elmondhatjuk, hogy ez a módszer nem vezet minket a kívánt eredményre, hiába a módszer részletes körüljárása. A módszer viszonylagos egyszerűsége, könnyen érthetősége és a módszer hatékonysága miatt véltük érdemesnek az ilyen részletességű vizsgálatot.

5.2.1.2. További jellemzők levezetése

Mivel a pixelek RGB adataival nem tudtunk előrébb haladni a feladat megoldásában, ezért további jellemzőket kívántunk bevezetni, amelyek jellemzőek a szövetekre. Úgy véltük, hogy a színcsatornák gradiensei olyan jellemzői lesznek a szöveteknek, amelyek már elegendő információt hordoznak magukban, hogy a szöveteket külön tudjuk választani. Itt kiemelném, hogy még mindig csak a maszkok alatti területeket vizsgáljuk.

Gradiens: egy adott pontban, mekkora a legnagyobb lejtés és ez melyik irányba mutat.



5.15 ábra: Példa a gradiens bemutatására

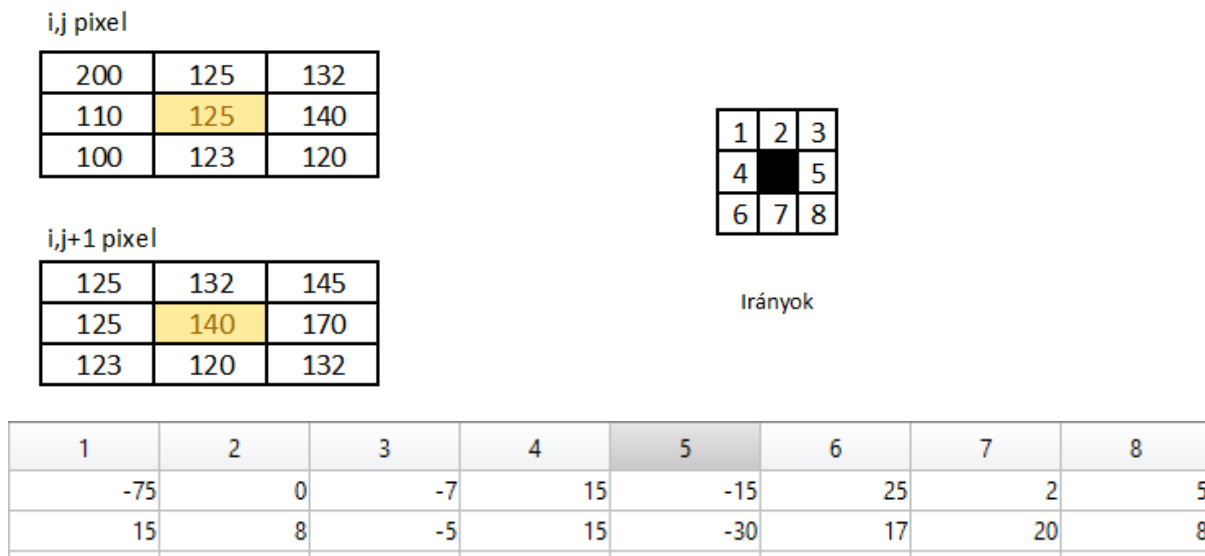
Első gondolattal azt mondtuk, hogy ezzel ki tudnánk használni az szövetek azon tulajdonságait, hogy ezek a szövetek sok esetben irányítottak és valamilyen szerkezettel rendelkeznek, vagyis amelyik irányba halad a szövet, arrafelé a gradiens értéke közel 0 lesz, míg a rá merőleges irányban a maximális.

Alapvetően a gradiens a legnagyobb lejtés mértékét és irányát mutatja. Mivel a későbbiekben használt osztályozási módszereknél majdnem irreleváns, hogy hány tulajdonságot használunk fel, ezért nem csak a gradiens értéket vettük, hanem azt mondtuk, hogy az adott pixelünkönél minden irányba mutató lejtést figyelembe veszünk. Így egy tulajdonságmátrixot kaptunk, amelynek soraiban a pixelek oszlopaiban a tulajdonságok találhatóak. A mátrix méretét az osztályozásba bevont pixelek és a felhasznált tulajdonságok száma határozza meg.

5.2.1.3. A tulajdonságmátrix létrehozása

A tulajdonságmátrixot úgy képzeltük el, hogy a sorok fogják képezni a pixeleket, az oszlopok pedig az adott pixelhez tartozó bármely tulajdonságot. Az oszlopok száma tulajdonképpen tetszőleges, ez egyedül a függvény futási idejét lassítja, de alapvetően az n dimenziós teret is ugyanúgy tudjuk osztályozni, mint az egyszerűbb 2-3 dimenziós teret.

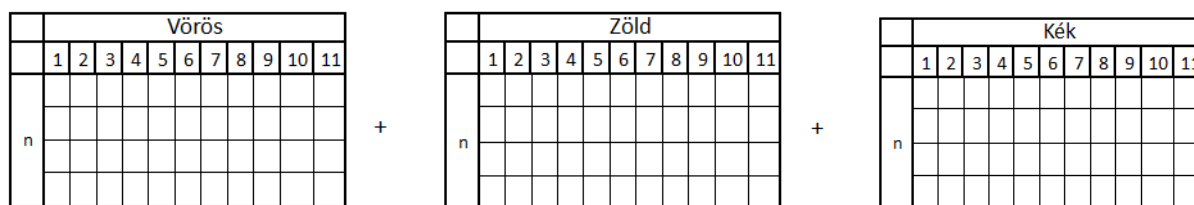
Az első 8 oszlop feltöltése a már korábban említett lejtésnagyságokkal került feltöltésre. Ezt az alábbi módon kell elképzelni:



5.16 ábra: Tulajdonságmátrix előállítás

A tulajdonságmátrix oszlopainak kialakítása a következő módon történt. Vegyük az első kép első sávjának bármelyik pixelét, amely nem a szélső sorok valamelyikében helyezkedik el. Ennek az egy pixelnek lesz 8 darab szomszédja, ez a 8 szomszéd fogja képezni a tulajdonságmátrix vörös csatornájának első 8 oszlopát. Majd, hogy ne csak ezekre a pusztá értékekre támaszkodjunk, levezettünk még további, nagyon egyszerű mérőszámokat. Így a 9. oszlopban elhelyezkedő attribútum tárolta azt a számot, hogy összesen hány nem 0 értékű irány van, jobban megfogalmazva, hány irányban van lejtés a kiszemelt képpontunktól. A 10. oszlop tárolta azt a tulajdonságot, hogy mekkora a maximális intenzitáskülönbség, ez az adott pixel gradiensének nagysága. Végül 11. tulajdonságként felvettük, hogy a 10. oszlopban található maximális intenzitáskülönbség hányszor fordul elő a pixel környezetében.

Ez így 11 tulajdonság. De ez csak egyetlen sáv egyik pixele. Ennek a pixelnek megvannak ezen tulajdonságai a zöld, illetve kék csatornában is, ami azt jelenti hogy még 2-szer elő lehet állítani ezt a 11 oszlopból álló mátrixot. Mivel ennek a 3 mátrixnak a sorai ugyanazon pixel tulajdonságait rejtik, ezért megtehetjük azt, hogy ezt a 3 mátrixot egymás mellé helyezzük, ezáltal előállítva egy olyan tulajdonságmátrixot, amelynek soraiban a pixelek vannak az oszlopokban pedig a 3 sávból levezetett 33 tulajdonság.



5.17 ábra: A 33 elemből álló mátrix összetétele

Ezzel meg is volt a koncepció, hogy mik alapján akarjuk azonosítani a különböző szöveteket. Jelen fejezetben azt tűztük ki célunknak, hogy ellenőrzött osztályozást hajtunk végre úgy, hogy csak a maszkolt pixelekkal foglalkozunk. Ebből adódik, hogy nekünk van 7 db maszkunk 10 darab képen. Először is összevonjuk az összetartozó szövetek képpontjait (jelen pillanatban

nem foglalkozunk vele, hogy melyik képről származik az a képpont) mindhárom csatornában, ezzel előállítva egy olyan mátrixot, amely egyetlen szövetnek tartalmazza mind a 33 tulajdonságát. Ezt elvégezzük az összes szövetre, így lesz 7 db teljesen különálló mátrixunk. Mivel az előállítás során ügyeltünk rá, hogy az egyes oszlopokban mindig ugyan az a típusú tulajdonság szerepeljen, ezért megtehetjük azt, hogy a 7 db mátrixot egymás alá helyezzük, így megkapva egy nagy, de mindent magában foglaló, tulajdonságmátrixot.

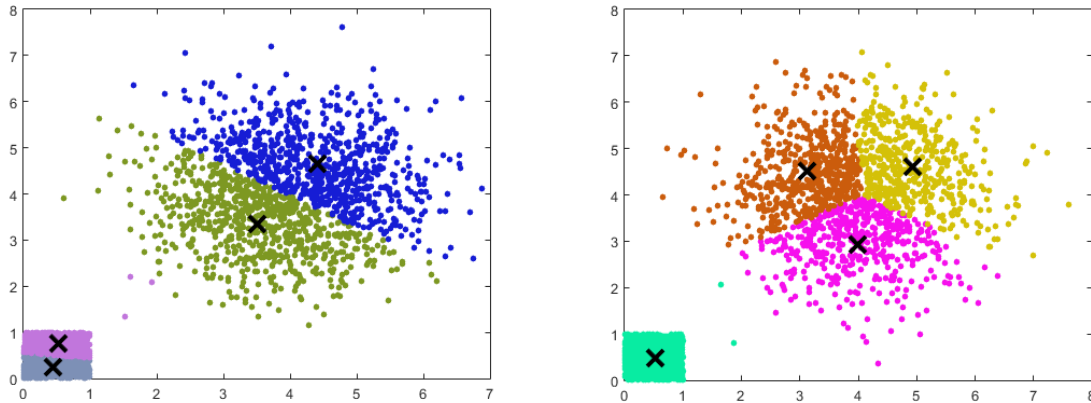
Szövet(pixelszám)	1-33. tulajdonság		
Elasztikus porc (i)	$i \times 11$	$i \times 11$	$i \times 11$
Festett artéria (j)	$j \times 11$	$j \times 11$	$j \times 11$
.			
.			
.			
Zsírmirigy (n)	$n \times 11$	$n \times 11$	$n \times 11$

5.18 ábra: Az osztályozni kívánt mátrix végleges formája

Ezen munkarész kapcsán feltételeztük, hogy ennyi tulajdonság elég lesz ahhoz, hogy a megfelelő osztályozást végrehajtsuk. A teljes forrásadatunkat strukturálva előállítottuk, így elkezdhattük az osztályozás menetének végiggondolását.

5.2.1.4. Osztályozás I.

Az osztályozás a *kmeans* paranccsal történt a Matlab programon belül. A *kmeans* osztályozás egy heurisztikus súlypont kereső algoritmus, melynek működési elve a következő. Van egy adathalmazom, aminek az elemei a sorokban található, míg a tulajdonságait az oszlopok írják le. Az oszlopokban található adatok felfoghatók akár koordinátáknak is, amely egy derékszögű koordináta-rendszerben akár ábrázolhatók is (3 dimenzióig). Minden elemet ábrázolva megkaphatjuk az elemek tulajdonságterét. Maga az algoritmus ebben a térben lerak annyi súlypontot véletlenszerűen, ahány osztályt szeretnénk mi a végén látni, majd az összes elemet hozzárendeli a legközelebbi súlyponthoz (térbeli távolságszámítás). Ezzel elért az első lépés végéhez, miután megvannak az osztályozott elemek a meglévő súlypontot áthelyezi az adott csoport tényleges súlypontjába és újra elvégzi az összes elem hozzárendelését. Ez mindaddig folytatódik, amíg a súlypont nem változik, illetve az ismétlés szám el nem ér egy bizonyos felső iterációs korlátot. A szemléletesség kedvéért egy véletlenszerűen generált adathalmaz osztályozása látható az 5.19 ábrán.



5.19 ábra: Kmeans függvény működése gyakorlati példán

Az ábrán szereplő elemek két tulajdonságát az x és y tengelyeken ábrázoltam, látható, hogy a centroidokat hova helyezte el az algoritmus, illetve hogy ahhoz a súlyponthoz mely elemek tartoznak.

