

Óbudai Egyetem
Neumann János Informatikai Kar
Alkalmazott Informatikai Intézet



Tudományos Diákköri Dolgozat

Titán-nitrit elektródák sejtkárosító hatásának vizsgálata patkányok digitalizált agyi metszetein

Szerző: **Bényi Zsolt Ferenc**
mérnök informatikus szak (BSc)

Konzulensek: **Dr. Vámosy Zoltán**
egyetemi docens
Dr. Wittner Lucia
tudományos főmunkatárs
MTA TTK

Budapest, 2014.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani Dr. Vámosy Zoltánnak és Dr. Sergyán Szabolcsnak a szakmai támogatásért és Dr. Wittner Luciának a lehetőségért, hogy egy számomra érdekes és izgalmas kutatásban vehettem részt.

Tartalomjegyzék

1.	Bevezetés	1
2.	Irodalomkutatás.....	4
2.1.	Egyszerű színegyensúly kalibrálás.....	4
2.1.1.	Áttekintés.....	4
2.1.2.	Konklúzió	5
2.2.	Lokális szíinkorrekciónem lineáris maszkolással	5
2.2.1.	Áttekintés.....	5
2.2.2.	Konklúzió	7
2.3.	Szelektív kontraszt kiegyenlítés Poisson egyenlet segítségével	7
2.3.1.	Áttekintés.....	7
2.3.2.	Poisson képszerkesztés	8
2.3.3.	Szelektív kontraszt kiegyenlítés.....	9
2.3.4.	Fourier transzformáció használata a módszerben	10
2.3.5.	Algoritmus ismertetése	10
2.3.6.	Konklúzió	11
3.	Rendszer felépítése.....	12
3.1.	A keretrendszer	12
3.1.1.	Előfeldolgozás.....	13
3.1.1.1.	Terület meghatározás.....	13
3.1.2.	Sejtek szegmentálása	14
3.1.3.	Eredmény kiértékelés	15
3.2.	A felhasználói felület.....	16
4.	A rendszer megvalósítása	17
4.1.	Implementáció során használt eszközök.....	17
4.2.	Terület meghatározás.....	18
4.3.	Feldolgozás	21

4.3.1.	Előfeldolgozás: kezdeti próbálkozások	21
4.3.2.	Új módszerek alkalmazása az előfeldolgozásban	23
4.4.	Szegmentálás.....	24
4.4.1.	Első verzió	26
4.4.2.	Második verzió.....	27
5.	Eredmények ismertetése	30
5.1.	Eredmények	30
5.1.1.	Hatékonyság.....	31
5.1.1.1.	Előfeldolgozás	31
5.1.1.2.	Szegmentálás	32
5.1.2.	Futási idő	34
5.2.	Értékelés	35
6.	Továbbfejlesztési lehetőségek.....	37
7.	Összefoglalás	38
8.	Irodalomjegyzék.....	40
9.	Melléklet	43

1. Bevezetés

A tudomány előrehaladtával újabb és újabb lehetőségek állnak rendelkezésünkre saját testünk működésének megértésére és mélyrehatóbb vizsgálatára. Közel százéves múlttal rendelkező területnek tekinthető az agy elektrofiziológiai vizsgálata. Ez a módszer képes nagy téri és időbeli felbontással információt szolgáltatni az agyműködés strukturális, funkcionális és viselkedéses összefüggéseiről. Az agy megismerését szolgáló alap kutatás mellett az orvosi tudományokban például mély agyi stimulációt, epilepsziás rohamok diagnosztizálását, előrejelzését illetve elnyomását, vagy akár motorikus funkciók visszaállítását is el tudják végezni elektrofiziológiai módszerekkel. Hasonló kutatási terület, az agy és számítógép közötti kapcsolat megteremtésével foglalkozó tudomány, ahol a cél a gépek irányítása agyi elektromos jelek segítségével. Az elektrofiziológiai elvezetés és stimulálás alapja az elektróda, vagyis az az eszköz, amellyel az agyi jeleket el tudjuk vezetni, illetve amely segítségével a stimuláláshoz szükséges elektromos jeleket közvetíteni tudjuk az idegrendszer felé.

A dolgozatomban csak regisztrálásra alkalmas elektródákkal foglalkozom, így a továbbiakban a stimuláló elektródákról nem fog szó esni. A felvevő elektródákat többféle csoportosításban lehet bemutatni. Egyrészt megkülönböztetünk nem invazív elektródákat, mint például a konvencionális EEG (elektorenkefalográfia) elektródákat, amelyekkel az élő szervezetből sérülés nélkül lehet agyi elektromos jeleket elvezetni. Ennek hátránya, hogy az elektromos jeleknek át kell haladniuk a koponyán és a bőrön, így mire eljutnak az elektródáig, jelintenzitásukban jelentős csökkenés tapasztalható. A másik az invazív elektródák csoportja, amelyeket közvetlenül az agy felszínére, illetve magába az agyszövetbe helyezünk bele. Ezek előnye, hogy a jelek az agyból közvetlenül az elektródára érkeznek, ugyanakkor hátránya, hogy használatukhoz a koponyát meg kell nyitni. Az invazív elektródák szintén nagyon sokfélék lehetnek. Humán klinikai diagnosztikában és kutatásokban nagy területeket lefedő, agyfelszínre helyezhető elektródákat használnak, míg az agykutatásban a technika fejlődésével együtt egyre inkább a kis agyterületek feltárását célzó úgynevezett mikroelektródák használata terjedt el.

A mikroelektródák felépítésük, anyaguk, és felhasználásuk szerint igen változatosak lehetnek. A több csatornás mikroelektródák leggyakrabban használt típusai a következők. A tetród négy párhuzamos kivezetésű, leggyakrabban wolfram szál as elektróda, amelynek fő felhasználási területe a regisztrált idegsejt populáción belül az egyes sejtek tüzelésének elkülönítése. A szilícium hordozóanyag alapú elektródák csoportja szintén igen változatos, vannak egy ágú, sokcsatornás lineáris elhelyezkedésű rendszerek, ilyenek azok, amelyek a

dolgozatomhoz kapcsolódnak. Vannak több ágú, azokon egy vagy több csatornával rendelkező elektróda mátrixok, mint amilyen például a „Michigan array”. Ezekkel az elektródatípusokkal a tetródnál nagyobb agyterület aktivitását lehet feltérképezni, beleértve az agyi hullámokat és a lényegesen több sejtet számláló helyi idegsejtpopuláció tüzelési mintázatát is. A szilícium alapú elektródák elektróda kontaktusának anyaga a leggyakrabban platina-irídium [1][2].

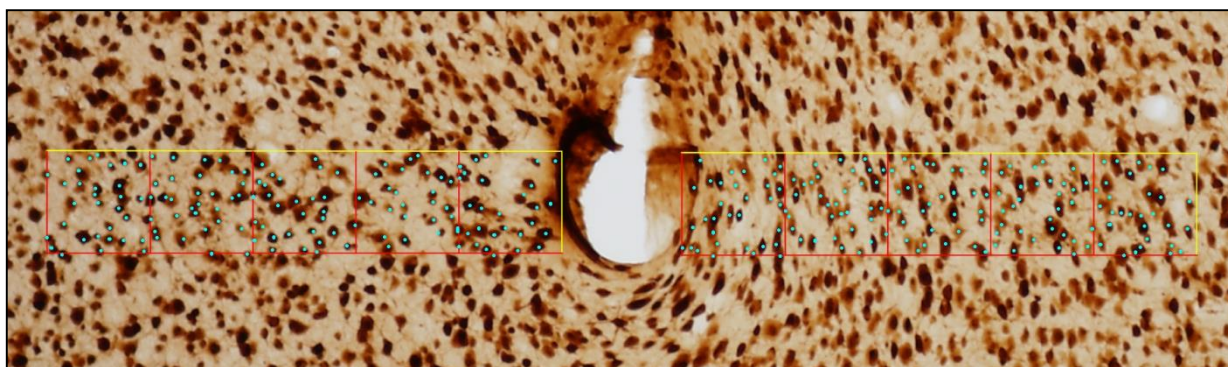
A Magyar Tudományos Akadémia, Természettudományi Kutatóközpontjában a Kognitív Idegtudományi és Pszichológiai Intézetében folyó kutatás célja, a napjainkban használatos elektródánál olyan kisebb méretű elektródák kutatása, amelyekkel nem adott idegsejt populációk, hanem egyes sejtek aktivitását lehetne egy kétdimenziós térben feltérképezni. A platina-irídium fizikai tulajdonságai komoly határokat szabnak a méret csökkentésében, így a jelen kutatásban titán-nitrit elektróda kontaktusokat alkalmaznak, amelyekkel az adott célhoz jóval közelebb lehet jutni. A szilícium és a platina-irídium agyszövetre gyakorolt hatásairól sok tanulmány készült, ugyanakkor jelenleg nem rendelkezünk kellő ismeretekkel, hogy milyen hatást gyakorol a titán-nitrit az idegsejtekre. Emiatt szükséges a megfelelő kísérletek elvégzése, hogy később megfelelő biztonsággal alkalmazható legyen esetleg embereken is.

A kísérlet lényege, hogy patkányokat előre meghatározott ideig élni hagyják az agykérgükbe beültetett elektródákkal. Mindezek után az agyukból agyi metszeteket készítenek, melyeket megfestenek idegsejtekre specifikus markerekkel, majd digitalizálnak. A digitalizált képeken, az elektróda által ejtett szöveti sérülés környékén, 5–5 darab $100\mu\text{m} \times 100\mu\text{m}$ méretű, négyzet alakú területet határoznak meg, ahol manuálisan megszámlálják a sejteket és statisztikai értékeléseket készítenek belőle. A statisztikai értékeléshez egy úgynevezett kontroll szám kerül alkalmazásra, amit mind a bal illetve a jobb oldali területre külön kerül kiszámításra. A kontrollszám meghatározása a következőképpen zajlik. A meghatározott és öt egyenlő részre felosztott terület közül, három távolabbi területen sejtszámának átlagát veszik. Ez az érték képezi a sejtek 100%-os előfordulását a környező területen. A kontrol számhoz viszonyított két belső területen található sejtek eltérő darabszáma adja a sejtek elpusztulásának százalékos értékét. A statisztikai adatok következtetni engednek arra, hogy az idő múlásával a vizsgált területen a sejtpusztulás mértéke milyen módon változik. Ebből pontosan megállapítható, hogy a titán-nitrit fejt-e ki bármiféle károsító hatást az agyszövetre.

Célom jelen TDK dolgozattal a titán-nitrit elektródák sejtkárosító hatásának vizsgálatát támogatni automatizált rendszerrel. A sejtszámlálás képfeldolgozással történő segítségével kiváltható lenne a manuális sejtszámlálás, amely egy meglehetősen időigényes folyamat a

programmal történő feldolgozáshoz képest. A kutatók így a sejtszámlálásra fordítandó idejüket a kutatás számára fontos tevékenységekkel tölthetik el.

Célom a projekttel kapcsolatban: A futtatáskor bemeneti adatként a program 4080×3072 pixel felbontású digitalizált képeket kap feldolgozásra, a bemeneti adat idegsejtekre specifikus markerekkel megfestett digitalizált agyi metszet. Amelyen találhatóak agysejtek, véredények illetve az elektróda által ejtett sérülés, elsőként beazonosítjuk a szöveti károsodás helyét. Pontos terület meghatározása után, a fentebb említett két azonos területet elhatárolása az objektum szemközti határától kiindulva (1. ábra).



1. ábra A megszámlálandó sejtek területe

Algoritmusok, eljárások felhasználásával képek módosítása, aminek hatására az objektumok szeparálhatóságának nehézsége csökken. Másképpen fogalmazva, a zavaró objektumok eltüntetése, nem összefüggő háttér homogenizálása, emellett a későbbi feldolgozás számára fontos jellemzők megőrzése. Mindezek után szegmentáló metódussal a kijelölt területen található sejtek elszeparálásra majd megszámlálásra kerülnek. Előre meghatározott szimbólummal minden egyes sejt megjelölésre kerül, a bemenetként megjelenő kép másolatán egy előre meghatározott színnel. A program a megszámlált sejtekből statisztikai adatokat készít.

2. Irodalomkutatás

A fejezet azokat az irodalomban megismert újabb előfeldolgozó eljárásokat foglalja össze, amelyek az első „alpmódszerek” után vizsgálatra kerültek, majd pedig alkalmazásukkal jelentős hatékonyság növekedést sikerült elérni a rendszerben.

2.1. Egyszerű színegyensúly kalibrálás

2.1.1. Áttekintés

A színegyensúly algoritmusok megpróbálják javítani azokat a képeket, melyek alulexponáltak, vagy mesterséges illetve természetes fénynél készültek. A témával foglalkozó irodalom több szofisztikált algoritmust jegyez, amelyek színegyensúly kalibrálással vagy különféle szín kontraszt kiegyenlítéssel foglalkoznak. Ha a képen színes fényt, például elektromos fényt használtak, akkor az R illetve G komponens dominál, a szín egyensúly a B csatornát fogja emelni. Habár ez nem szükségszerűen javítja a képet. A "Simplest Color Balance" mindig az ilyesfajta olvasható jellemzőket emeli ki [3].

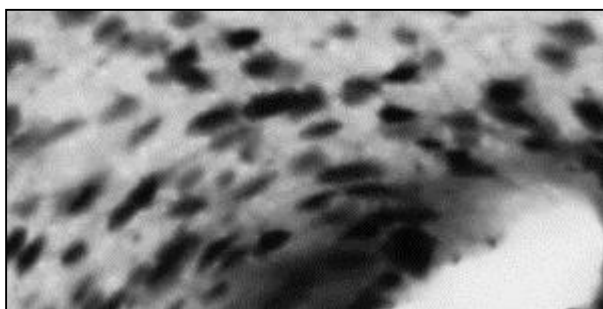
Mondhatni egy egyszerű széthúzó algoritmus, ami kiterjeszti a három csatornát, hogy alkalmazni tudjuk a maximális lehetséges területet, vagyis a $[0,255]$ tartományt. Erre a legegyszerűbb módszer, amivel elvégezhető az az affín transzformáció $(ax+b)$, ahol az a , az aktuális csatorna legnagyobb értéke, azaz 255, illetve b , ami a legkisebb, vagyis 0. Habár előfordulhat az az eset is, hogy úgynevezett aberrált pixelek találhatók a képen, ami azt jelenti, hogy ezek a pixelek már a 0 vagy a 255 intenzitás értékkel rendelkeznek az aktuális csatornán. Ezeknél az eseteknél az affín transzformáció előtt előjavítást kell végezni. Levágjuk a 0-hoz és a 255-höz legközelebbi pixelek egy kis százalékát. A művelet után szokatlan telítettség vehető észre a képen, fehér és fekete összefüggő területek alakulhatnak ki. Ezért ezt a százalékos értéket óvatosan kell beállítanunk.

Az algoritmus egy robosztusabb megközelítést tartalmaz a legkisebb és a legnagyobb értékek feltérképezésével, melyek a kimeneti tér szélsőértékeivel lesznek azonosak. Miután definiálva lettek ezek a szélsőértékek, a felhasználó által százalékosan megszabott pixelek arányát elhagyjuk ebből az intervallumból. Ez a cikk a képek N darab pixeleit csatornánként sorba rendezi, azaz hisztogram tömböt készít belőle. A felhasználó beállít két szükséges paramétert S_1 és S_2 . Ez a két érték egy százalékot reprezentál, amely a hisztogram elején és végén található intenzitás értékek meghatározott százalékát telíti. A kiszámítása a $(N \times S_1)/100$ és $\left(\frac{N \times (1-S_2)}{100}\right) - 1$ képletekkel történik. Ezzel meghatároztuk az osztóértéket.

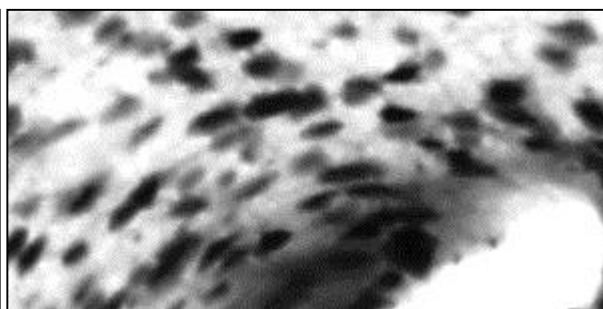
Ezek után telítjük a meghatározott intenzitás értékeket. Végezetül pedig affin transzformációt hajtunk végre, ami csatornánként a $[V_{min}; V_{max}]$ tartományból a $[0; 255]$ tartományba képezi a képet. [3].

2.1.2. Konklúzió

Az algoritmus egyszerű alapokat használó módszernek mondható. De ez nem jelenti azt, hogy nem hatékony, mivel a bemeneti kép hisztogramjának két szélét a két szélsőértékhez telíti az intenzitás értékek előre meghatározott százalék értékével.



2. ábra Eredeti kép



3. ábra Simplest Color Balance használata

Véleményem szerint ezzel a sejtek köztes területei jól általánosíthatóak, mint ahogy a tesztelésem eredménye az 2. és a 3. ábrán jól látható.

2.2. Lokális színtkorrekció nem lineáris maszkolással

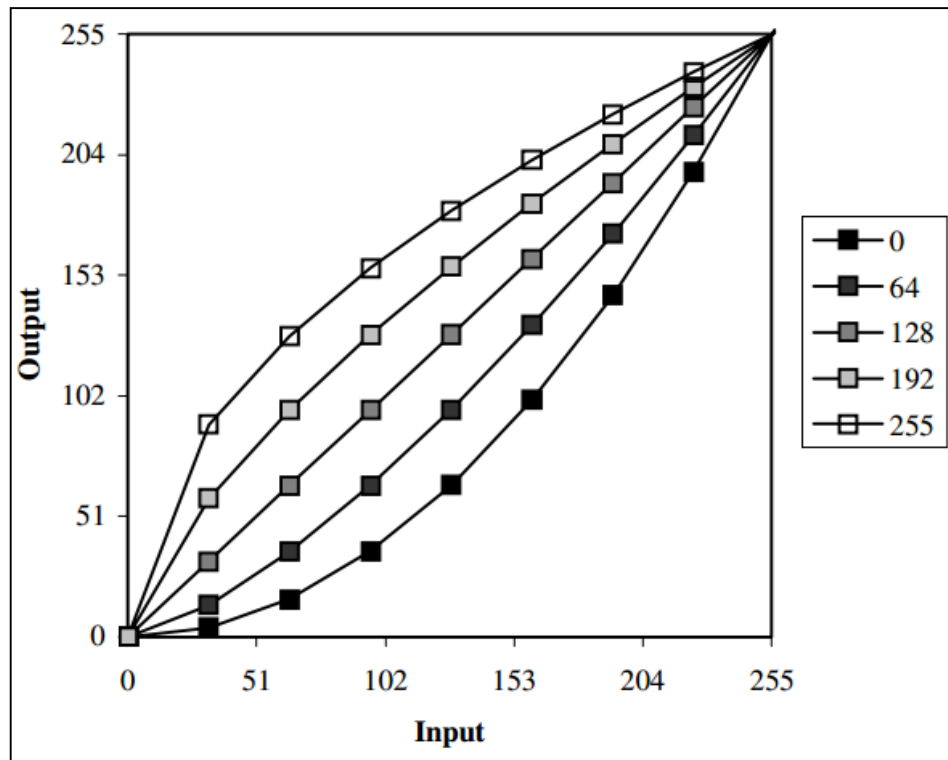
2.2.1. Áttekintés

Árnyalat reprodukció a kulcs komponense a kép minőségének, és a képek esetében globálisan használták. Későbbi vizsgálatok eredményei azt mutatták, hogy a lokális korrekciók kiemelkedőbb eredményeket mutattak a globális korrekcióval szemben. A [4] cikk a lokális szín korrekciót taglalja, aminek alapjául nem lineáris maszkot használtak. A publikáció szerint az alkalmazás gyors és nem igényel manuális beavatkozást paraméter beállításához.