A mi valódi esetünkben is ugyan ezt kell végrehajtani, csak nekünk 33 tulajdonságunk van, így ennek az ábrázolása nem kivitelezhető, viszont a matematikai térbeli távolságszámítás képlete 3-nál több dimenzióra is tökéletesen alkalmazható.

$$d_{xy} = \left(\sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2}$$

5.20 ábra: Térbeli távolság számításának általános képlete n dimenziós esetben

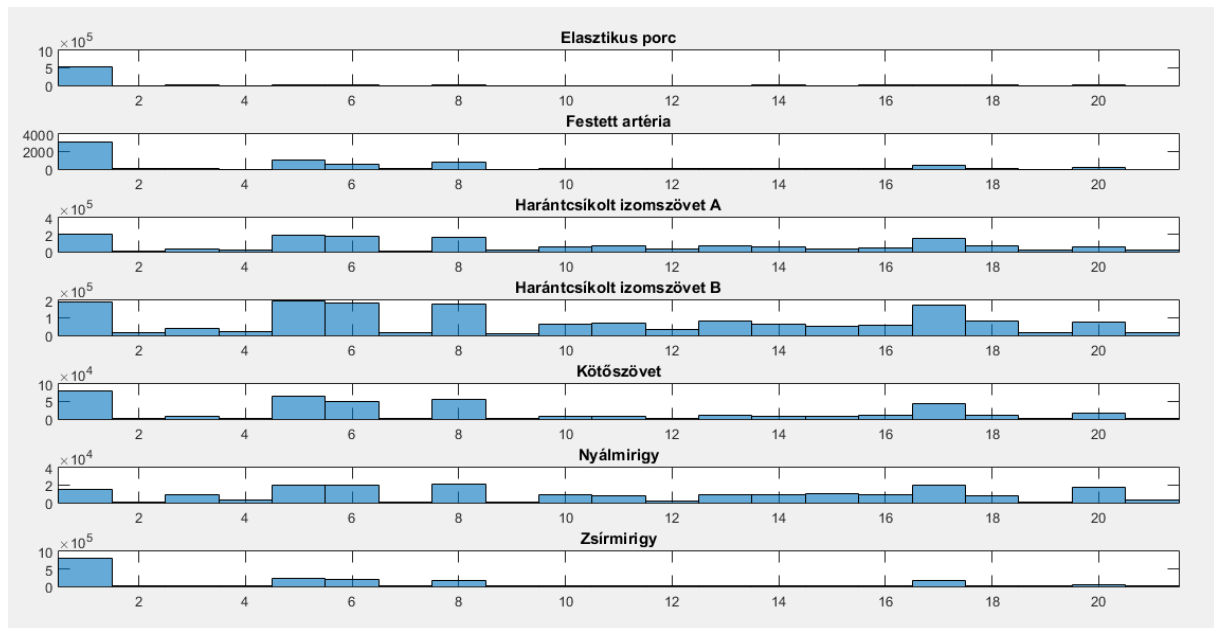
Most már csak meg kell határozni a klaszterszámot, hogy eredményként hány osztályba sorolja be a képpontjainkat. El lehet gondolkodni rajta, hogy mivel 7 szövetünk adatát rakjuk bele, ezért 7 klaszter elég lesz-e. Erre a gondolatmenetre kicsit rációzol a fenti példa, mely azt mutatja, hogy a bal alsó sarokban hiába van egy jól elkülöníthető rész, mégis két klaszterbe sorolja azokat az elemeket. Így a tapasztalat és a szakirodalom alapján inkább azt mondjuk, hogy 2-3-szoros többlet klasztert hozunk létre és azt, hogy akkor egyetlen szövetünk akár több osztályból is összeállhat. A megválasztott klaszterszám így 21 lett.

Miután sikeresen előállítottuk az adatainkat, lefuttathattuk a *kmeans* függvényt, amely eredménye egy vektor volt, melynek hossza megegyezett az elemszámmal, vagyis minden egyes pixelünknek adott egy azonosítót, hogy őt melyik csoportba is sorolta be.

Itt megint hivatkoznék a nagy elemszámra, amely a táblázat szemlélésével történő elemzést teljesen ellehetetleníti. Így más, olyan módszereket kell kitalálni, amelyekkel ránézésre el tudjuk dönteni, hogy ez a módszer használható-e a mostani formájában, esetleg van olyan rész, amire nagyon jól használható, vagy kiderül, hogy rossz irányba indultunk el az elején és az egész adathalmazzal nem érdemes foglalkozni.

Ehhez a legcélszerűbbnek azt láttuk, hogy valamilyen gyakoriság alapján csoportosítsuk az adatokat. Most minden képpontunkhoz egy klasztert és egy szövettípust rendeltünk, ezáltal kétfajta csoportosítást is végre fogunk tudni hajtani. Elsőnek azt mondjuk, hogy van 7

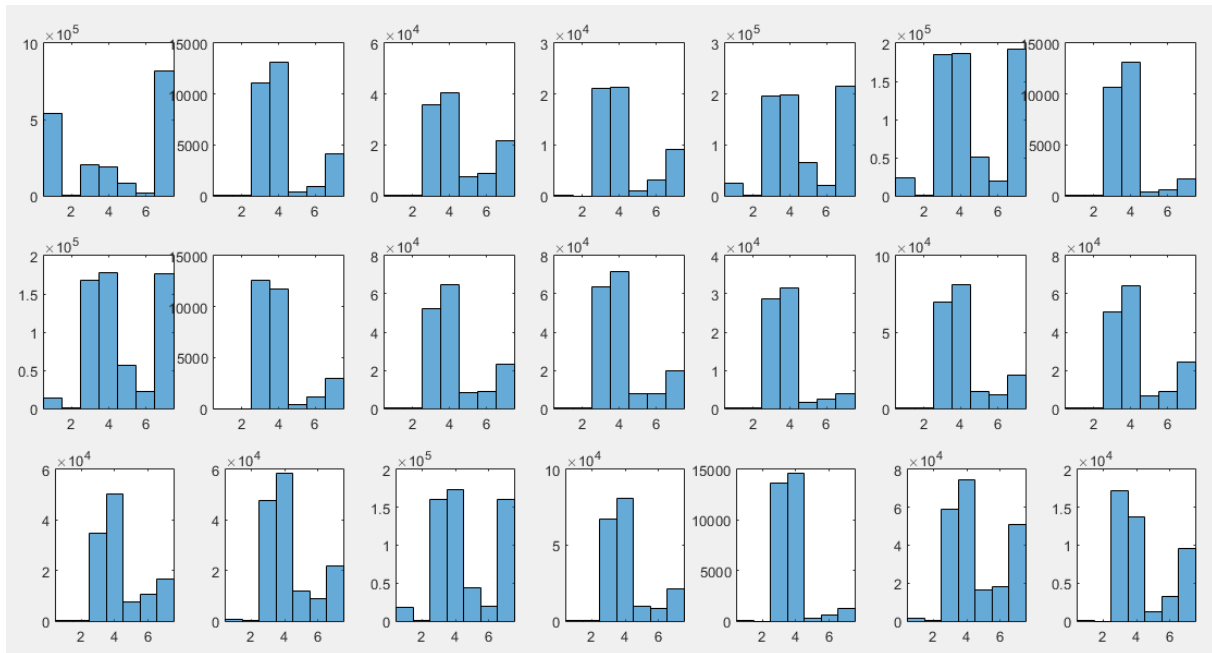
szövetünk, nézzük meg minden szövet képpontjait, hogy hány klaszterbe lettek besorolva. Itt az lenne a kívánatos, ha egy szövethez 2-3 klaszter csatlakozna és eközben ezek a klaszterek nem jelennek meg másik szöveteknél.



5.21 ábra: Az egyes szövetekhez tartozó klaszterek előfordulása

Az ábrából látszik, hogy nem pont úgy válnak szét a klaszterek, ahogy mi azt szeretnénk. Az 1-es klaszterből például mindegyik szövet kap egy elég nagy hányadot, míg van olyan klaszter, amelyből alig található bármelyik szövetnél. Ilyen például a 21-es klaszterünk. Emellett az is látszik, hogy a harántcsíkolt izomszövetek nagyon sokfajta klaszterből tartalmaznak pixeleket, ami szintén ellehetetleníti az elkülönítést. Tehát mondhatjuk azt, hogy az eddigi eredmények alapján nem tűnik a leghatékonyabb megoldásnak az intenzitáskülönbségek felhasználása.

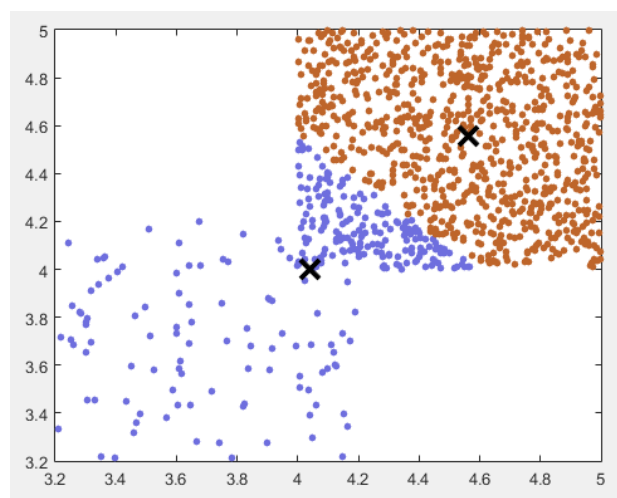
Hiába jött ki az az eredmény, hogy ez alapján nem lehet osztályozni megfelelően a képpontokat, még nem érdemes eldobni a lehetőséget. Van még a másíkfajta osztályozási lehetőség, ami magában rejtheti a végső megoldás egy rész megoldását. A csoportosítás lényege az előző csoportosításnak a fordítottjaként is felfogható. Most azt mondjuk, hogy van a 21 klaszterünk és arra vagyunk kíváncsiak, az egyes klaszterek milyen szövetekből tartalmaznak mintákat. Tehát vesszük például az első klasztert és megnézzük, mennyi pixel tartozik ezek közül a különböző szövetekhez.



5.22 ábra: Az egyes klaszterekhez tartozó szövetek előfordulása

Az előző eredményből már várható volt a mostani eredmény, de mindenféleképpen érdemes megcsinálni ezt a fajta vizsgálatot is, mert előfordulhat, hogy van olyan klaszter, amely csak egyetlen szövethez csatlakozik, ezáltal egy részeredményként a későbbiekben fel tudnánk használni. A mostani esetben sajnos ez az eshetőség sem áll fenn.

További lehetőségként fennállhat a következő két lehetőség. Vagy dolgozunk tovább ugyan ezekkel az adatokkal, csak választunk egy teljesen más mechanizmuson alapuló eljárást, vagy eldobjuk az eddigi adatainkat, származtatunk további tulajdonságot, majd az új adathalmazon elvégezzük újra az osztályozást. Mi most azt mondtuk, hogy próbáljuk meg más módszerrel. Ez a döntés azért született meg, mert a *kmeans* egyrészt véletlenszerűen rakja le a kezdőpontjait (ez természetesen konvergál, csak lehet olyan eset, amikor ez a bizonytalanság számít az adatoknál), másrészt, ha a pontok között van egy kicsi átfedés, akkor a súlypontok beállhatnak kedvezőtlen helyzetben.



5.23 ábra: A *kmeans* eljárás működése kis átfedés és eltérő elemszám esetén

A fenti példa már a következő problémát is szemlélteti, mely azt mondja, hogy ha az egyes csoportokban eltérő az elemszám, akkor az eltérő elemszámok miatt el tud húzódni a súlypont a több elemű csoport irányába. Ami az osztályozásnál azt a következményt vonja maga után, hogy az egyik klaszter mindkét ponthalmazból fog elemeket tartalmazni. Kvázi egy ilyen eredményt kaptunk korábban és nem lehet kizárni, hogy nem éppen ez volt a probléma oka.

5.2.1.5. Osztályozás II.

A következő osztályozási módszer a neurális hálózatokon alapszik. A hálózat csomópontokból és a csomópontokat összekötő élekből tevődik össze. A neurális hálózatoknál a csomópontokat neuronoknak nevezik. A módszer lényege, hogy van egy hálózatunk és ennek a hálózatnak a neuronjai versengenek, hogy ők kapják meg a soron következő elemet. Amelyik neuron győz, a saját értékét módosítja úgy, hogy a megnyert elemhez közelebb kerüljön az értéke. Ezt az értéket súlynak nevezi a szakirodalom. Tehát amelyik neuron megnyeri az adott elemet, annak a súlya megváltozik. Majd, ha az algoritmus végigfutott az összes elemen, akkor kezdi előlről a folyamatot, a már módosított hálózaton. Mindezt előre megadott iterációs számig.

$$\mathbf{w}_i(q) = \mathbf{w}_i(q - 1) + \alpha[\mathbf{p}(q) - \mathbf{w}_i(q - 1)]$$

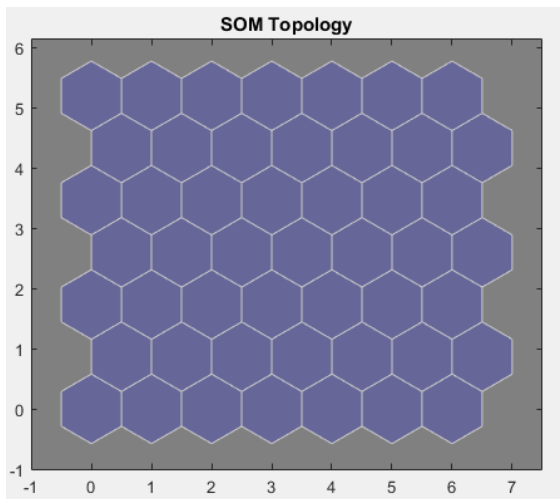
5.24 ábra: Egy győztes neuron súlyának változása

A képletben a w jelöli a súlyt, a q jelöli a hozzárendelések (az epoch) számát, a p jelöli a tulajdonságvektort, az i pedig jelöli, hogy hányadik neuronról van szó. Emellett van még egy α tényező, ami azt határozza meg, hogy a neuron mostani súlya és a megnyert elem különbségének hányad részével javítsa meg a jelenlegi súlyt.