A programot a következő főbb lépésekre lehet bontani. Először a bemeneti képből készít egy maszkképet. Következő lépésként pedig a maszkképet kombinálja a bemeneti képpel. A képből készített monokróm képet invertálják és végül a kép homályossá válik egy nagy sugarú filterrel. A monokróm maszk használatának az oka, hogy elkerüljük a kép színárnyalatainak a torzítását. Az invertálás célja, hogy jelezze számunkra a kép világos és sötét területeit. A képet annyira homályosítják, hogy a későbbiekben a kép jellemzői ne legyenek felismerhetőek. Ha a maszkot nem homályosítjuk a kép kontrasztja nagymértékben lesz csökkentve, de ha túlhomályosítjuk, akkor az algoritmus leeredukálódik egy egyszerű gammakorrekcióra. A feldolgozás gyorsaságának szempontjából a leghatékonyabb megoldás maszk készítésére ikon

kép használata vagy a bemeneti képből úgynevezett alképek készítése, és a maszk e részképekből való előállítás. Az összetett műveleti kód magába foglal egy egyszerű erő függvényt (1), ahol a kitevőt a maszk értékéből származtatják. Ha a maszk értéke nagyobb, mint 128 akkor a kitevő értékének kisebbnek kell lennie, mint egy, és ha a maszk értéke kisebb, mint 128 akkor a kitevő értékének nagyobbnak kell lennie, mint egy. Ha pedig a maszk értéke 128, akkor a kitevő értéke egy lesz és a bemeneti adat nem változik.

$$Output = 255 \times \left(\frac{Input}{255}\right)^{2^{\left(\frac{128-Mask}{128}\right)}} \quad (1)$$



4. ábra Kimeneti és bemeneti értékek közötti kapcsolat [4]

Erre az árnyalat reprodukciós felületre melyet az (1) határoz meg, úgy gondolhatunk, mint a tónus reprodukciós görbe kétdimenziós verziójára. Ahol a görbületek reprezentálják számunkra a maszk érték függvényeként a 8 bites RGB bemeneti értékek feltérképezését a kimeneti értékekhez. A 4. ábrán jól látható, hogy a tónus görbe a bemeneti értékek és a kimeneti értékek közötti kapcsolatot reprezentálja, jelen esetben, két dimenzióban. Jegyezzük meg azt, hogy a maszk levezetésének a folyamata hasonló lehet a megszokott módon használt szűrő maszknál, de a nem lineáris kombinált működés jelentősen különbözik a szűrő maszknál használatos additív működéstől, ami a lokális a lokális színcorrekció eredményeinél is tükröződik.

2.2.2. Konklúzió

Egy alulexponált kép esetén, ha globális korrekciót használnánk, a világos területeken a részletességet csökkentené. Ezzel szemben a publikáció által tárgyalt eljárás használatakor a kimeneti képen nem tapasztalható határok gyűrűződése és bármiféle érték túllendülés. Ráadásul, ha a képen fehér határ található, a feldolgozás nem végez rajta torzulást, ami két tényező eredménye. A nem lineáris maszkolás segítségével a fehér és a fekete képpont értékek nem korrelálnak egymással, vagyis a határoknál alacsonyabb a kiegyenlítés értéke, mivel itt a gyűrűződés gyakrabban előfordul. Második tényező a maszk aluláteresztő természete, ami azt jelenti, hogy a tónus korrekció lassan megjelenik a kimeneti képen. Ezek a fokozatos átmenetek eredményeznek lokális tónus korrekciókat, amik nem egyértelműek komplex régiók és képek esetén. De másrészt egyszerű képeknél a maszk is egyszerű és a szélsőértéknél leredukálódik egy egyszerű globális tónus korrekcióvá.

2.3. Szelektív kontraszt kiegyenlítés Poisson egyenlet segítségével

2.3.1. Áttekintés

A lokális kontraszt vagy szín kiegyenlítéssel foglalkozó műveletek célja, hogy a képen egy vagy több lokális tartalommal bíró képrészletet kijelöljön. Ezeket a részeket manipulálva egy módosított képet hozunk létre, amely a számunkra kézenfekvőbb paramétereket vagy jobban elkülöníthető stílusjegyeket hordoz. Képszerkesztő programok esetében a műveletek a bemeneti I képen, manuálisan kijelölt R téglalap alakú területen manuálisan kijelölnek egy Ω régiót. A feladat, hogy kitöltsük ezt a régiót, azon információkból melyet az R/Ω terület, vagy egy másik kép, vagy az Ω által körülhatárolt kép szelektív kontraszt változásából nyerünk. A legtöbb korszerű algoritmus a probléma megoldásának alapjául parciális differenciál egyenletet használ. A legnagyobb kihívás a kép szerkesztésében, hogy elkerüljük azon szín vagy textúra módosításokat, amik a módosított kép részen $\partial\Omega$ körvonalakat, árnyképeket eredményeznek. Erre a problémára már született publikáció, Perez és társai [6], ami megoldásként a Poisson függvény (2) használatát ajánlották eredményesen.

$$\begin{cases} \Delta u(x) = \text{div}(V)(x) & \text{Ha } x \in \Omega \\ u(x) = g(x) & \text{Ha } x \in \partial\Omega \end{cases} \quad (2)$$

ahol

$$\Delta u(x, y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (3)$$

ami az u totális deriváltja (3) és V pedig az Ω vektortere. A megközelítés egy nagyon hatékony megoldás az említett problémára. A Poisson féle képszerkesztés, javítja a sötét képek kontrasztját. Az ismertetésre kerülő numerikus metódus lehetőséget ad az automatizált lokális kontraszt kiegyenlítésre [6]. Ez az újfajta szemlélet gyors, hatékony és nem igényel manuális beavatkozást és lehetővé teszi régiók szerkesztését tetszőleges komplex topológiával bármiféle költség nélkül. Továbbá világosabbá teszi az alacsony kontrasztú területeket anélkül, hogy kárt okozna a jó kontrasztal rendelkező területek saturációjában.

2.3.2. Poisson képszerkesztés

Rekonstruáljuk a képet az irányított vektor mezőből. Legyen R , ami egy zár részhalmaza a síknak, a kép egy területe. Legyen Ω egy zárt részhalmaza R -nek a $\partial\Omega$ határokkal. Meg kell találnunk kép azon területét, ahol a négyzetes eltérés szerint a gradiensek a legközelebbiek (4) az irányított vektormezőn, vagyis v -n ami az Ω -án értelmezett, az $\partial\Omega$ által megadott határfeltételek szerint. A kép többi részét pedig R/Ω adja meg.

$$\min_u \int_{\Omega} |\nabla u - v|^2, \text{ ahol } u|_{\partial\Omega} = f|_{\partial\Omega} \quad (4)$$

Ez egy egyedi megoldása a Poisson függvénynek, Dirichlet határfeltételekkel (5).

$$\Delta u = \operatorname{div} v, \text{ ahol } u|_{\partial\Omega} = f|_{\partial\Omega} \quad (5)$$

A Poisson képszerkesztéssel jó eredményeket érhetünk el, de időigényes a feldolgozás. A folyamat gyorsíthatóságára Morel és a többiek [7] publikáltak megoldást Fourier Transzformációval. A Poisson függvény Fourier megoldása pontos, nem iteratív és a gyakorlatban gyorsabb, mint egy teljes többhálós metódus, még ebben a téglalap alakú terület esetében is. Kiterjesztjük az irányított vektormező meghatározott területét a teljes R képterületre, és megoldjuk variációs problémát a téglalap alakú képterületen, amit könnyen megtehetünk a Fourier technikával. Az Ω felett értelmezett V irányított vektormezőt kiterjesztjük R -re a (6)-os képlet szerint.

$$V = \begin{cases} v & \text{értelmezve } \Omega \text{ felett} \\ \nabla f & \text{különbén} \end{cases} \quad (6)$$

Ahol f a kép azon része melyet R/Ω által értelmezünk. A probléma a minimalizálás, (7) egy olyan minimalizáló, amely egyedien definiált az Euler-Lagrange függvény szerint. Ahol értelmezzük

$$\min_u \int_R |\nabla u - V|^2 \quad (7)$$

a homogén Neumann határ feltétellel

$$\frac{\partial u}{\partial n} = 0 \text{ értelmezve } \partial R \text{ felett,} \quad (8)$$

ahol n merőleges a határra.

2.3.3. Szelektív kontraszt kiegyenlítés

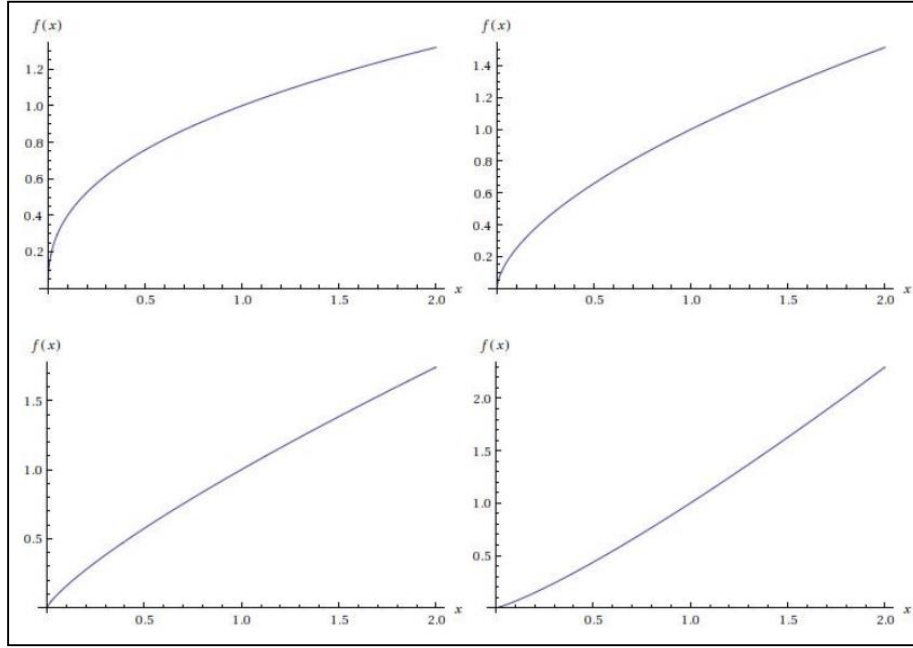
A Perez és társai [6] a kép dinamikus tartományán csak finoman módosítottak. Megoldásuk szerint a logaritmizált kép gradiens mezejét transzformálják, ezzel redukálták a magas értékű gradienseket és növelték a túl alacsony értékűeket. Az átalakított gradienst a (9) határozza meg.

$$v = \alpha^\beta |\nabla f|^{-\beta} \nabla f \quad (9)$$

Ahol $\alpha=0.2$ szerepe az f átlag gradiens normáljának a Ω felett, $\beta = 0.2$, ami az irányvektor mező gradiensének konkáv függvénye a logaritmus tartományban. Ez az átalakított gradiens használható a kép logaritmusának rekonstruálására, úgy hogy megoldja a Poisson egyenletet. A Perez és társai [6] publikáció nehezen implementálható mivel az Ω területet manuálisan jelöli ki, emiatt a komplex sötét területeket nem tudja megfelelően kezelni. A gradienst változtatjuk egy konkáv vagy konvex erő függvény segítségével. Ekkor az irányvektort (10) szerint definiáljuk.

$$V = |\nabla f|^{\alpha-1} \nabla f \quad (10)$$

A vektor tér normál vektora megegyezik a gradiens normál értékével. A konkáv hatás erősíti a kis képi részleteket. A konvex hatások pedig alkalmazhatóak a túlexponált képeknél. A függvényeket az 5. ábra szemlélteti, különböző α értékekkel.



5. ábra Erőfüggvények, $\alpha=0.4, 0.6, 0.8, 1.2$ értékekkel [6]

2.3.4. Fourier transzformáció használata a módszerben

A parciális differenciálegyenlet megoldásához vezető leggyorsabb és a legegyszerűbb mód a Fourier transzformáció használata. A cikk által bemutatott Retinex Poisson egyenlettel [8] ez jól látható. A Neumann féle határfeltételeket kizárólagosan úgy használjuk, hogy az eredeti képet kiterjesztjük szimmetrikusan az oldalain keresztül úgy, hogy a kiterjesztett kép négyszer nagyobb lesz, ezáltal szimmetrikussá és periodikussá válik. A diszkrét Fourier transzformáció lehetővé teszi, hogy kiszámoljuk a sávzáró Fourier együtthatókat.

2.3.5. Algoritmus ismertetése

A cikkben [16] tárgyalt Poisson egyenlettel szemmel láthatólag jó értékeket értek el. Főként a legegyszerűbb lehetséges függvény lineáris szorzatával, az Ω képterületek azon részén ahol a sötét régiók faktora nagyobb, mint 1. Ez az eredmény a publikáció által ismertett két bemutatott lehetőség közül az elsőhöz vezetett, amit kiegyenlített sötét opció néven említenek. Az Ω -val jelölt régiót a (11)-gyel definiáljuk.

$$\Omega = \{x \in R: f(x) \leq T\} \quad (11)$$

Ahol a $T = 50$ kezdőértéket kap, és az irányvektort a (12) szerint definiáljuk, ahol $a = 2.5$ kezdőértéket kap.

$$V = \begin{cases} a \nabla f & \text{az } \Omega - n \\ \nabla f & \text{különben} \end{cases} \quad (12)$$

A kezdőértékek nem általánosított bevett értékek, használatának okául a nyolcbites átlag minőségű képek és a sötét régiók vizuális érzékelhetőségének átlagérzékenysége szolgált. Többnyire nyolcbites színmélységű képek megjelenítése közel sötét, alacsony 50 alatti fényerőséggel történik. Ebből adódóan a T kezdőértéke 50, ami egy alap küszöbérték, melyet a sikeresen teszteltek a példaképeiken. Az α alapérték beállításának az oka a következő. Ha veszünk egy megfelelő minőségű képet, ahol a zaj normál eltérése nem haladja meg a négyet, ebben az esetben, ha a kontraszt értékeket szorozzuk 2.5-del, akkor eredményül a zaj normál eltérési értéke tíznél kisebb lesz, ez az a határ, ahol a zaj még nem zavaróan szembeűnő a felhasználó számára. A második esetben a gradiens módosítása egy α erőfüggvénnyel történik. Erre a megközelítésre globális kiegyenlítés opcióként hivatkoznak. Ekkor a gradienst nem csak a sötét területeken módosítjuk, hanem az egész képre nézve. Alkalmazva a konkáv erőfüggvényt a nagy gradienseket lecsökkenti, a kis gradienseket pedig megnöveli. Az irányított vektormező a (10) szerint képződik.

A legegyszerűbb módja annak, hogy megoldjuk a Neumann határfeltétellel rendelkező Poisson egyenletet, ha Fourier Transzformációt használunk. Ha inverz Fourier transzformációt használunk, aminek következtében a rács minden pontja u -beli érték lesz, melyeket tetszőleges konstans Fourier együtthatóval határoz meg. Az említett eljárás monokróm képek esetében használható. RGB képek esetén két lehetőségünk van feldolgozás használatára. Az első opció a következő. A kiegyenlítést minden csatornán végrehajtjuk egymástól függetlenül. A másik lehetőségünk, ha az eredeti képből készített intenzitás képen végezzük el a feldolgozást. Az első esetben a pixelek színértékei a feldolgozás során módosulnak. Második esetben pedig létrehozunk a színes kép alapján a szürkeárnyaltos képet, az algoritmust pedig a létrehozott képen végezzük el.

2.3.6. Konklúzió

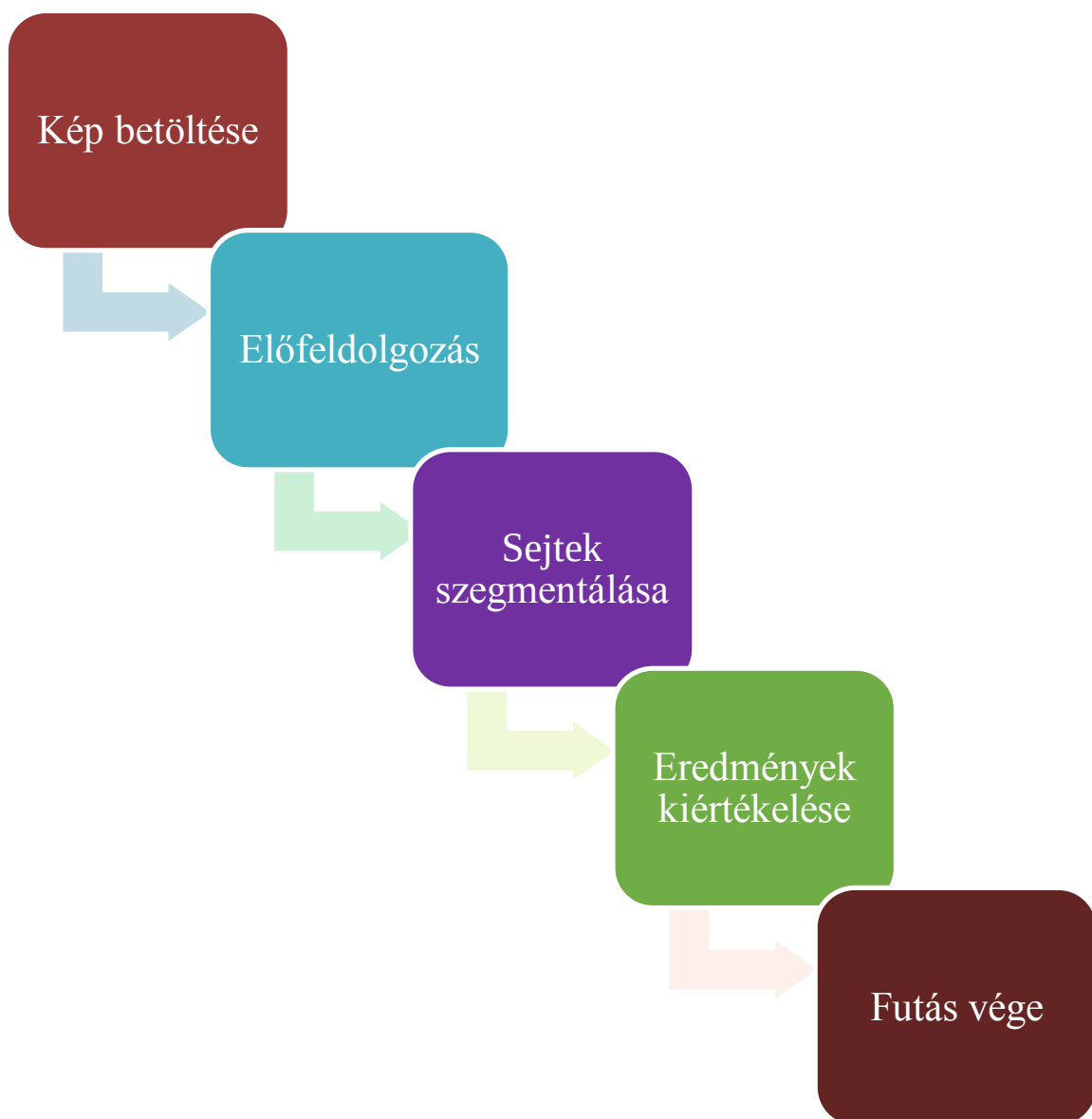
Az ismertetett publikáció egy hatékony, gyors kontrasztjavító mechanizmust mutat be. Lehetőségünk van sötét komponens kiegyenlítésre illetve globális kiegyenlítésre. A felhasználótól és a kép típusától függ, hogy mindegyik csatornát egymástól függetlenül vagy az intenzitás komponenseket dolgozzuk fel. A sötét komponensek kiegyenlítése jobb eredményt szolgáltat, mint a globális. A globális kiegyenlítés előnyei közé tartozik, a képek minőségének javítása. Az algoritmus megtartja a részletességet és ezzel együtt javítja a kép tulajdonságait. Előfeldolgozáskor feltehetőleg az említett módszer hasznosnak bizonyulhat.

3. Rendszer felépítése

A rendszer három, jól elkülöníthető részre bontható: keretrendszere, feldolgozó algoritmusokra, felhasználói felületre.

3.1. A keretrendszer

A keretrendszer hivatott magába foglalni azon algoritmusok összességét, amik a bemenetre kapott képeken hajtanak végre módosításokat sorfolytonosan, végül pedig a kimeneten megjelenő értékek eredményül szolgálnak a program felhasználói számára. A felépített keretrendszer a 6. ábrán bemutatott rendszerterv alapján épült fel.



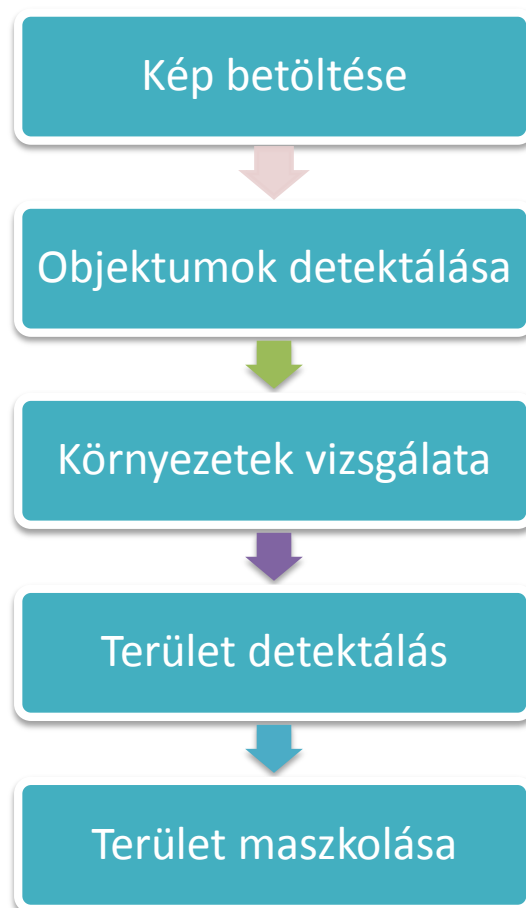
6. ábra Rendszervázlat

3.1.1. Előfeldolgozás

Az előfeldolgozás célja a bementen megjelenő képen az előre definiált feltételeknek megfelelő objektum és környezetének a meghatározása. A meghatározott területen pedig a szegmentálás szempontjából javító algoritmusok, metódusok végrehajtása.