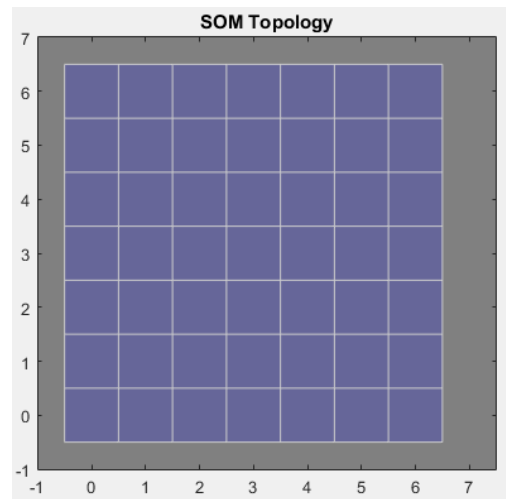
Ezek után jön be a hálózat és annak a topológiájának a szerepe, ugyanis nem csak az adott neuron erősödik, hanem a vele kapcsolatban lévő szomszédos neuronok is. Természetesen a szomszédok súlya nem olyan határozottan változik meg, mint a győztes neuroné, hanem valamilyen lecsengő függvény szerint, amely figyelembe veszi a neurontól való távolságot. Vagyis minél messzebb van a két neuron egymástól, annál kisebb változást fognak kapni. Ezek után tisztába kell tenni a topológia és a távolság fogalmát.

A topológia a neuronok szomszédsági viszonyát határozza meg. Ez esetünkben azt jelenti, hogy egyetlen neuronnak, hány darab szomszédja lesz. A Matlabban beépített topológiák találhatóak, amik egyszerűen a hálózatkészítő függvénynek paraméterként beadható. Alapvetően három fajta topológia közül választhatunk, ezek a „gridtop” (négyzethálós topológia), a „hextop” (hatszögalakú topológia) és „randtop” (véletlenszerűen generált topológia).

Munkámban az első két változatot alkalmaztam.

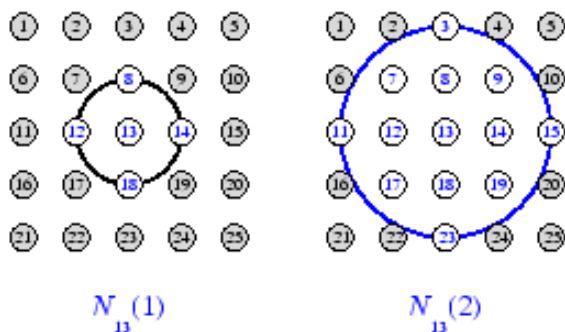


5.25 ábra: Hatszög topológia (hextop)

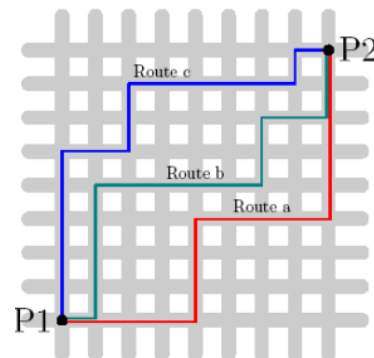


5.26 ábra: Négyzethálós topológia (gridtop)

A következő tisztázandó fogalom a távolság. A távolság számításának módszerére szintén választhatunk egy beépített listából. A Matlab a következő 4 választási lehetőséget kínálja föl nekünk: „dist”, „linkdist”, „mandist”, „boxdist”. Ezekből a feladat megoldása során a „linkdist” és a „mandist” távolságdefiníciókat használtuk fel. A linkdist a két neuront összekötő élek számával jellemzi a távolságot, míg a mandist a Manhattan-féle távolságdefiníciót foglalja magában.



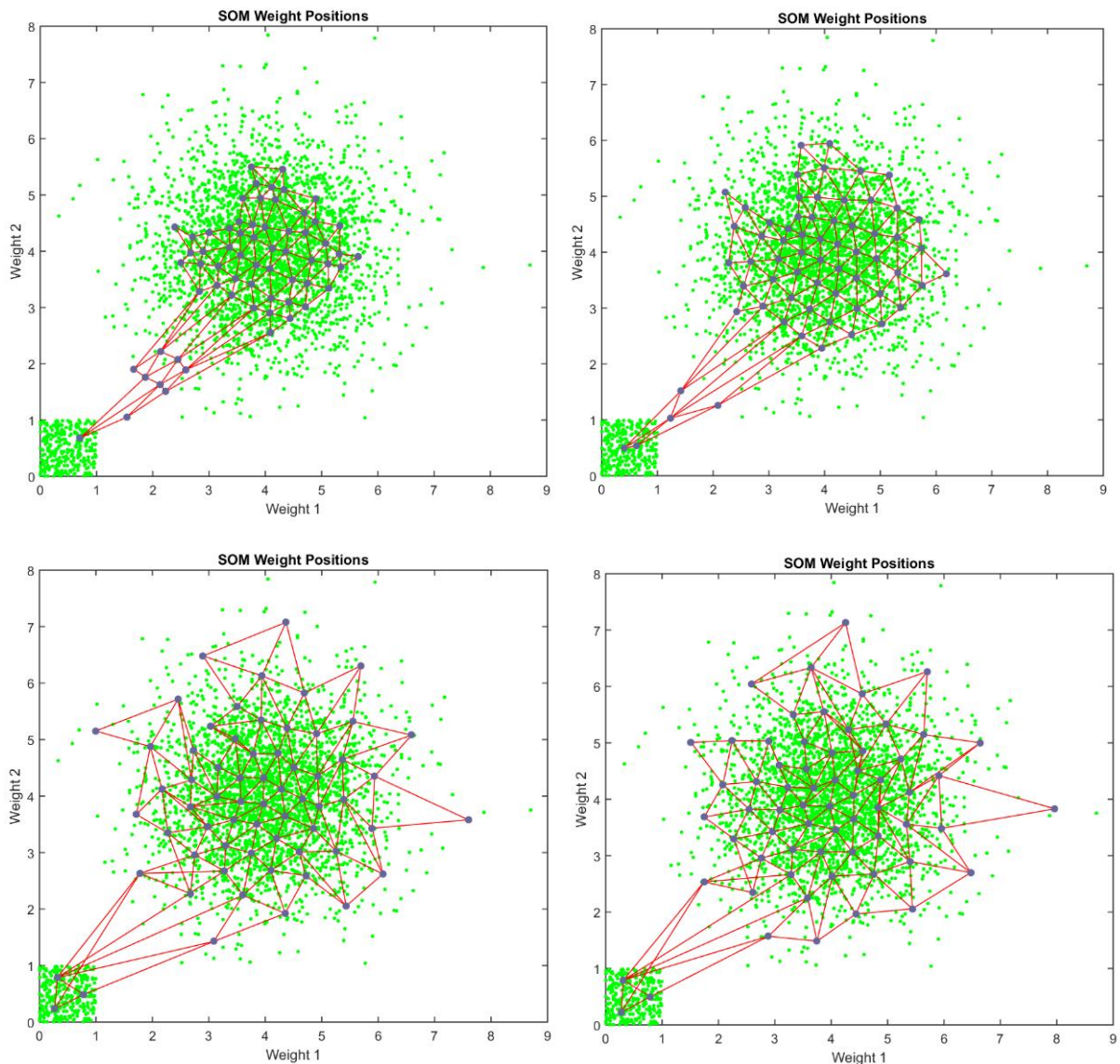
5.27 ábra: A linkek távolsága (linkdist)



5.28 ábra: Manhattan távolság

Miután tisztában vagyunk az alapvető fogalmakkal, neki is kezdetünk az adatok feldolgozásának. Még egy apró megjegyzés, hogy a neurális hálózatoknál az inputadatot tartalmazó mátrixot, a korábban említett tulajdonságmátrix transzponáltjával kell elindítani. A függvény a sorokban kéri a tulajdonságokat és az oszlopokban az elemeket.

Mielőtt bemutatnánk a futtatásból kapott eredményeket, érdemes megnézni egy kétdimenziós példát amelyen jól szemléltethető a hálózat és a hálózat alakulása bizonyos iterációként. Vegyünk egy hasonló pontfelhőt, mint amelyet a korábbi osztályozási módszernél használtunk, majd nézzük meg, hogy alakulnak az eredmények.



5.29 ábra: A hálózat terjedése az iterációk előrehaladtával (50, 100, 150, 200 iterációk utáni állapot)

Ennél a módszernél is több fajta megoldás közül választhatunk. Ezek az önszerveződő hálózatok úgy épülnek fel, hogy először végre kell hajtani egy tanítási folyamatot. Ezen folyamat során az adathalmazra ráilleszti a korábban meghatározott hálót, majd történik egy szimuláció, amely során az elemezni kívánt adatot kell beadnunk és ennek az eredménye lesz a végső osztályozás.

Alapvetően ennek a lépésnek az eldöntéséhez is két irányt szemlélhetünk. Vagy vesszük az előbbi osztályozásban is felhasznált mátrixot, azt berakjuk, mint tanuló anyagot, majd a szimulációt az összes kép összes képpontján végigfuttatjuk. Az eredmények elemzéseként pedig előállítjuk ugyan azokat a hisztogramokat, amiket korábban is használtunk. A másik lehetséges megoldás, hogy úgynevezett *in-sample* tesztet hajtunk végre, amely mintán belüli osztályozást jelent. A klaszterek kialakításához szükséges tananyag felhasználásával végezzük el a szimulációt és állapítjuk meg a felismerés minőségét. Ezáltal megkapjuk azt, hogy az osztályozás és a mintavételezés mennyire volt sikeres.

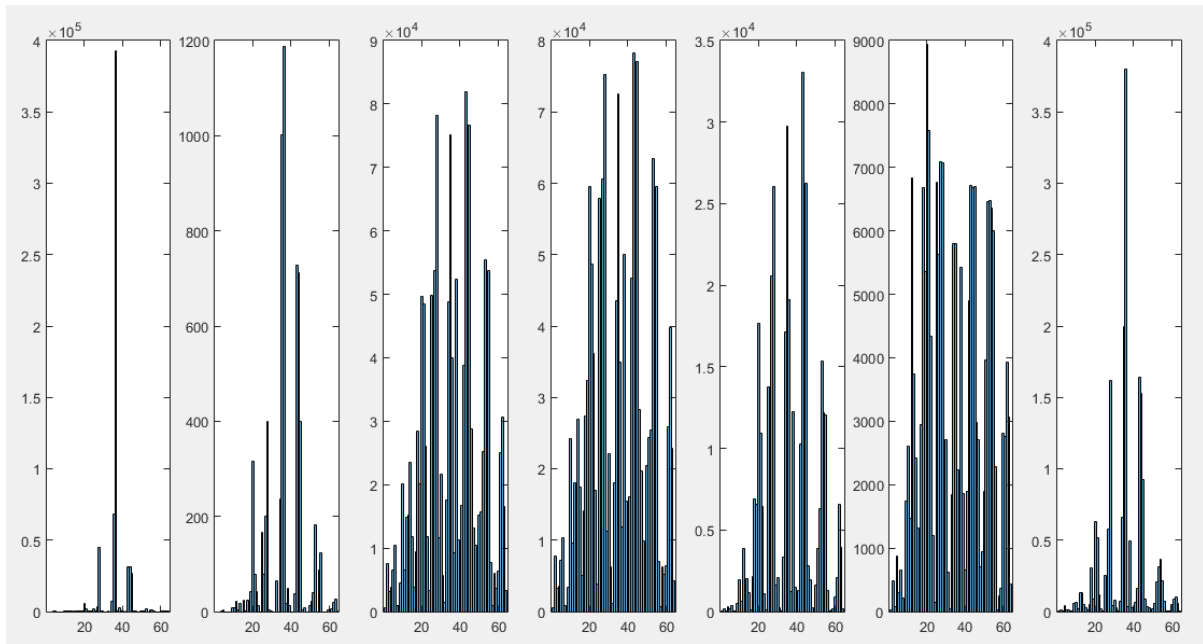
A feladat során mi ezt az utóbbi lehetőséget választottuk. A magyarázat pedig a következő: hiába nagyon gyorsak és hatékonyak ezek az önszerveződő hálózatok, ilyen nagy adatmennyiségnél már igen komoly erőforrásokat igényel egy-egy iterációnak a végigfuttatása, nem beszélve arról, hogy az iterációk száma a legtöbb esetben akár a több százat vagy ezret is meghaladhatja. Ezek elég indoknak bizonyultak ahhoz, hogy az első módszert teljes mértékben elvessük.

A mintán belüli osztályozást pedig úgy képzeljük, hogy van a 33 oszlopból és több millió sorból álló tulajdonságmátrix, ebből mintavételezéssel kivesszük minden századik elemet, azt feltételezve, hogy ez már egy megfelelően reprezentatív eredményt fog adni számunka. Lefuttatjuk a tanulási folyamatot, majd az eredeti több millió soros mátrixot beadjuk a szimulációba és eredményként kijönnek az osztályozott elemek. Ezt az eredményt hisztogramok segítségével ábrázolni tudjuk, és szemléléssel el tudjuk dönteni, hogy tényleg a várt eredményt kaptuk-e, vagy esetleg valami anomália lépett fel az egész rendszerben. Ha esetlegesen tényleg fellépne valamiféle anomália, akkor a bemenő paramétereken módosíthatunk, ezáltal változtatva a végeredményen. A következő paraméterekkel tudunk módosítani az eredményen: megváltoztathatjuk a topológiánkat, változtathatjuk a távolság meghatározási módszerünket, megváltoztathatjuk a neuron számot, az iterációk számát, illetve növelhetjük a tanulási folyamatban résztvevő adatok számát.