3.1.1.1. Terület meghatározás

A terület meghatározás célja, hogy a feldolgozandó kép azon részén történjen processzálas, melyek a meghatározott feltételeknek megfelelnek. Felesleges és számításigényes a nagyfelbontású képeken az összes sejtet megszámlálni. Az elektróda által ejtett szöveti károsodás közvetlen környezete és a meghatározott terület alapján úgynevezett alképeket hozunk létre az eredeti képből. A terület meghatározás főbb lépéseit a 7. ábra szemlélteti.



7. ábra Feldolgozandó terület meghatározása

3.1.2. Sejtek szegmentálása

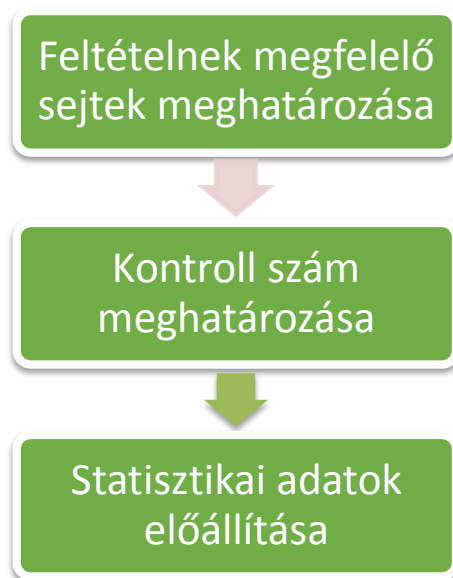
A feldolgozás folyamán a meghatározott alképeken a megfelelő képfeldolgozó algoritmusok segítségével javítjuk a kép minőségét. A javított minőségű képen pedig megszámoljuk a szegmentált sejteket. A folyamat lépéseit a 8. ábra szemlélteti.



8. ábra A feldolgozás folyamata

3.1.3. Eredmény kiértékelés

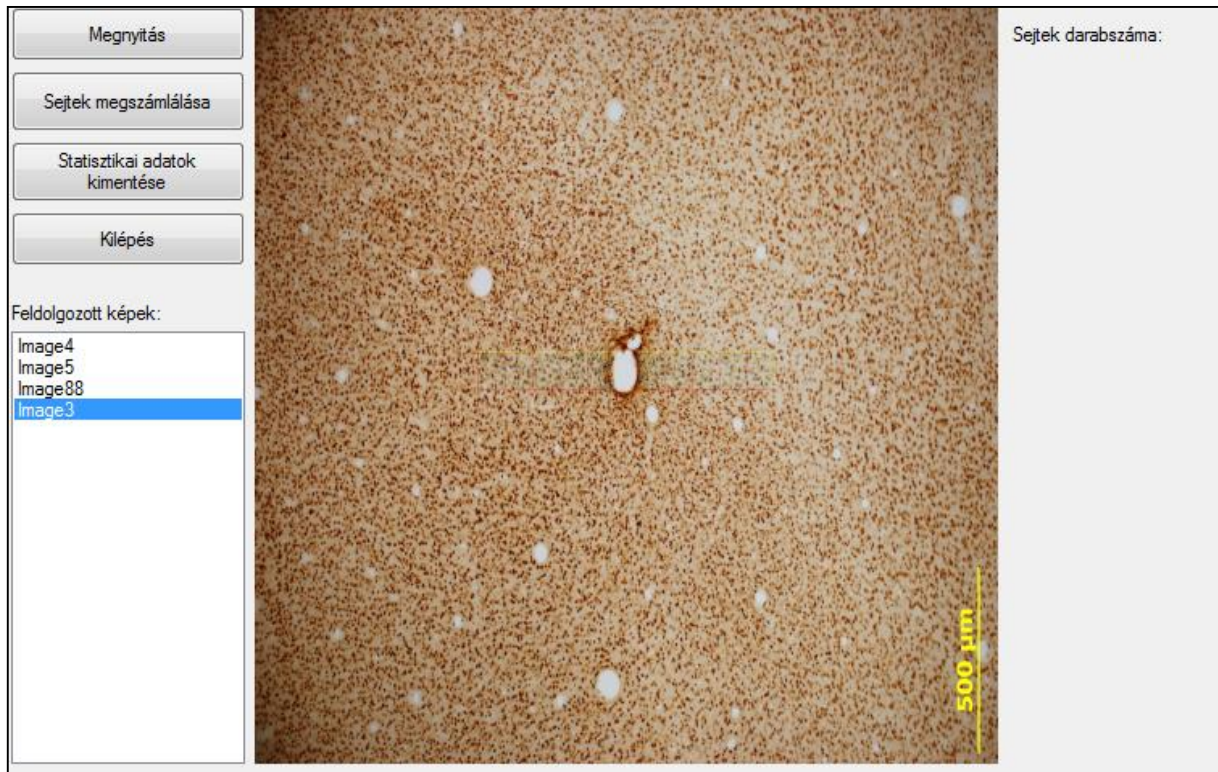
A feldolgozás folyamán meghatározott sejteket megszámlálásra kerülnek. Adott területen található sejtekből mérőszám kerül meghatározásra. A meghatározott mérőszám segítségével pedig statisztikai adat készül a felhasználó számára. A folyamatot 9. ábra szemlélteti.



9. ábra A feldolgozás folyamata

3.2. A felhasználói felület

A kezelőfelület a felhasználó számára jól látható és elkülöníthető részekből áll (10. ábra). A program bal oldalán találhatóak a programot vezérlő gombok. Alatta pedig a feldolgozáshoz betöltött képek. A felület közepén jól láthatóan a listából kiválasztott feldolgozott kép jeleníthető meg. A jobb oldalon a feldolgozás alatt behatárolt területen található sejtek darabszáma jeleníthető meg.



10. ábra Felhasználói felület

4. A rendszer megvalósítása

4.1. Implementáció során használt eszközök

A program implementálásakor mindenekelőtt törekedtem a megismert képfeldolgozással kapcsolatos alapeljárások kipróbálására, alkalmazására. Miután a témával kapcsolatos képek jellegzetes jellemzőit egyre jobban kezdtem megismerni, szembesültem azzal a ténnyel, hogy az elsajátított ismeretek inkább erős alapként szolgálnak a sokszor bonyolult és kombinált eljárások megismerése, későbbi alkalmazására. A feladat megoldásához szükségesek az Irodalomkutatás fejezetben ismertetett hatékony módszerek.

A kutatásaim folyamán több olyan modullal találkoztam, amik kimondottan specializált algoritmusokat implementálnak. Mint például képfeldolgozás, komplex matematikai műveletek, gépi tanulás. Ilyen például az OpenCV, Aforge.NET, Accord.NET, VXL, CVIPtools. Mindezek közül három ingyenesen hozzáférhető modult implementáltam a megvalósítás során. Ezek a programfüggvény könyvtárak az EmguCV [12], az Aforge.NET [11] illetve az Accord.NET. A modulok ingyenes hozzáférhetőek.

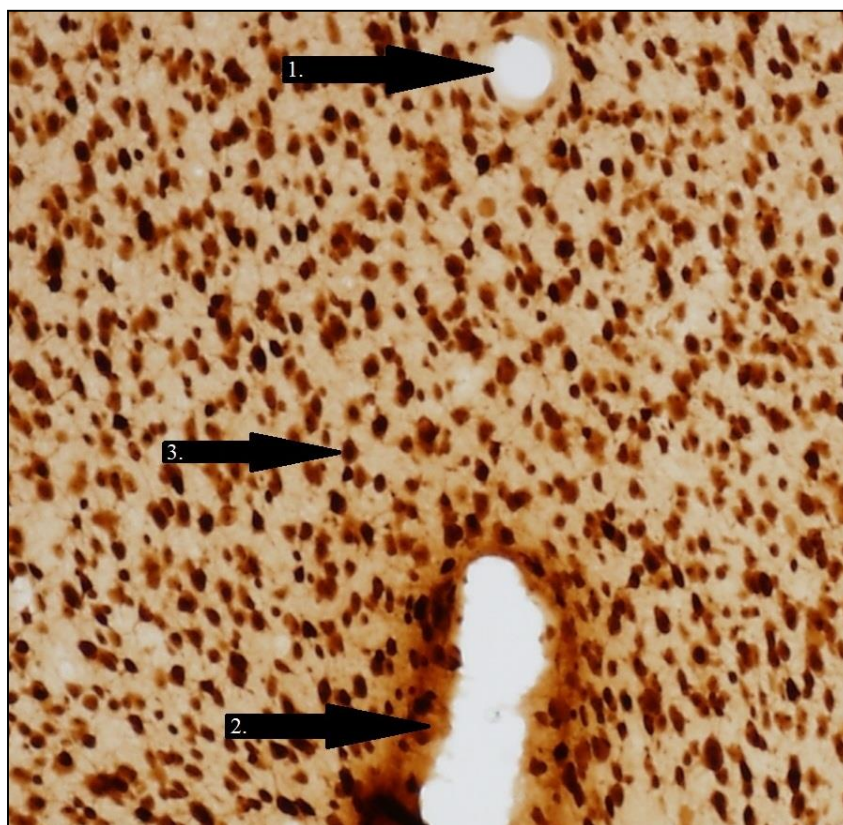
Az EmguCV az OpenCV (Open Source Computer Vision) könyvtár C# program nyelvre implementált változata [10]. Az OpenCV egy C++-ra fejlesztett programozói függvénykönyvtár, de ettől eltekintve több programnyelvre is készült már átírata. A függvényei platform függetlenek, nyílt forrású BSD licenchez kötöttek. Létrejöttének a célja a valós idejű gépi látás fejlesztése az Intel Russia kutató központ berkein belül. A közreműködésben a nagy optimalizációs tapasztalatokkal rendelkező Intel teljesítmény könyvtár csapata vett részt. Kezdetben az OpenCV céljai a következők voltak: optimalizált haladó gépi látás kutatások alapvető gépi látó infrastruktúrákra; terjeszteni a gépi látással kapcsolatos ismereteket, haladó látás-alapú programok készítése kereskedelmi célokra. Később elkészült az OpenCV2 ami olyan fő módosításokat tartalmazott, amiknek célja a könnyebb típusbiztos sablonok és függvények implementálása illetve a legjobb implementációk készítése a jelenleg kihasználható teljesítményre, kimondottan a több magos rendszerek esetén. A későbbiekben a projekt fő támogatója egy non-profit szervezet lett. Napjainkban megjelentek olyan függvénykönyvtárak, amik a műveletek gyorsabb elvégzése miatt a grafikus kártya processzorát használják.

Az Aforge.NET függvény könyvtár fő irányvonala a gépi látással, mesterséges intelligencia és a robotika, kimondottan a .NET framework-re, GPL licenccel [11]. Fő fejlesztési céljai a projektnek a gépi látás, képfeldolgozás, video feldolgozás, neurális hálózatok, genetikus algoritmusok, Fuzzy logika és gépi tanulás.

A harmadik implementálásra került modul az Accord.NET. [20], amely alapvetően az Aforge.NET könyvtár kiegészítése. Fő fejlesztési irányvonala a képfeldolgozás és a mesterséges intelligencia.

4.2. Terület meghatározás

A teljes kép azon területének meghatározásához, amely a későbbiekben kiértékelésre kerül, először szükségünk van alapszintű biológiai ismeretekre. Jellegzetes jegyek ismerete nélkül, az objektumok felismerése nem megvalósítható. Szabad szemmel a képeken három objektumot tudunk jól elkülöníteni egymástól (11. ábra). A legkisebb elemek a sötétbarna vagy fekete „pöttyök”, amelyek a neuronok sejtmagjai. A későbbiekben ezek a sejtek kerülnek megszámlálásra. Az idegsejteknél nagyobb, fehér körszerű objektumok a véredények. A nagyobb illetve több esetben legnagyobb méretű fehér, többnyire kör vagy ellipszis alakú objektum a keresendő szöveti károsodás.



11. ábra Szöveti minta elemei a véredény (fent, 1-es számmal jelölve), a neuronok (középen, 3-as számmal jelölve) és az elektróda által ejtett szöveti károsodás (lent, 2-es számmal jelölve)

Arra a következtetésre jutottam, hogy a legszembeűnőbb tulajdonsága a keresendő objektumot gyűrűként körül ölelő véraláfutás. Ez a jelleg minden esetben megtalálható az elem körül. Az objektumdetektálás ezt a jellemzőt veszi alapul. A feldolgozás első lépéseként

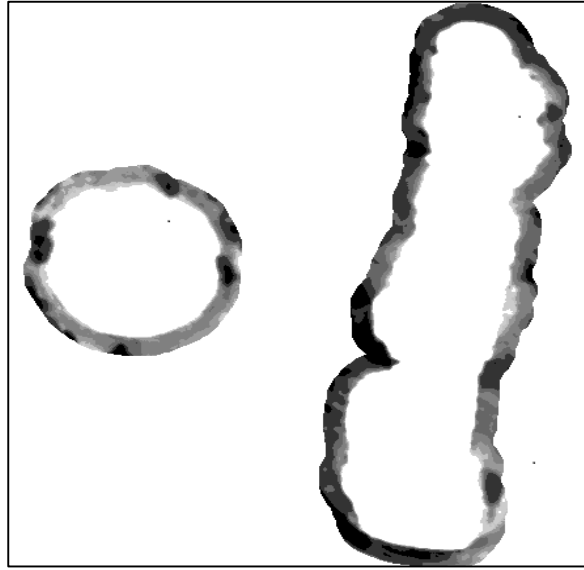
megemlítenék a feldolgozandó képek egy fontos jellemzőjét. Az agy esetében a sejtek különféle rétegződést mutatnak, ami a neuronok sűrűségét és a metszeti képek színárnyalatát befolyásolja. Ha egy metszeti kép több szintet tartalmaz, akkor a színárnyalat változó lehet egy kép esetében. Ez a probléma csillapításra kerül egy úgynevezett CLAHE [13], vagyis kontraszt limitált adaptív hisztogram kiegyenlítés segítségével, amely megköveteli az intenzitás képpé való konvertálást. A bemeneti képünket intenzitás képpé transzformáljuk. Az intenzitás kép előállításához használatos transzformáció függvénye lehet a (13)-as egyenlet, ebben az esetben egyenlő arányban osztódnak el a csatornák, a (14)-es egyenlet az úgynevezett CIE colorimetric függvény, ami fényerősség tartó tulajdonsággal bír.

$$OutputValue_{x,y} = \frac{(R_{x,y} + G_{x,y} + B_{x,y})}{3} \quad (13)$$

$$OutputValue_{x,y} = 0.2126 * R + 0.7152 * G + 0.0722 * B \quad (14)$$

A választás a (14)-es egyenletre esett, a fényesség megőrző tulajdonsága miatt. A módosított képen következő lépésként binarizációs eljárás következik. A küszöb meghatározásánál egy módosított Otsu [14] eljárás kerül felhasználásra, az eljárás definiálása a tapasztalati eredmények alapján történt.

A binarizált képen található objektumok száma kellemetlenül nagy a képen található véredényekhez képest. A probléma kiküszöbölése céljából, az objektum detektálás folyamán a gyorsabb feldolgozás érdekében csak a definiált kritériumoknak megfelelő objektumok környezete kerül megvizsgálásra. Objektum detektálás a Kapcsolódó objektumok [15] eljárás segítségével történik, előre definiált feltételek mellett. Mint kritérium, ilyen a téglalap oldalainak egymáshoz viszonyított arányára szabott intervallum, megengedett minimális és maximális méret meghatározása, amely a véredények és a szöveti károsodás méretét határolja körül. A feltételek alkalmazása az elemeket leredukálja legfeljebb 20 elemre a több száz helyett. Morfológiai erózió és dilatació segítségével minden objektumhoz előállításra kerül egy maszk.



12. ábra Objektum körüli határterület, egy véredény (bal) és egy szöveti sérülés (jobb) esetében

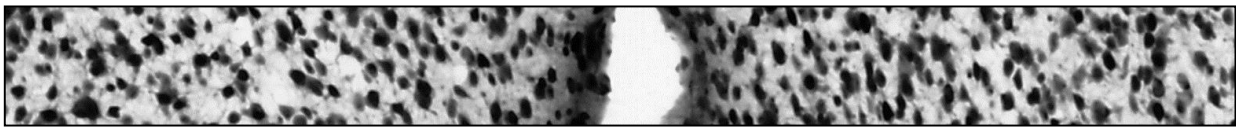
A binarizált képen az objektum pontjait jelölje X . A használt szerkesztő elem erózió esetén egy 7×7 -es Euklideszi, míg dilatació esetén 5×5 -ös Euklideszi szerkesztő elem. Először létrehozásra kerül egy maszkkép, amelyen dilatalás hajtodik végre ismétlődően, a (15) összefüggést felhasználva [9], ahol $\oplus$ jelöli a morfológiai dilataciót, a $\ominus$ jelöli a morfológiai eróziót. Egy másik maszk kép is létrehozásra kerül, melyen objektum fogyasztás történik. Az erodált maszk kép kivonásra kerül a dilatált maszk képből, létrehozva egy gyűrű maszkot (12. ábra).

$$X \oplus_k Y = X \oplus Y^{(k)} \quad (15)$$

$$X \ominus_k Y = X \ominus Y^{(k)}$$

A módszerrel meghatározott objektumokhoz normalizált hisztogram kerül előállításra. Ezzel méret függetlenül vizsgálható az intenzitásértékek eloszlása a maszkolt területeken. A normalizált hisztogram a $[0, 255]$ értékekhez rendel egy $[0, 1]$ közötti értéket, ahol 1 jelenti a maszkolt területen az adott színérték 100%-os előfordulását. A detektálás szempontjából a sötét pixelértékek viszonyított száma fontos, emiatt elemenként a $[0, 128]$ tartomány értékeit összegezzük. A vizsgált képek közül meghatározható egy maximum érték, amely egyértelműen beazonosítja a keresett elektródasebet.

Utolsó lépésként a processzálandó terület kerül meghatározása. Először a beazonosított objektum jobb, illetve bal oldalától kiindulva megjelölésre kerül 5–5 darab $100\mu\text{m} \times 100\mu\text{m}$ méretű terület, vagyis $500\mu\text{m} \times 100\mu\text{m}$ méretű terület, feltételezve a standard képméretet. Kivágva a meghatározott részt az eredeti képből, előállításra kerültek a processzálandó képek (13. ábra).



13. ábra Feldolgozásra kijelölt bal és jobb oldali terület

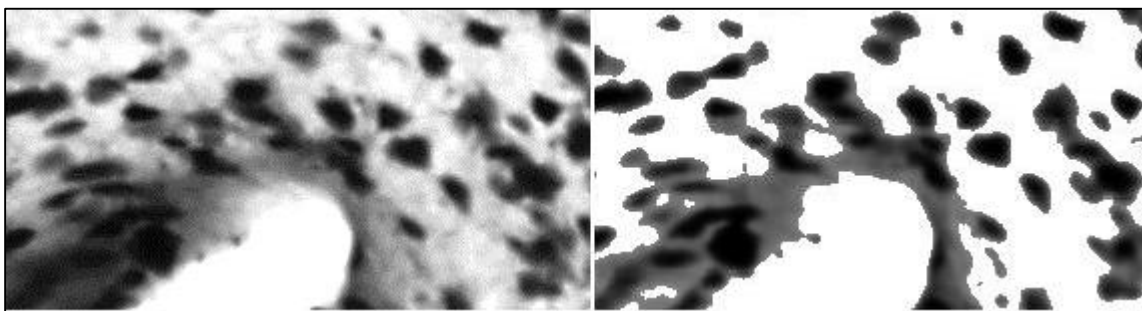
4.3. Feldolgozás

A feldolgozás célja, a detektált területen található sejtek elszeparálhatóságát zavaró tényezők enyhítése vagy teljes eltüntetése. A 11. és a 13. ábrán látható, hogy az agysejtek különböző sötétségi árnyalattal és mérettel rendelkeznek. Ez a neuronok típusbeli, méretbeli és térbeli elhelyezkedéséből adódik. A térbeli helyzetből adódóan a metszeti képen a sejt teljesen véletlenszerű részét fogja tartalmazni. Ez lehet a sejt közepe vagy széle is akár. Ebből következik, hogy az elemet jelölő szín tartomány és méret alap esetben egy nem diszjunkt halmazt képez a területen található elemekkel. A szegmentálás megfelelő hatékonyságának érdekében elengedhetetlen a feldolgozás.