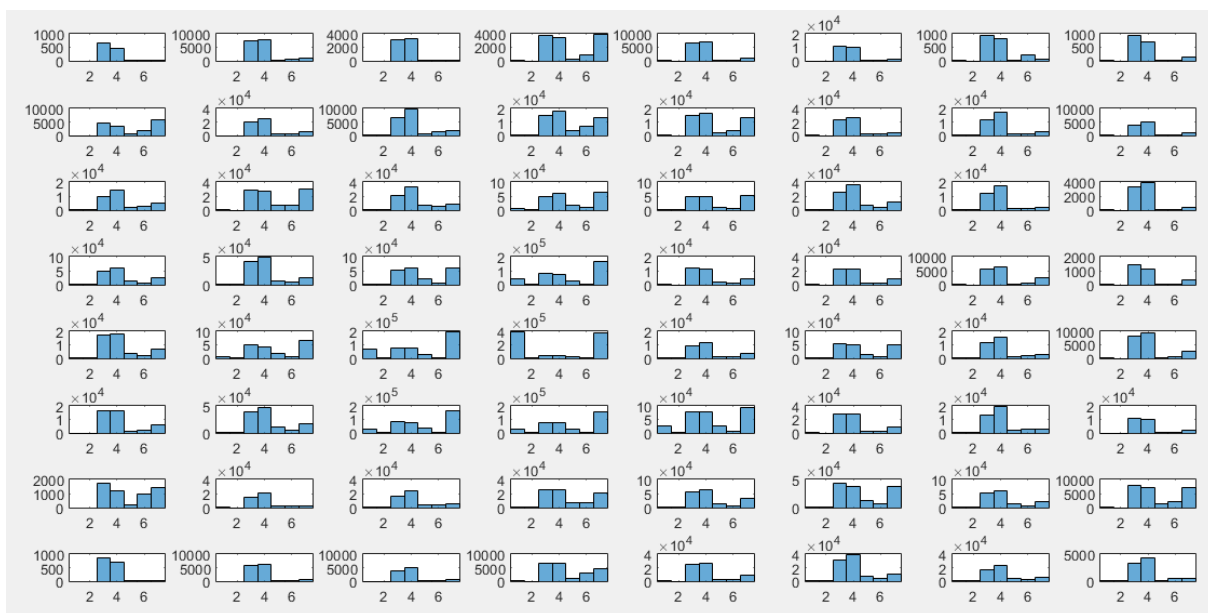
Látható, hogy nagyon sok kombináció lehetséges az input oldalon, ami nagymértékben megváltoztathatja a kimenő adatokat. Azt látjuk előre célszerűnek, hogy ha mégis negatív eredmény jönne ki, akkor csak vesszük a kódunkat, egy bemenő paramétert megváltoztatjuk, és újra lefuttatjuk az egészet. Ezáltal módszertani megállapításokat is le tudunk vonni az egyes metódusok között.

Már az egész folyamattal tisztában vagyunk, el lehet kezdeni a kód futtatását. Bemenő paraméternek egy 8×8 -as méretű és négyzethálós topológiát, a korábban már említett „linkdist” távolsághatározást, és egy 300-as iterációt állítottunk be. A bemenő adatmennyiség pedig a tulajdonságmátrix 1%-a volt.

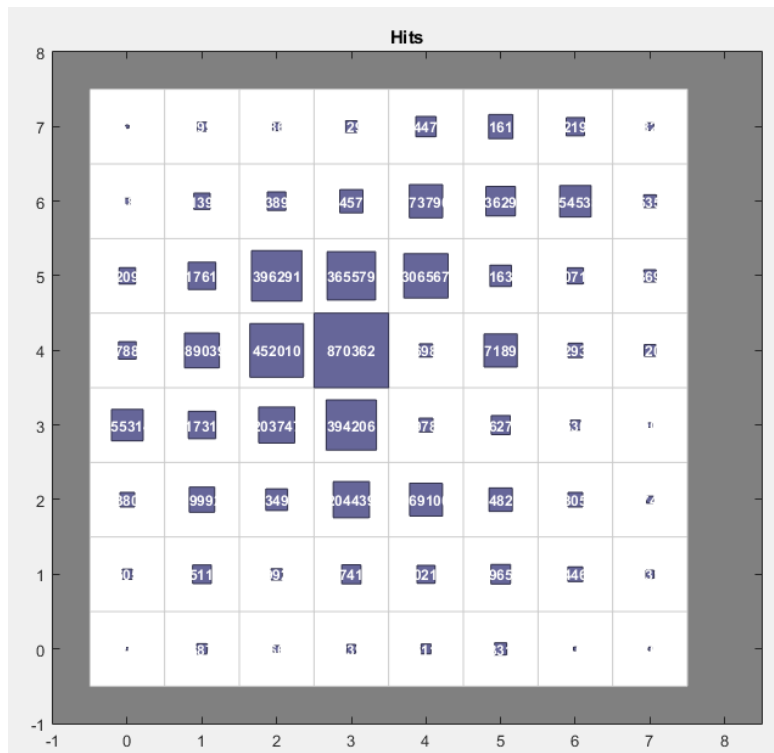
Az eredmény ábrázolásánál a *kmeans* osztályozásnál is használt két fajta hisztogramot használtuk. Amikor is az egyik ábra azt mutatta, hogy az egyes szövetek milyen neuronokhoz lettek besorolva, illetve milyen mennyiségben. A másik esetben pedig azt vizsgáltunk, hogy az egyes neuronok mely szövetekhez lettek hozzárendelve.



5.30 ábra: Az egyes szövetek, mely neuronoktól tartalmazznak elemeket



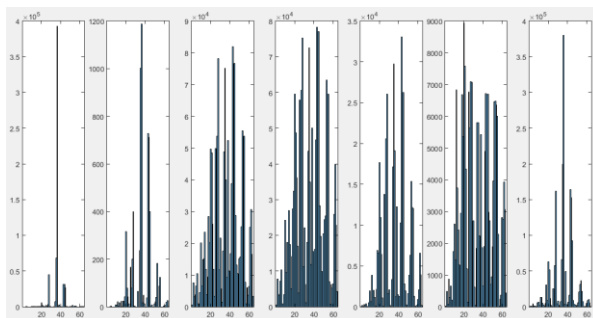
5.31 ábra: Az egyes neuronok, mely szövetekből tartalmazznak elemeket



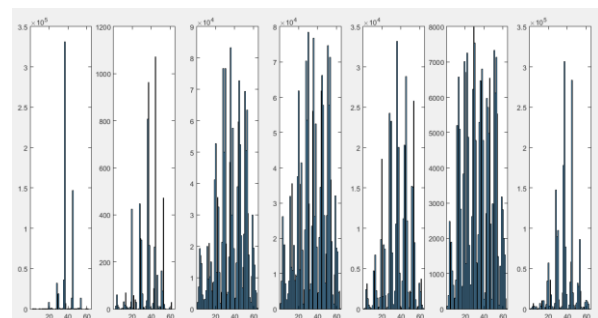
5.32 ábra: Az egyes neuronok között milyen megoszlásban vannak az elemek

Az eredményből látszik, hogy ez sem azt mutatja, hogy ebből egyértelműen azonosítani tudjuk a szöveteinket. Korábbi, nem bemutatott eredményekből már tapasztaltuk, hogy amikor egy 5×5 -ös méretű hálót adtunk be paraméterként, akkor a középső neuron kimagaslóan sok elemet kapott az osztályozás során, ezért gondoltuk, hogy egy nagyobb hálót fogunk alkalmazni. De még itt a 8×8 -as méretű hálónál is meg lehet figyelni ezt a csomósodást a 4-es sor 3-as oszlopánál. Ezt lehetne finomítani azzal, hogy nagyobb méretű hálót alkalmazunk, esetleg az iteráció számát megnöveljük, azzal a feltételezéssel, hogy a 300 iteráció még nem konvergált megfelelően a tökéletes végeredményhez.

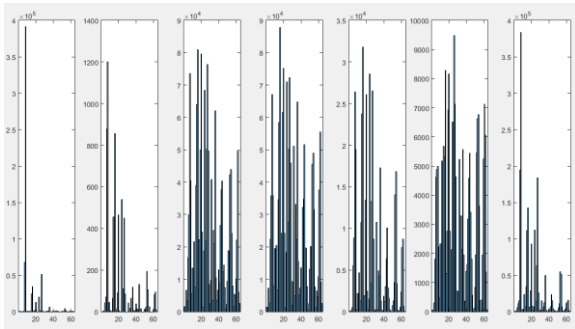
Már ebből az egy eredményből láthatjuk, hogy nem lesz egyszerű eljutni addig a pontig, amikor azt a megállapítást tudjuk levonni, hogy egyértelműen azonosíthatók a pontjaink. Így most mondjuk azt, hogy veszünk még néhány kombinációt, és amelyek a leginkább kedvező eset, azt lefuttatjuk egy viszonylag nagy iterációs számmal és meglátjuk, hogy ez mit fog eredményül adni.



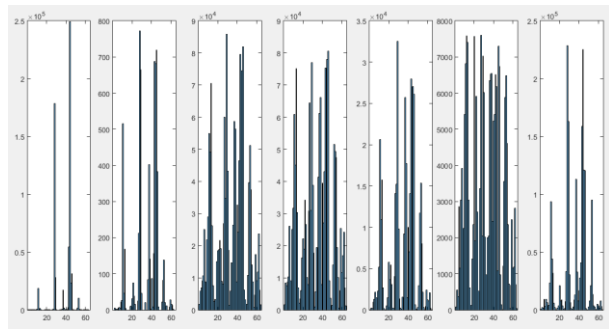
5.33 ábra: Linkdist, gridtop



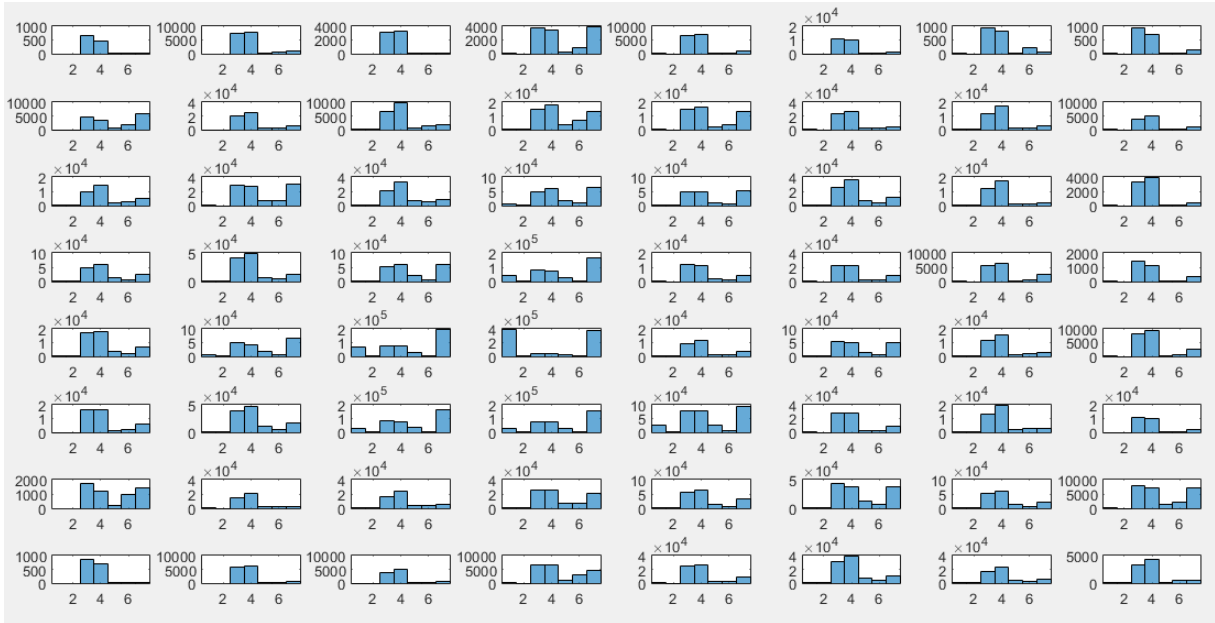
5.24 ábra: Mandist, gridtop



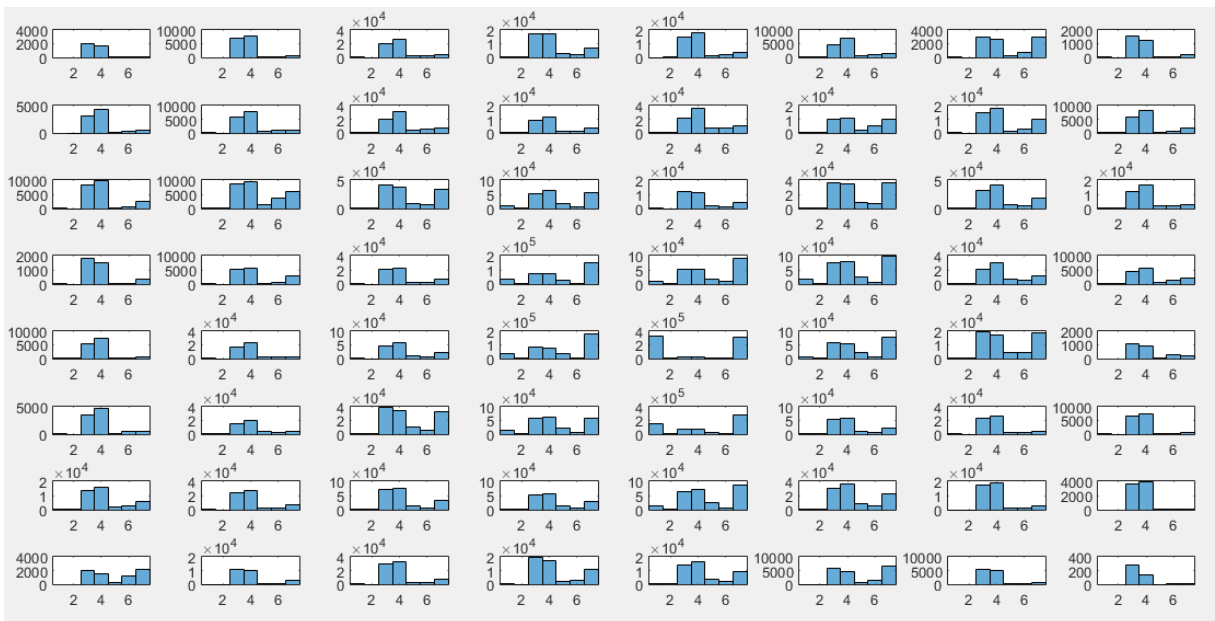
5.35 ábra: Linkdist, hextop



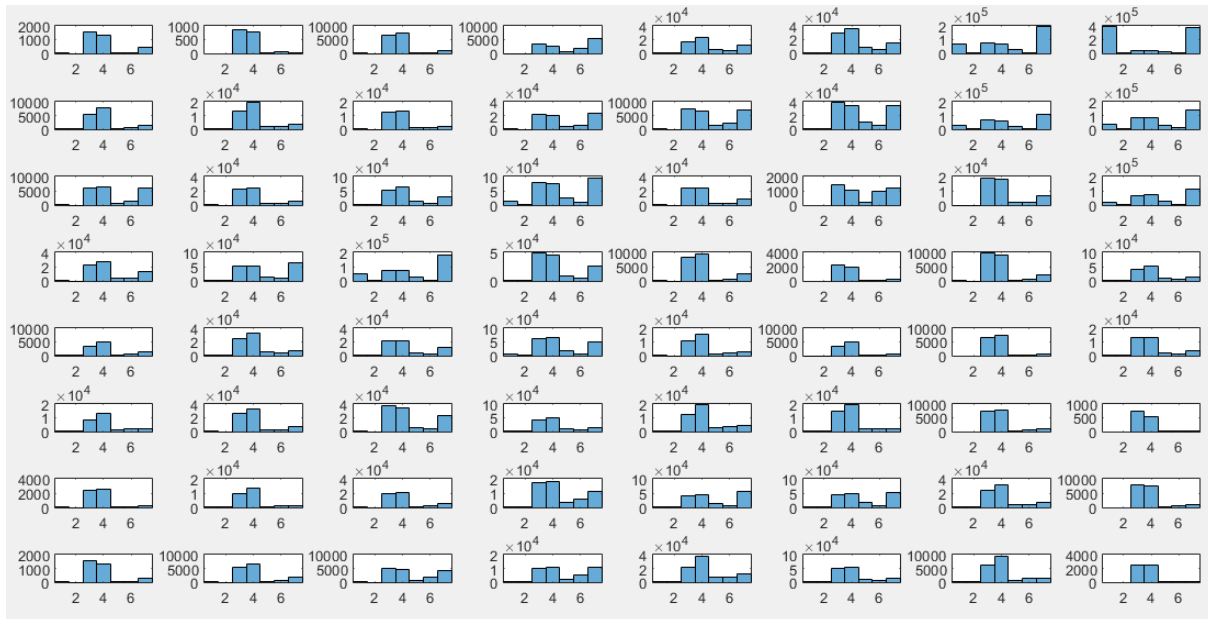
5.36 ábra: Mandist, hextop



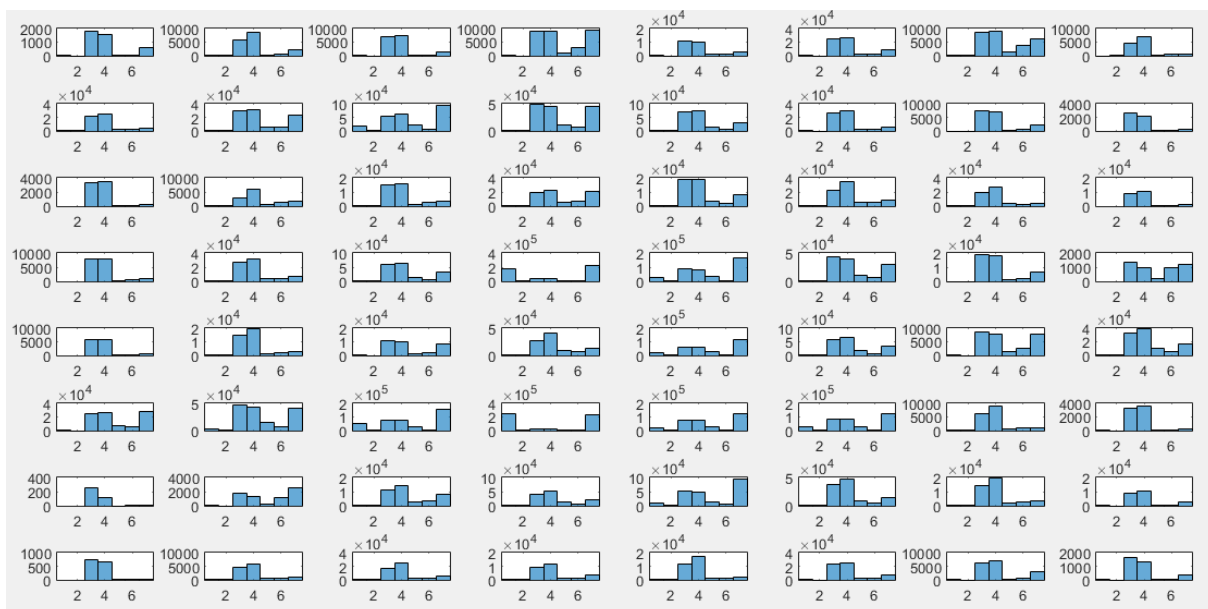
5.37 ábra: Linkdist, gridtop



5.38 ábra: Mandist, gridtop



5.39 ábra: Linkdist, hextop



5.40 ábra: Mandist, hextop

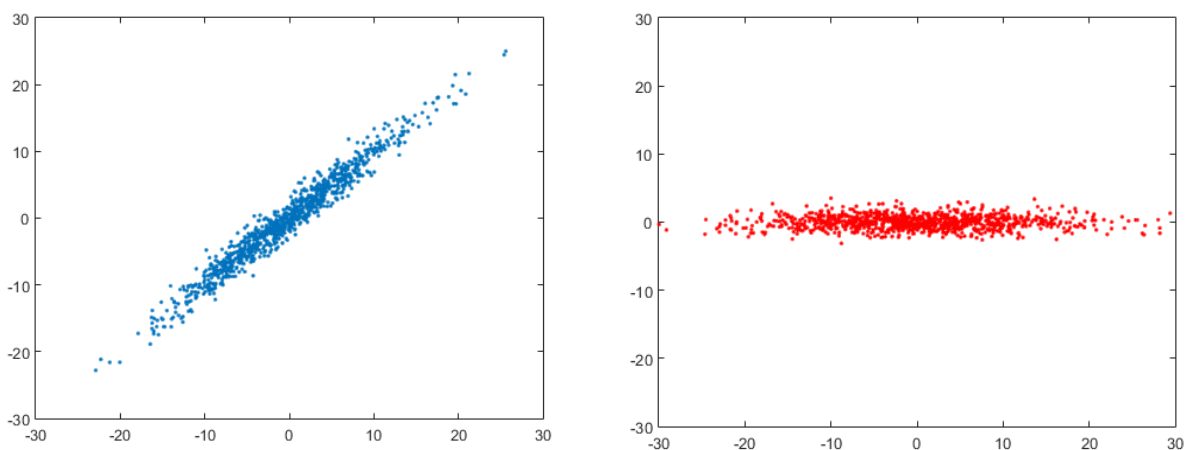
Előállt az összes általunk kipróbált kombináció. Látható, hogy az egyes neuronokhoz, vagy éppen szövetekhez rendelt elemek változtak, de összességében az eredmény körülbelül ugyan az. Ez többféle szempontból kifolyólag lehet így. Lehet, hogy kevés volt az iteráció és nem tudott kellő pontossággal ráilleszkedni erre a többdimenziós pontfelhőre, de lehet az is a probléma, hogy a pontok annyira egy halmazban helyezkednek el és olyan sok az átfedés, és ezáltal az elkülönítés nem lehetséges bármely módszert is választjuk.

Az elmélet igazolására vagy éppen a cáfolására indítunk egy nagyon nagy iterációs számmal rendelkező futtatást, amely ötletet egyébként egy közösségi videó megosztó portálon található videó inspirálta (<https://www.youtube.com/watch?v=aQklg69ZAXs>). A videó szemlélteti, hogy iterációról iterációra hogyan változik a hálózat. Azon példa esetében olyan

Az 5.41 ábrára ránézve láthatjuk, hogy hiába a nagy ismétlésszám, az eredmény körülbelül ugyan azt a tematikát adja vissza. Az eddigi eredményekből úgy tűnik, hogy ezeket a különböző „intenzitás-lejtés” adatokat nem tudjuk felhasználni úgy, hogy ebből egy megfelelően pontos osztályozást végre tudjunk hajtani.

5.2.1.6. Főkomponens analízis (PCA)

A neurális hálózat tanításában a bemeneti dimenziószám csökkentésére és a tanulási sebesség növelésére a szakirodalom a főkomponens-elemzést és –transzformációt javasolja (Duda et al. 2009). A főkomponens transzformáció felfogható egy olyan koordináta-rendszer forgatásnak, ahol tengelyeket úgy választjuk meg, hogy minden tengely a lehető legnagyobb variációval rendelkezzen. Ez számunkra azért lenne jó, mert az adatok rendezettségétől függően a 33 dimenziós mátrixunkat redukálni tudnánk 3-5 dimenzióra, ami sokkal könnyebben kezelhető, illetve 3 dimenzió esetén még ábrázolható is. Nem melleleg az eredeti információtartalmat is viszonylag jó közelítéssel reprezentálná.

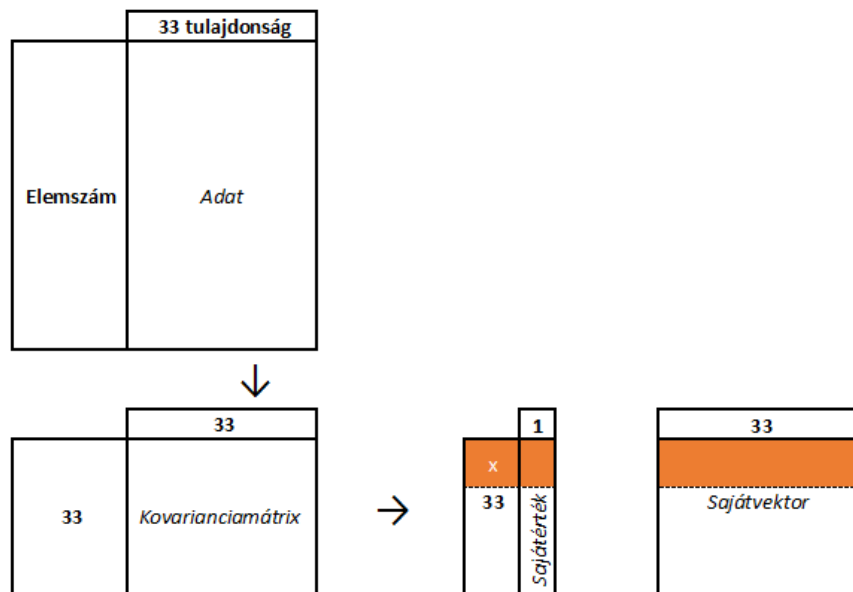


5.42 ábra: Az eredeti ponthalmaz (balra) és a transzformált állapot (jobbra): látható, hogy a transzformált állapot esetében a függőleges tengelytől kicsi, a vízszintes tengelytől pedig nagy a függése a ponthalmaznak

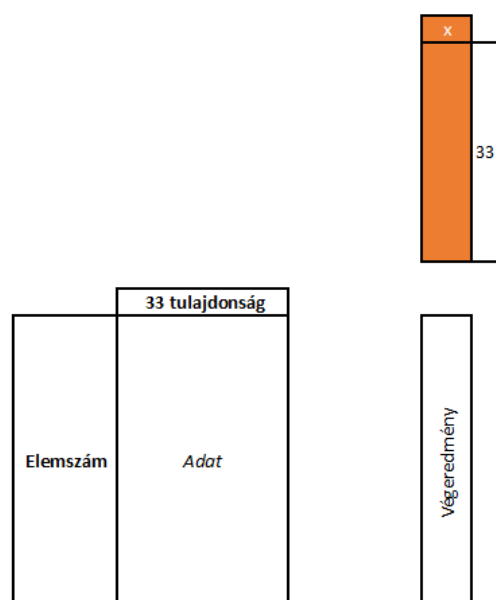
Matematikai értelemben a folyamat abból áll, hogy az adatmátrixból előállítjuk annak kovariancia mátrixát, amelyből kiszámíthatók a sajátértékek és sajátvektorok. A sajátértékek mutatják, hogy az egyes tengelyek mekkora hányadát tartalmazzák a variációnak, vagyis mekkora a szóródásuk. Ami értelmezhető úgy is, hogy ha az adott tengely körül nagyon kicsi a szóródás, akkor azon tengely mentén a ponthalmazunkat egy vonal mentén látnánk elhelyezkedni, ami azt is jelenti, hogy nem hordoz számunkra releváns adatokat. Viszont ha a szóródás nagy, akkor ahhoz a tengelyhez tartozó adatok sok információval rendelkeznek az eredeti adathalmazra nézve. Tehát ha a sajátértékeket normalizáljuk vagy vesszük a százalékos értéküket, akkor megláthatjuk, hogy az aktuális tengely milyen mennyiségű információt tartalmaz. Ha veszünk egy kumulált összeget, akkor rögtön azt is láthatjuk, hogy felülről haladva, addig a tengelyig összesen, mennyi információ maradt meg az eredeti adatmennyiségből.