4.3.1. Előfeldolgozás: kezdeti próbálkozások

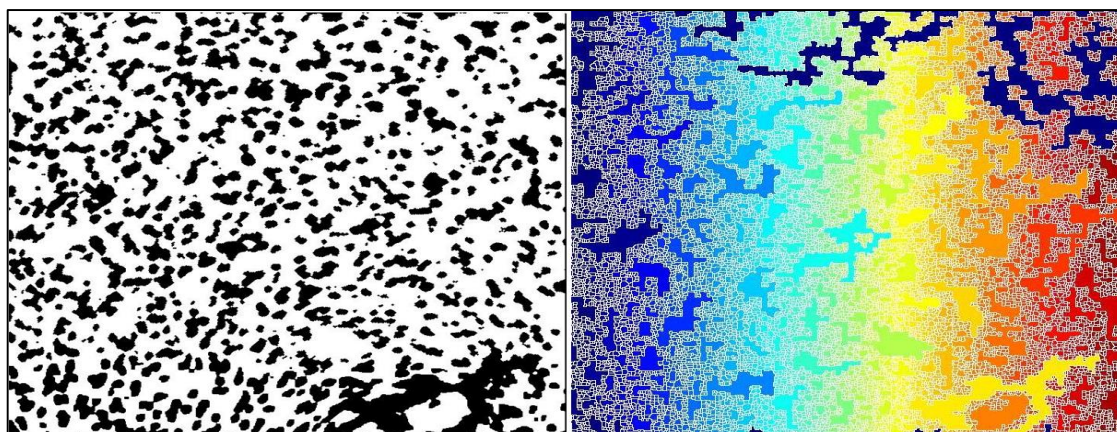
Irodalomkutatáskor megismert módszerekkel szemben az egyszerű metódusok alkalmazása sem került kizárásra. Ennek következtében elsősorban az egyszerű módszerek kombinációjának alkalmazása került előtérbe.

A bemeneten feldolgozásra megjelenő adatot intenzitás képpé alakítjuk a (14) –es egyenlet szerint. Ennek következtében alkalmazhatunk rá szűrő metódusokat. Felmerült a rétegenkénti szűrés is, de ez a nézőpont elvetésre került. Az intenzitás képen kontraszt széthúzást végzek, aminek következtében a képen található színértékek a teljes $[0, 255]$ tartományból vesznek fel értékeket. Ennek következtében a különböző objektumok színértékei kicsit jobban elkülönülnek, vagyis a vágási tartomány nagyobb lesz. A továbbiakban bármilyen elem detektálást, vagy objektum meghatározást megakadályoz a területek magas inhomogenitása. A probléma minimalizálása érdekében elmoszuk a képet. Alacsony eltérésű szomszédsági pixeleket normalizálásra kerülnek Gauss szűrő segítségével. A szűrővel elkerülhető a határvonalak begyűrűződése. A megszámlálandó neuronok intenzitás értéke a 0 értékhez, vagyis a feketéhez közelít, amíg a háttér és a kép többi objektuma a fehérhez, 255-höz közelít. Erre a jellemzőre alapozva, egy tapasztalati úton meghatározott T küszöbérték felett levágom a világos pixeleket. Eredményül olyan területet kapunk, amelyről döntő többségben sikerült eltávolítani a háttérrel (14. ábra).



14. ábra Jól látható zajok az intenzitás képen (bal), amik a háttérrel együtt leválasztásra kerültek (jobb)

A feldolgozás folyamán létrejött képen megfigyelhetők kisebb elemek, amik nem tekinthetők sejteknek a minimális méretük miatt. Ez a vágási helyzetnek köszönhető, a sejt elhanyagolható részé vágta csak le. Ezeket a sejteket szükséges eltüntetni, mivel érdemi értéket nem hordoznak számunkra. A binarizált képen, 5×5 -ös Euklideszi szerkesztő elem felhasználásával morfológiai szűrést végzünk, eredményül rendelkezésre áll egy olyan maszk, amin már nem találhatóak meg a fentebb említett elemek. A feldolgozás következő lépése a maszkolt képen a sejtek szegmentálása. A szegmentáláshoz Watershed [16] algoritmusra esett a választás. A 14. ábra jobb oldali képen látható, hogy a neuron magok sötétebbek. Abban az esetben, ha egy összefüggő területen több sejt található, felhasználhatjuk ezt a jellegzetességet, és így képesek lehetünk ezzel az algoritmussal megfelelő hatékonyságú szegmentálást elérni (15. ábra).



15. ábra Maszk kép (bal), Watershed algoritmus eredménye (jobb)

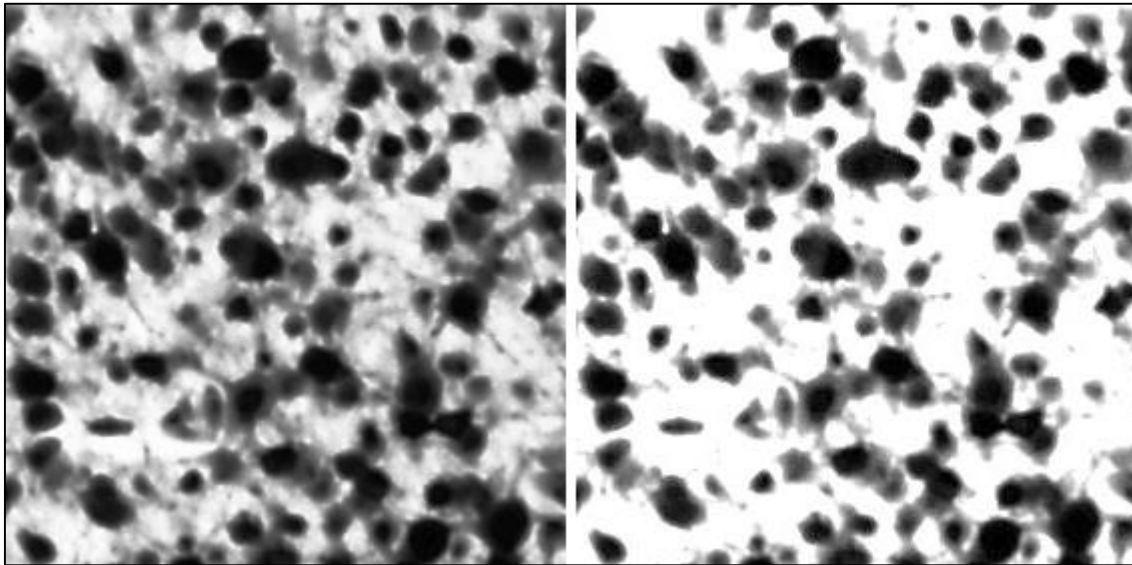
Az alkalmazott egyszerű eljárások kombinációival, hasznos tapasztalaton kívül érdemi eredményt nem sikerült elérni. A feldolgozandó bemeneti adatok színárnyalata képenként, és sok esetben egy képen belül is inhomogén, ez a háttér színekészletére is igaz. Emiatt egy tapasztalati úton beállított küszöb alkalmazásával egyszerűen levágott háttér alkalmazása nem megvalósítható. A heg körül található roncsolódás következtében elsötétülő területeken található sejtek szinte azonos intenzitás tartományba esnek, globális módszerekkel az

elszeparálhatóság nem megoldható. A szegmentálás során túlszegmentálás figyelhető meg, feltehetőleg az összefüggő objektumok minimális mérete, zaj és az eljárás globális alkalmazása miatt. Arra a következtetésre jutottam, hogy az egyszerű algoritmusok használata nem teljesen elvetendő, de mint fő alkalmazási irány nem javallott, a komplexebb területek esetén.

4.3.2. Új módszerek alkalmazása az előfeldolgozásban

A témában feldolgozott publikációk (2. fejezet) és összetett eljárások ismeretében, a következőkben említett feldolgozási eljárás került megvalósításra. Az első probléma, amivel szembetaláljuk magunkat a bemeneti képen, az árnyalati eltérések, amik régiókra bontják a bemeneti képet.

Ennek a kiküszöbölésére, a terület meghatározás kezdeti lépéseként egy úgynevezett CLAHE [13], kontraszt limitált adaptív hisztogram kiegyenlítést használunk. Az eljárás hatására az intenzitás egyensúly javul és a globális eljárások hatékonyabb működést eredményeznek. Következő lépésként kontrasztmódosító algoritmusok kerülnek alkalmazásra azzal a céllal, hogy a különböző objektumokhoz tartozó intenzitás értékek elszeparálhatósága egyszerűbb legyen. Első lépésként a Simplest Color Balance [3] eljárást használjuk. Az eljárás folyamán a 0 és a 255 intenzitás értékekhez intervallum definiálható. Az adott intervallumba tartozó intenzitás értékeket a folyamat a 0 és a 255 határ intenzitás értékhez telíti. Ezt a tulajdonságot felhasználva, a világos intenzitás értékeket a 255 értékhez telítjük, hogy a világosabb tartományok még világosabb értékeket vegyenek fel. Az objektumok és a háttér közötti különbséget kicsit sikerült megnövelni. Az intenzitáskép kontraszt egyensúlya megváltozott, az egyensúly módosításához egy másik algoritmushoz folyamodunk, mivel egy finomabb eljárásra van szükségünk a sötétebb területek javításához. A felhasznált publikáció [16] a képen lokális tartalommal bíró területeket jelöl ki, és ezen a területen végez módosító eljárásokat. A gyorsítás érdekében a Poisson függvény Fourier megoldását használja. A kép logaritmizált gradiens mezejét transzformálva a magas értékű gradienseket redukálja, amíg a túl alacsony értékűeket növeli, így a sötétebb kontrasztú objektumok homogén területei világosabb értékeket vesznek fel. Felmerülő probléma lehet, hogy a sötétebb árnyalatú objektumok határainak megőrzése nem minden esetben garantált, ezt nagymértékben kiküszöböli a cikk által taglalt Poisson egyenlettel történő használat. Az eljárás eredményeképpen, a kapott képen a sötét területek szeparálhatósága javul. Az eljárás az összetapadt sejteknél, illetve a heg körüli véraláfutásos területeknél mutat jelentősebb javulást.