Ha sikerült eldöntenünk, hogy végeredményként hány dimenzióban szeretnénk látni az adatainkat, akkor annyi sajátvektort kell venni, majd ezzel a mátrix-szal – mint együttható-

mátrix-szal - kell megszoroznunk az eredeti adathalmazunkat. Ebből már megkapjuk a végeredményt, ami annyi sorból fog állni, amennyi az elemszámunk volt és annyi oszlopból, ahány dimenzióban szeretnénk látni az adatainkat. Az alábbi ábrán ezt a dimenziószámot x -szel jelöltem. Ez az érték 1-től 33-ig bármilyen értéket felvehet esetünkben. Általános esetben egy n dimenziós halmaz esetén az x értéke maximálisan n lehet. Ha ez az $x = n$, akkor a koordináta-rendszer egy ekvivalens transzformáción esik át, amely nem tartalmaz semmiféle elhanyagolást.



5.43 ábra: A PCA működési elve. (x jelöli a dimenziószámot, amit szeretnénk a végén kapni)



5.44 ábra: Az eredeti adathalmaz transzformációja a sajátvektorokkal

Látható, ha egyszer előállítottuk a sajátvektorokat és a sajátértékeket, akkor utána már csak rajtuk áll, hogy ezekből az adatokból mennyit használunk fel. Ennek az eldöntése azért jó, mert össze tudjuk hasonlítani, hogy kisebb nagyobb adatmennyiségek kiválasztásával, mennyivel változik meg annak információtartalma.

A gyakorlati megoldás szempontjából azt szeretnénk, hogy a most meglévő tulajdonságmátrixot minél tömörebben le tudjuk írni a megfelelően jellemző információtartalommal. Előzetesen azt mondjuk, hogy három esetet szeretnénk vizsgálni és ebből a legjobbat kiválasztani. Az egyik az lenne, hogy a sajátértékekből levezetett kumulált százalékos érték 90% körül legyen, majd következőnek ez az érték 95% körüli legyen. Végül pedig egy olyan vizsgálatot is szeretnénk csinálni, ahol az első három sajátértékhez tartozó sajátvektort vesszük.

Az előálló eredménymátrixokat, mint korábban, beadjuk a neurális hálózatnak és amelyek a legjobb közelítéssel visszaadja a korábbi eredményeket, azt fogjuk eltárolni.

5.2.1.6.1. 90%-os információtartalom

Ahhoz, hogy eldöntsük, hol található ez a 90%-os küszöb, a sajátértékeket használjuk föl. Ezen sajátértékek, bármilyen értéket felvehetnek, ezért azt kell meghatározni, hogy az egyes értékek, mekkora részét (százalékát) teszik ki az összes sajátértéknek. Ekkor megkapjuk az egyes sajátértékek részarányát. Ha ezeket a részarányokat sorban összeadogatjuk, akkor megkapjuk, hogy az éppen aktuális elemünkkel együtt, összesen mekkora információtartalom birtokában vagyunk.

	1
1	314.8388
2	215.9204
3	100.3396
4	32.1678
5	23.9324
6	13.4521
7	12.2438
8	8.2061
9	8.1370
10	6.1744

	1
1	41.1320
2	28.2089
3	13.1088
4	4.2026
5	3.1266
6	1.7574
7	1.5996
8	1.0721
9	1.0631
10	0.8067

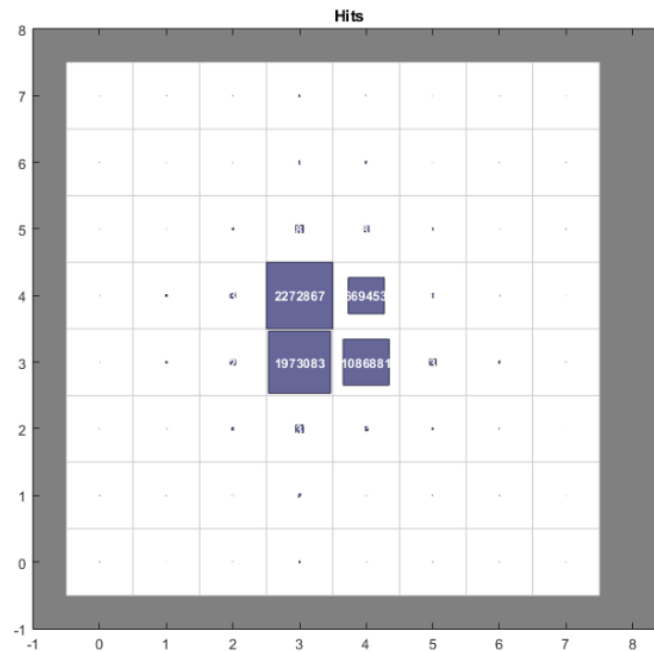
	1
1	41.1320
2	69.3409
3	82.4497
4	86.6523
5	89.7789
6	91.5363
7	93.1359
8	94.2080
9	95.2711
10	96.0777

5.45 ábra: Az átalakítás művelete (balra: sajátértékek, középen: százalékos részarány, jobbra: kumulált százalékos összeg)

Így az 5.45 ábra harmadik oszlopából könnyen meg tudjuk állapítani, hogy nekünk hány sorra is lesz szükségünk. Ahhoz, hogy közel 90%-os információtartalom maradjon meg nekünk, ahhoz az első öt sorát kell majd venni a sajátvektornak és azzal kell transzformálni az eredeti tulajdonságmátrixunkat. Ebből előáll egy olyan mátrix, ahol a sorok száma a felhasznált pixelekével, az oszlopok száma pedig ötlet fog megegyezni.

Ezután elő kell állítani azokat a diagramokat, amelyeket korábban is megcsináltunk és az eredményeket össze kell vetni. Amennyiben ez megfelelő hasonlóságot mutat, akkor azt

mondhatjuk, hogy ez a főkomponens analízis hasznunkra fog válni, ha pedig nem, akkor meg kell nézni, hogy nagyobb információtartalom mellett mit mutat.

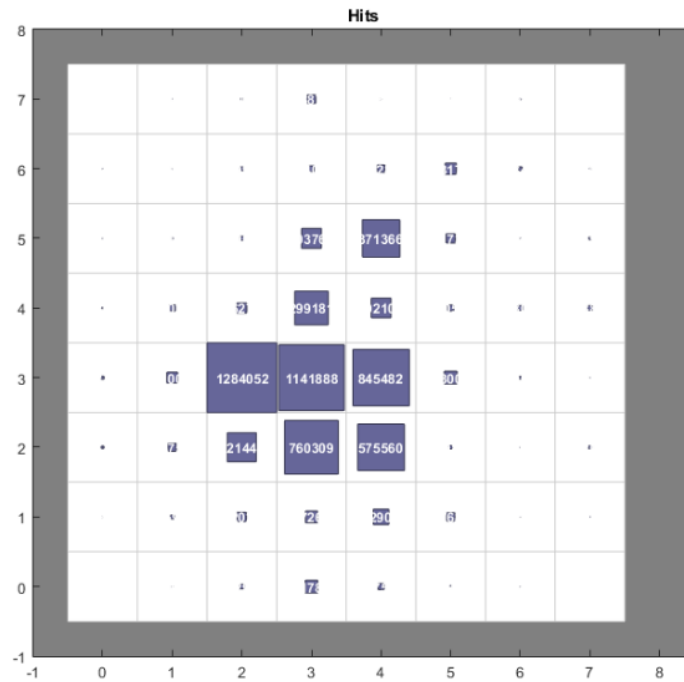


5.46 ábra: 90%-os információtartalmú transzformáció eredménye

Az első eredmény után nem is kell további diagramokat előállítani, ebből is le tudjuk vonni azt a következtetést, hogy ez nekünk nem lesz megfelelő. A neurális hálózati szimulációt ezres nagyságrendű iterációval hajtottuk végre, ami a korábbi eredmények alapján elmondható, hogy megfelelően konvergál a végeredményhez. Az 5.46 ábrán azt láthatjuk, hogy az egyes neuronok hány elemet kaptak az osztályozás során. Látható, hogy jelen esetben négy darab neuron elviszi az elemek nagyon nagy hányadát. Ez biztosan nem fogja visszaadni a korábban már látott eredményeket.

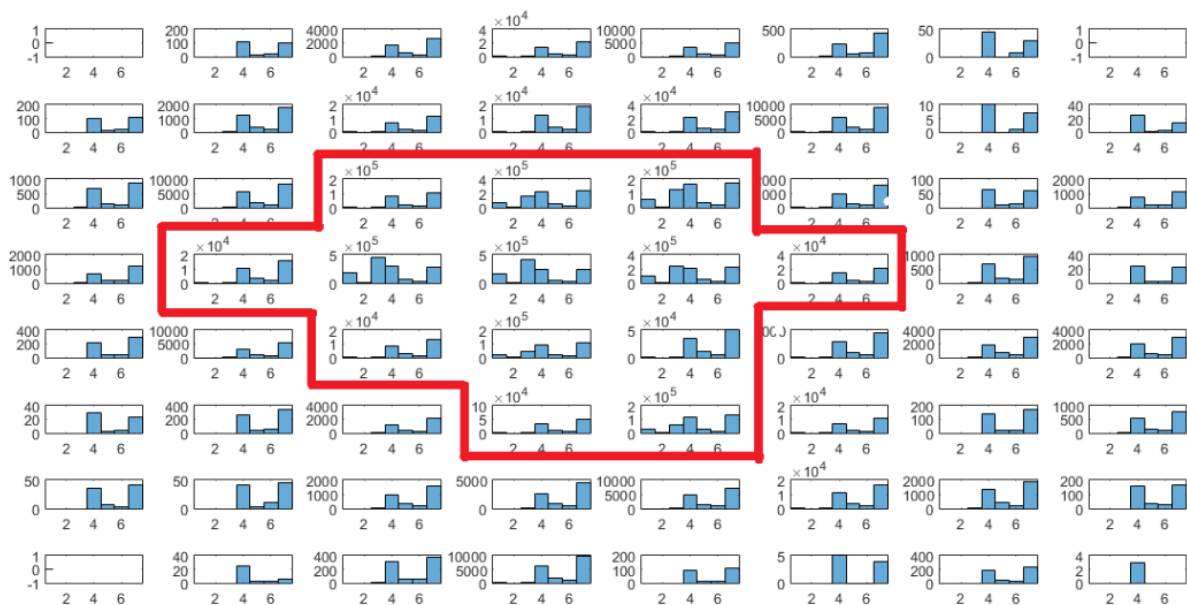
5.2.1.6.2. 95%-os információtartalom

A feladat nem más, mint az előbb, annyi különbséggel, hogy a kumulált százalékos értékek közül egy 95 körüli értéket keresünk. Az 5.45 ábra jobb oldali táblázatából jól látszik, hogy ez az első kilenc sor lesz. Vagyis az eredeti mátrixunkat a sajátvektor első kilenc sorával kell transzformálni. A módszer ugyan az, mint az előző esetben, így nézzük rögtön az eredményeket.



5.47 ábra: 95%-os információtartalmú transzformáció eredménye

A fenti ábrán látható, hogy itt már jobban eloszlanak az elemek a neuronok között, de még mindig nagyon koncentráltan fordulnak elő az elemek a középső neuronoknál. Valamint a szélső neuronok egyáltalán, vagy alig kapnak elemeket. Mivel hét különböző szövetet kívánunk megkülönböztetni egymástól, ezért van rá matematikai esély, hogy azonosítsunk egy-egy szövetet. Mindazonáltal a külső neuronokat el kell hagynunk, mert azok elemszáma olyan elenyésző, hogy azokból semmilyen további eredmény vagy azonosítás nem szűrhető le.



5.48 ábra: Az egyes neuronokhoz sorolt elemek szövetenkénti eloszlása

A pirossal kiemelt rész az, ahova az elemek nagy része be lett sorolva. Ez a rész, már jobban közelíti az eredeti adathalmazunkat, mint a korábbi eset, de osztályozásra még mindig alkalmatlan. Látható, hogy a szövetek egyes pixelai nagyon megoszlanak, nagyon szétszóródnak a különböző klaszterekben.

Elmondhatjuk, hogy a főkomponens analízissel nem tettünk előre olyan lépéseket, mint amikre számítottunk. 95%-os információtartalom mellett még nem adja vissza megfelelően (vagy esetleg jobban) a korábbi eredményeket, nem lehet vele jobban osztályozni és emellett kilenc oszlopot így is fel kellett használnunk, ami a vizsgálandó adatmennyiséget se csökkentette drasztikusan. A fejezet elején említett harmadik megoldást pedig ezen eredmények alapján nem is érdemes vizsgálni, hiszen ha az első kilenc sajátvektorral való transzformáció nem adott megfelelő eredményt, akkor belátható, hogy a három dimenziós változat se fogja ezt megtenni.

5.2.2. Az összes pixel automatikus osztályozása

Miután végignéztünk az ellenőrzött osztályozás lehetőségein, megnézhetjük, hogy ha a maszkokat nem tanulásra használjuk, hanem az eredmények ellenőrzésére akkor milyen megbízhatóságú osztályozást tudunk létrehozni.

Mint már korábban említésre került, ebben a fejezetben a kiindulási adataink maguk a képek. Jelen esetben a képek teljes terjedelmét kell valamilyen módon osztályozzuk, majd a leosztályozott képen el kell különíteni az ismert területeket és ezen területek elemzésével kell eldöntenünk, hogy a folyamat sikerrel járt avagy sem.

5.2.2.1. Képek osztályozása az RGB és YCbCr komponensek alapján

Ha az általánosan elterjedt fényképek világában gondolkodunk, akkor magától értetődőnek véljük, hogy a képeket az RGB intenzitásadataik alapján próbáljuk meg osztályozni. Mivel nem csak az RGB színmodell létezik, hanem sok más egyéb színmodellt is használnak a képmegjelenítés különböző területein, ezért úgy véltük, hogy valamilyen további színmodell komponenseit is bevonjuk az osztályozásba. A választásunk a YCbCr színmodellre esett, mely színmodell élesen eltér az RGB modell felépítésétől. Míg az RGB színmodell a különböző sávokban mért intenzitásértékeket tárolja, addig az YCbCr modell első komponensként a fényerőt tárolja, második és harmadik komponensként pedig az ehhez képest kék és vörös fényerejének eltérését (chroma). Tehát látható, hogy a két modell merőben eltérő módon tárolja a kép információit, amely eltérés az osztályozás során mérvadó lehet az eredmény szempontjából. Mellesleg a két színtér között az átjárás könnyedén megoldható. Az YCbCr színmodellen kívül választhattuk volna még az CMYK és a HSV/HSL színmodellt, de mivel most az egész képeket kell vizsgálni nem kívántuk túlterhelni a rendszert még további dimenziókkal. Első lépésként megpróbáljuk ezen hat komponens alapján elvégezni a kép osztályozását.

Az osztályozást a korábban már említett neurális hálózatokkal vagy a már szintén említett *kmeans* eljárással végezhetjük. Mivel jelen esetben is érdekes az, hogy mi történik, ha a képeket egyszerre, illetve ha egyesével vizsgáljuk, ezért most csak a neurális hálózatokat használjuk fel. A *kmeans*-el az a probléma, ha a képeket külön-külön osztályozzuk, akkor a végeredmény lehet jó, de nem lesz szemléletes. Mivel az elején a klaszterek súlypontjai véletlenszerűen kerülnek

elhelyezésre, ezért színekódolva a klaszterezett képek, ugyan azon részei más színűek lehetnek, noha a homogenitásuk (azonos osztályba tartozásuk) kimutatható.

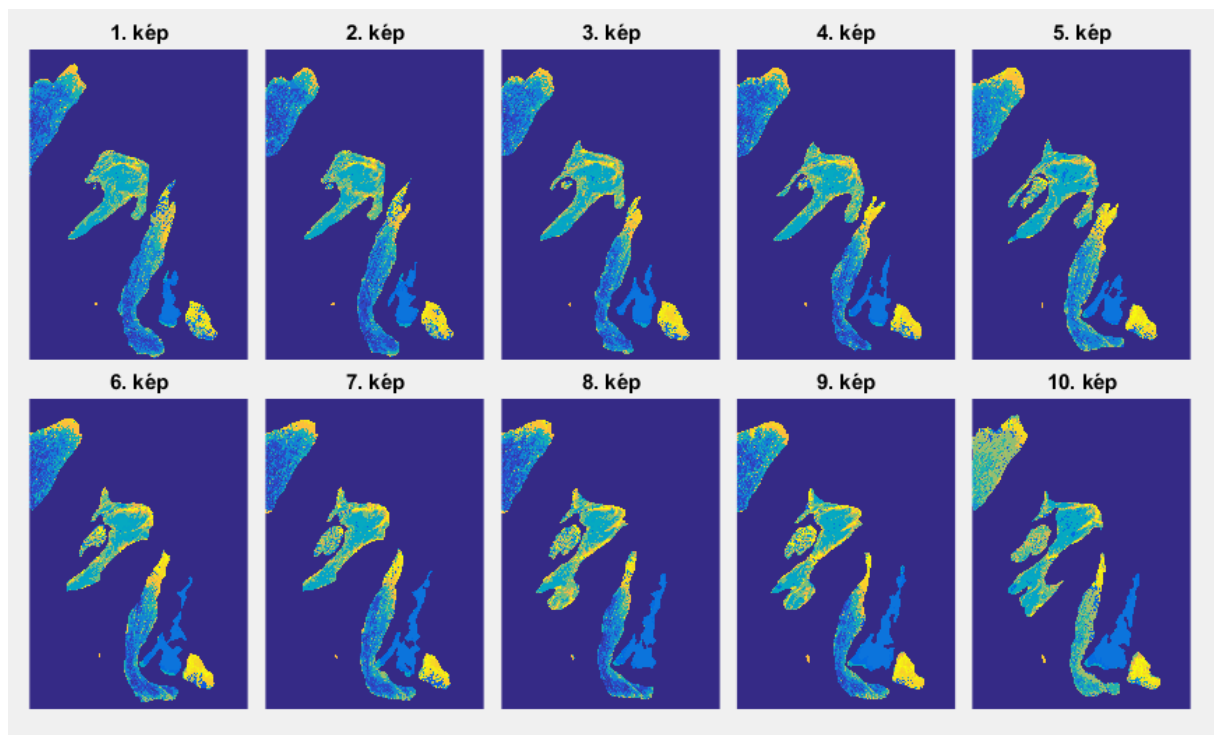
5.2.2.1.1. Képek osztályozása egységesen

A mostani esetben azt feltételezzük, hogy a képek ugyanolyan minőségben állnak rendelkezésre és a képek készítésének körülményei is változatlanok voltak. Így tudjuk azt mondani, hogy kezeljük egységesen a képeket. Vesszük az összes kép minden képpontját, elvégezzük az osztályozást, majd megnézzük, hogy a maszkok alatti területek mennyire egységesek.

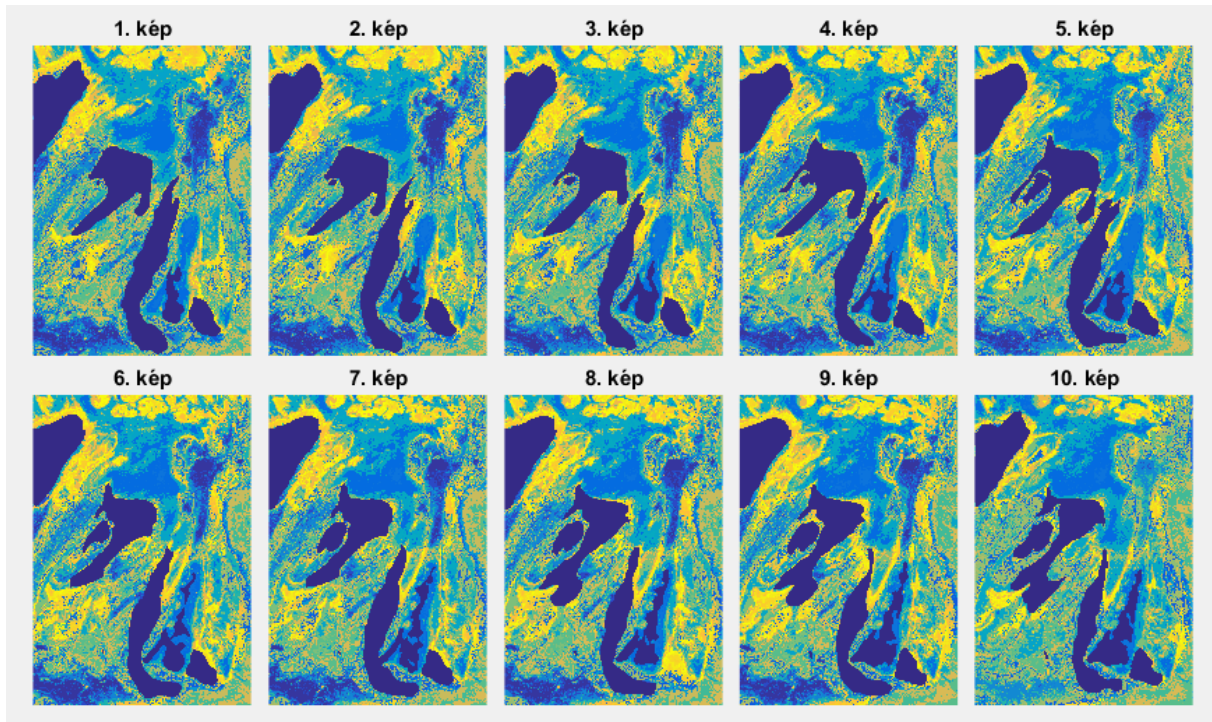
A korábbi ellenőrzött osztályozással ellentétben most nem csak a keresett szöveink jelennek meg az eredményben, hanem az összes számunkra nem releváns képpont is. Ebben az esetben nem lesz elegendő, ha egy szövetet egyértelműen lefed egy klaszter, hanem még annak is meg kell felelnie, hogy a keresett szöveten belüli klaszter ne forduljon elő a szöveteken kívüli területen. Amennyiben egy szövetet egyértelműen lefed egy klaszter és ez a klaszter nincs jelen más szövetekben, illetve a szöveteken kívül, akkor mondhatjuk el, hogy sikerült egy olyan rendszert létrehozni, amivel már elvégezhető a képeken az egyértelmű azonosítás.

A feladatot, mivel nagyon erőforrás igényes, mindössze egy 5×5-ös hálóval fedtük le és a tananyag is a 10 kép összes pixelének mintavételezésével mindössze tizede volt. (A futási idő így is órás nagyságrendű volt.) Majd a tanulási folyamatot 300 iterációval végeztük.

Jelen esetben nagyon szemléletes ábrát lehet készíteni az eredményről. Erről ránézésre meg tudjuk mondani, hogy a feladat megoldása szempontjából jó irányba haladunk vagy ezen a téren is kell valami bővítést hozzácsatolni a rendszerhez.

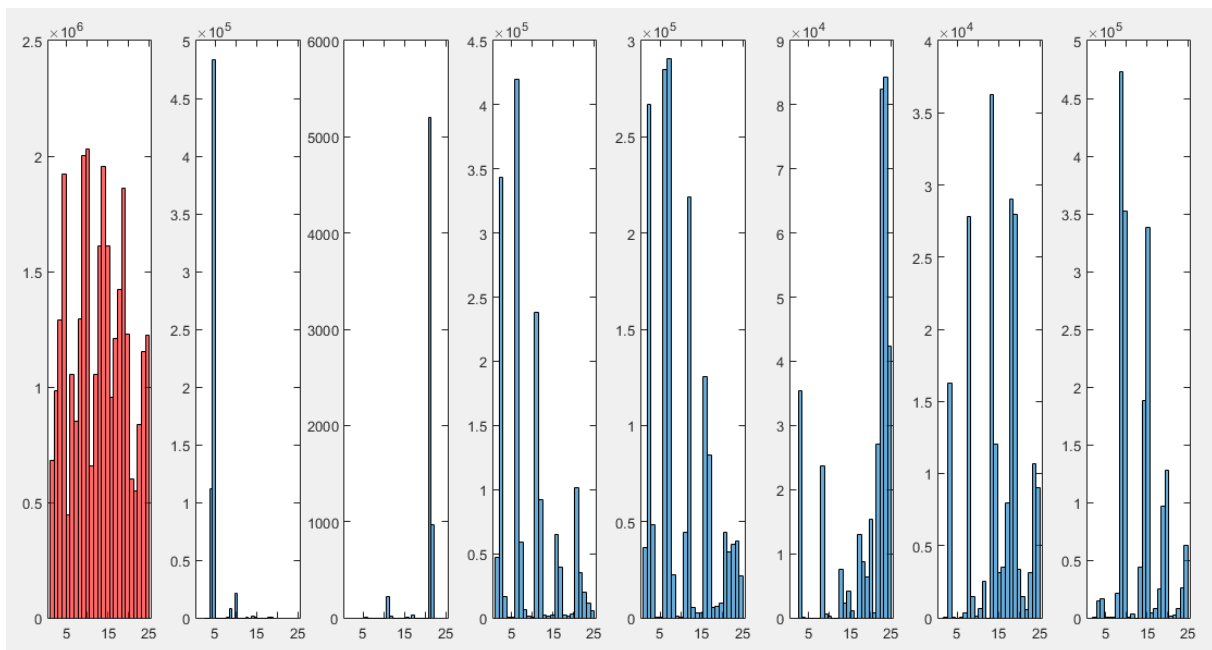


5.49 ábra: A szöveteken belüli klaszterek eloszlása színekódolva

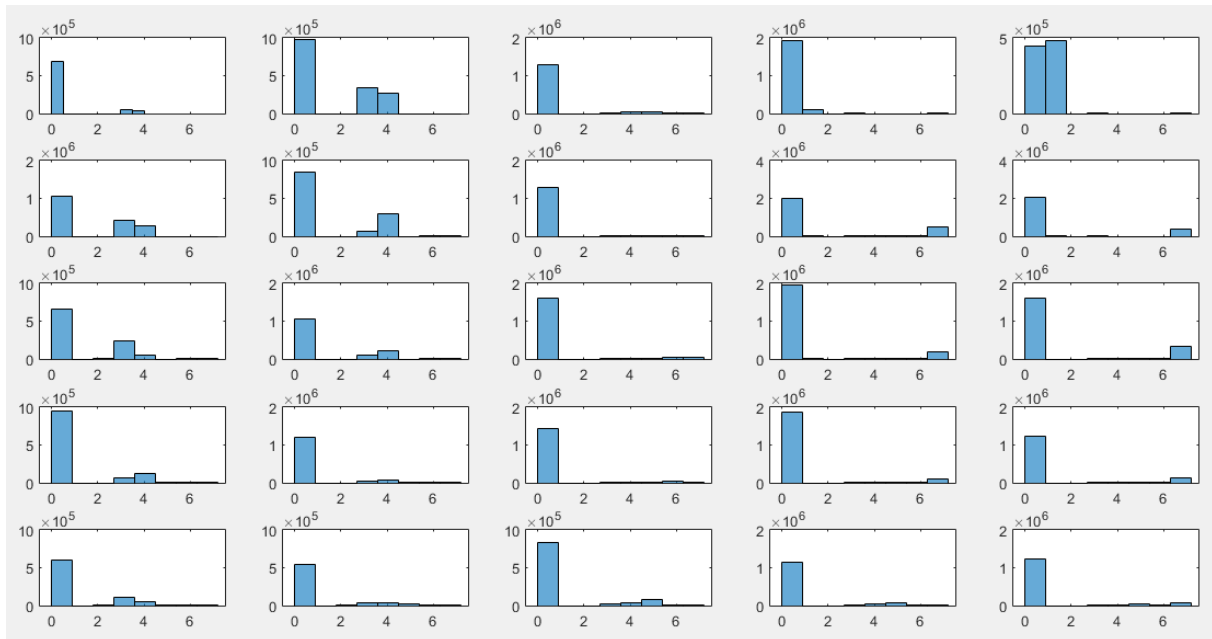


5.50 ábra: A szöveteken kívüli klaszterek eloszlása színekódolva

A fenti ábrák nagyon jól szemléltetik, hogy az osztályozásunk mely képpontokat vélte ugyan olyanak azonosítani. A szövetekre tekintve láthatjuk, hogy azért többé-kevésbé egy-két klaszter fed le egy-egy szövetet, ami az azonosítás felé biztató jeleket mutat. Viszont ránézve a szöveteken kívüli részre, az is látható, hogy ezen a részen igencsak változatosak az osztályok. Ezért érdemes itt is megnézni a korábban már használt statisztikákat és a pontos eredményeket ábrázolva vonjuk le végső következtetéseket a megoldással kapcsolatban.



5.51 ábra: A szöveteken belüli klaszterek eloszlása (bal oldali ábra: szöveteken kívüli rész)



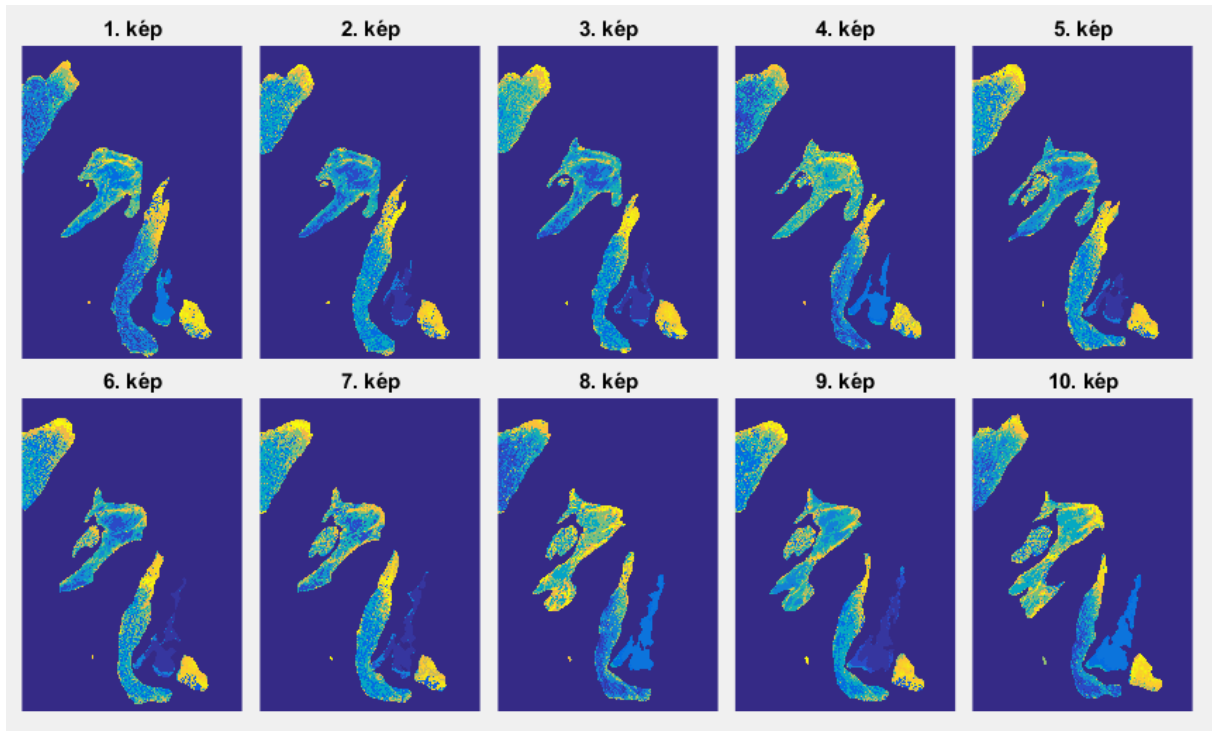
5.52 ábra: A klaszterek szövettartalma (bal oldali oszlopok: szöveteken kívüli rész)

Az utóbbi két ábra megerősíteni látszik azt, amit a színekódolt képeken is láthattunk. Ezek az ábrák kezdenek olyan alakúak lenni, mint amilyeneket mi szeretnénk látni. Az 5.51 ábrán látható, hogy minden egyes szövetnél 2-3 kimagasló érték található, ráadásul, ezek nem is ugyanabban az osztályban találhatóak. Ezt igazolja az 5.52 ábra, ugyanis látható, hogy minden egyes klaszternél 1-2 szövet található meg. A probléma az, hogy minden egyes klaszter tele van olyan képpontokkal, amely számunkra egyáltalán nem szükségesek. Ha most elvégeznénk egy esetleges osztályozást, akkor az eredmény az lenne, hogy a szövet mellett még nagyon sok olyan részt is szövetnek azonosítunk, ami egyébként nem az.

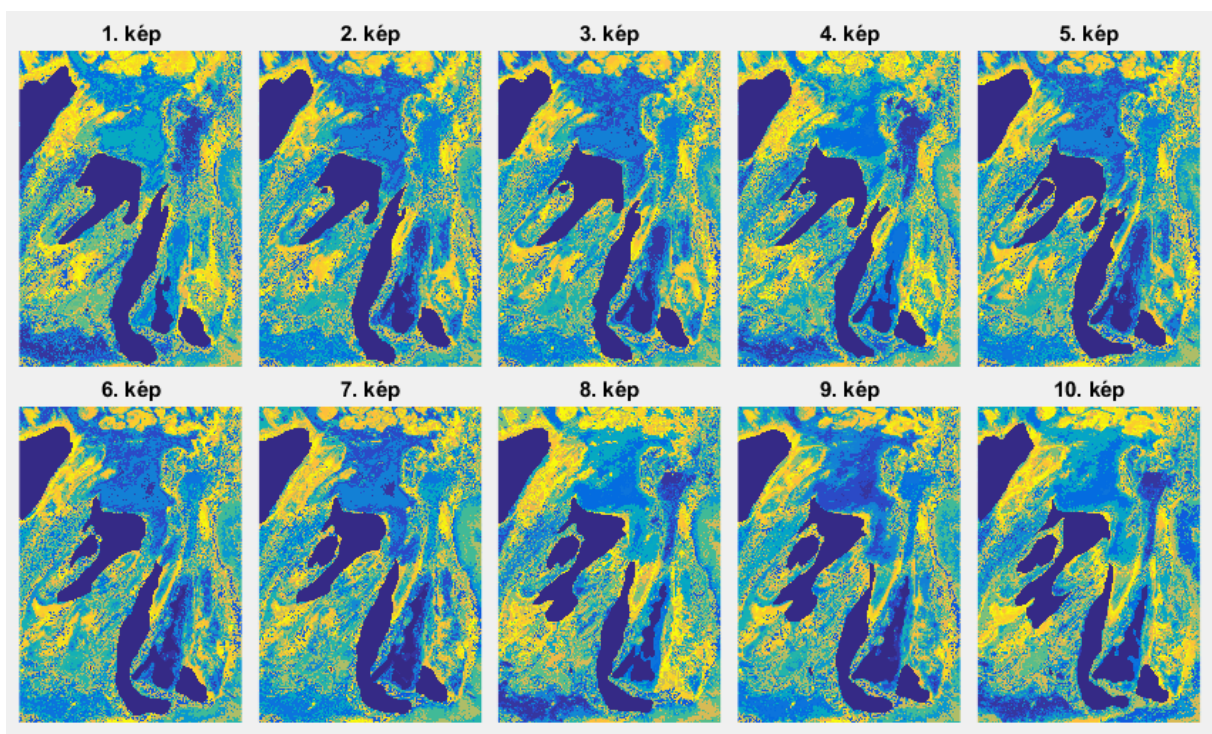
5.2.2.1.2. Képek osztályozása egymástól függetlenül

A mi szemszögünkből érdekes lehet megvizsgálni egy olyan esetet, hogy hogyan alakul az osztályozás abban az esetben, ha a képeket önmagukban osztályozzuk. A kívánatos eredmény az lenne, ha az ábrák ugyanazon részei nem, vagy csak minimálisan változnak. Ha a képpontok osztályba sorolása nagyon eltér, akkor lehet gyanakodni, hogy a képek készítésekor a körülmények változtak. Ezáltal le tudunk vonni egy olyan következtetést, hogy a képek készítésének módja megfelelő volt-e, vagy változtatni kell a felvételek elkészítésén. Ez egy fontos megállapítás lehet, mert ez befolyással lehet a korábbi osztályozások eredményeire is.

Az osztályozás ugyanazokkal a paraméterekkel készültek, mint a képek együttes kezelése esetén, annyi különbséggel, hogy mivel most egyszerre csak egy képet vizsgálunk, ezért a tananyagban bent hagytuk az összes pixelt, majd a szimulációba szintén ugyan ez az állomány került bele.



5.53 ábra: A képek külön-külön osztályozásának eredménye



5.54 ábra: A képek külön-külön osztályozásának eredménye (negatív)

Az eredményből látszik, hogy a képek külön-külön osztályozása nem mutat semmiféle nagyobb eltérést, ami azt jelenti, hogy a képek tekinthetők közel azonosnak. A negatívokat tartalmazó képeken néhány helyen fel lehet fedezni színárnyalat különbségeket, ami adódhat egy részből abból, hogy ezek a részek tényleg eltérnek a többitől, esetleg a visszavert fény volt

eltérő, másrésztől lehet hogy az osztályozás során az iterációk előrehaladtával a neuronok nem ugyanott helyezkedtek el és ezek a képpontok emiatt átkerültek egy másik neuronhoz.

Végkövetkeztetésképp azért elmondhatjuk, hogy ezen osztályozási forma és ezen változók felhasználásával nincs különösebb eltérés a képek között.

6. Jövőbeli tervek

Látható, hogy az elénk állított feladatra nem született olyan megoldás, amivel kellő bizonyossággal azonosítani tudnánk az adott szöveteket. Ezért vagy az eddigi meglévő eredményeket kell finomítani vagy valamilyen új módszereket kell bevezetni. Mivel láttuk, hogy nincs egy egzakt megoldása a feladatnak és nem tudjuk mi lesz a legcélravezetőbb, ezért érdemes elgondolkodni mind a két vonalon. A korábbiakat felhasználva vezethetünk le új tulajdonságokat a gradiensén kívül, esetleg a gradiensekkel vizsgált kört kibővíthetjük és nem csak a pixel közvetlen környezetét vizsgálhatjuk. Esetleg az automatikus osztályozásnál az RGB YCbCr tulajdonságokat még kibővíthetjük a HSV/HSL sávokkal vagy tanulóterületek közé bevonhatjuk egy „új szövetként” a számunkra elhanyagolható területeket.

Mondhatjuk azt is, hogy pusztán a színek alapján nem tudunk különbséget tenni a szövetek között, ezért valamilyen új megoldást kell keresni. Ebben az esetben további textúra-jellemzők vizsgálata látszik célszerűnek. Ilyen textúra jellemzők előállíthatók szűréssel (pl. Gabor-szűrőkkel), Fourier-transzformációval vagy az együtt-előfordulási mátrix (cooccurrence matrix) alapján levezetett textúra-mérőszámokkal (pl. entrópia, energia, homogenitás).

Felhasznált irodalom:

Agur, A.M.R.- Dalley, A.F. (2009): Grant's Atlas of Anatomy, 12th Edition, Kluwer

Duda, R.O. – Hart, P.E. – Stork, D.G. (2000): Pattern Classification, Wiley, New York

Mathworks Matlab User's Guide

Wikipedia