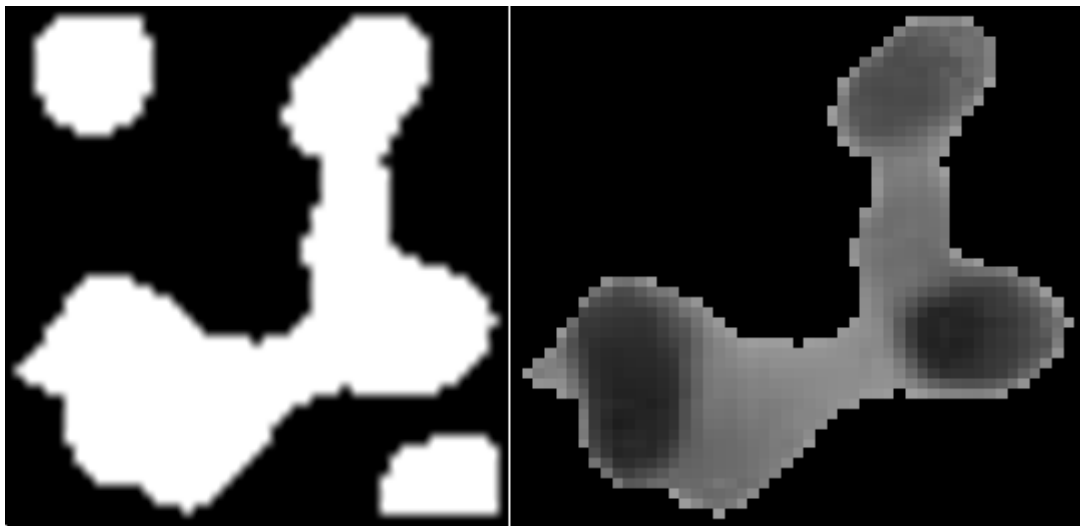
16. ábra A feldolgozásra kapott kép (bal) látható a feldolgozott képpel (jobb)

A létrejövő képen sikerült a határokat kicsit kontrasztosabbá tenni és a háttérrel kivilágosítani. A kép az eljárások után is zajosnak mondható. A sejtek közötti kapcsolatot jelentő nyúlványok a háttérrel felosztottá, pókhálószerűvé teszik és az összefüggő területek sem nevezhetők homogénnek. Egy zajcsökkentő simító szűrést hajtunk végre. A feladatra, egy széles körben használt szűrőre, a bilaterális filterre [18] esett a választás. Egy olyan nem lineáris szűrő, ami éltartó tulajdonsága révén a zajok elmosására kiválóan alkalmas, emiatt a jelen helyzetben jól alkalmazható. A szűrést finom értékhatárral használva a határok kivehetőek, de eltérésük mértéke nem elegendő mértékű. Ha a képen csak sötét területek lennének, akkor egyszerű gamma korrekcióval növelhető lenne a kontraszt a homogén területek között. Emiatt a megfelelő mértékű eltéréshez lokális szín korrekciót hajtunk végre, amit Juan Gabriel Gomila Salas és társai [17] publikáltak. A folyamat kellően gyors és hatására a lokális területek kontrasztosabbak lesznek. A lokális szín korrekcióval lezárult a szegmentálás számára a kép korrekciója. A folyamat kimenetén (16. ábra) olyan képet kapunk, amely feltehetőleg megfelelő mértékben szegmentálható és az összeragadt sejtek pedig egyszerűbben megszámlálhatóak.

4.4. Szegmentálás

A szegmentálás, mint folyamat az utolsó az eljárások közül, de fontossága vitathatatlan. Ebben a fejezetben kerül ismertetésre azon eljárások összessége, amik segítségével a képen található elemek felbontásra kerülnek, és ezáltal az egyes neuronok elkülönítésre kerülnek. A megvalósítások között, a szegmentálással foglalkozó eljárások palettája széleskörű, ezzel is jól szemléltetve a probléma sokszínűségét.

A feldolgozás első lépéseként, az elemekről maszkot készítünk. Az elkészítéshez először binarizáljuk a képet, Otsu [14] binarizációs eljárással. Az eljárás eredményeképpen, rendelkezésünkre áll a kép binarizált változata. A kapott képen morfológiai szűrés kerül végrehajtásra. A folyamat eredményéül a kis lyukak, apró zajelemek eltűnnek a képről. Következő lépésként, kapcsolódó komponensek segítségével meghatározzuk a kép objektumait. A meghatározott objektumokat kivágjuk. A kivágás négyzet alapú tulajdonsággal történik. Ebben az esetben, ha a maszk kép a közelében található szomszéd objektumok egy része beleeshet a maszk képbe, ilyen eset látható a 17. ábra bal oldali képén. Ekkor kapcsolódó komponenst alkalmazok a maszk képeken is. Ha több objektumot találna a program, akkor a legnagyobb méretű elemen kívül, az összes többi alakzat eltávolításra kerül.

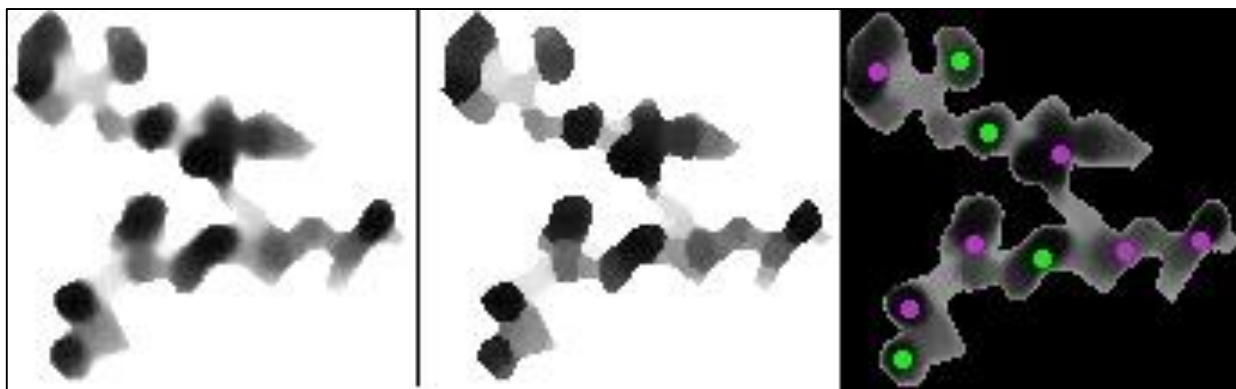


17. ábra Objektum (jobb) és a hozzá tartozó maszk kép (bal), ahol vágás után rajta maradtak a szomszéd objektumok

A feldolgozás következő lépéseként, a maszkolt objektumon található neuronok elemszámának meghatározása következik. A folyamat első lépéseként a maszkolt kép segítségével, meghatározásra kerülnek az objektum kontúrpointjai. A kontúrpointok között kitüntetjük azon pontokat, melyek az objektum szempontjából konvex görbületekre osztják az objektum határát. Ezekre a pontokra később „defektpontokként” hivatkozok. Abban az esetben, ha nem találunk ilyen defektpontokat, akkor az alakzatot egy konvex görbe írja le. Ebből következtethető, hogy az alakzat egy elemet határol körbe. Ebben az esetben a legegyszerűbb esetről beszélünk és a szegmentálás az adott kép esetén befejeződik. Ellenkező esetben a defektpontok pusztán számértéke nem enged arra következtetni, hogy pontosan mennyi az összetapadt elemek száma és további vizsgálat szükséges. Ezen a ponton az implementálás során két verzió került elkészítésre. A két feldolgozási eljárást a következőkben ismertetném.

4.4.1. Első verzió

Első lépésként, az intenzitás értékek segítségével, a sejtek elemszámának meghatározása történik. A 17. ábrán látható neuronok alacsonyabb intenzitás értékekkel rendelkeznek, mint a környezetük. Ezt a jellemzőt kihasználva, a feldolgozás első lépéseként egyszerű kontraszt széthúzást hajtunk végre a maszkolt területen. Ennek a következtében, az objektumok között található magasabb intenzitás tartalmú területek a fehér értékhez jóval közelebbi vagy fehér értékeket vesznek fel. Esetünkben a feldolgozandó területek mérete kicsire redukálódott. Jól láthatóan a sejt környezete és a sejtmag jobban elhatárolódik. A binarizálás pontosságának növeléséhez szükséges a minimálisan eltérő színértékek normalizálása, ha törekedve az objektum szélek megőrzésére. Erre a feladatra a Matalas és társai által publikált adaptív szűrés [21] került felhasználásra. A szűrt objektumon Otsu [14] binarizálást hajtok végre. A binarizált területen, a morfológiai szűrés segítségével eltüntetjük a feldolgozás folyamán keletkező zajokat. Megkeressük a képen az objektumokat jelölő maszkelemeket. Ezen maszkelemeket felülvizsgálásra kerülnek. Ha nem rendelkeznek defektponttal, akkor egy objektumot jelöl a vizsgált maszk. Abban az esetben, ha rendelkezik defektponttal, akkor pillanatnyilag pontosan nem meghatározható az összeragadt elemek száma. Erre csak egy hozzávetőleges becslést adható. (18. ábra)



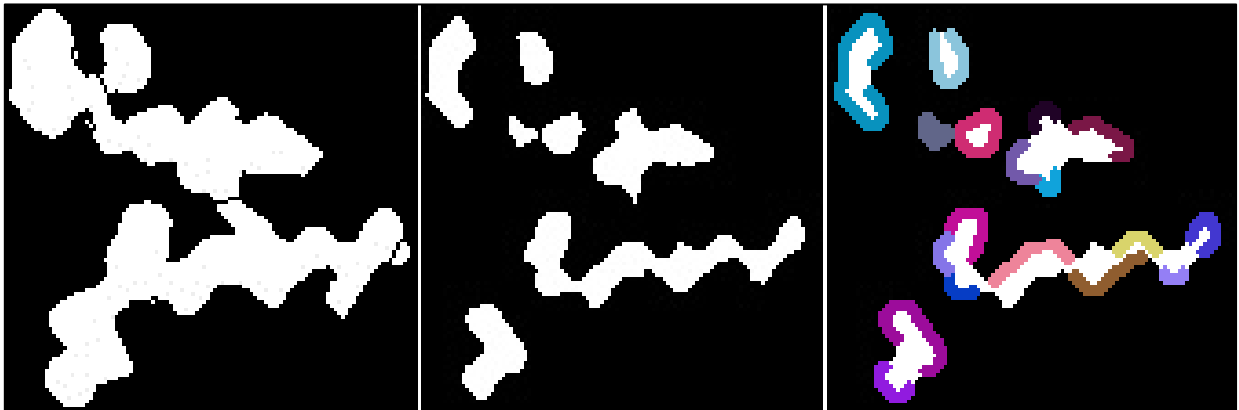
18. ábra A kontraszt szétnyújtás (bal) után adaptív szűrést végzünk (közép), a megtalált elemeket megjelöljük markerekkel (jobb), zölddel a pontosan azonosított, rózsaszínnel pedig a nem eldönthető elemeket

A meghatározott neuronok megjelöléséhez kétfajta marker kerül felhasználásra. Egy zöld marker, amivel a pontosan meghatározott sejtek kerülnek megjelölése. Egy rózsaszín marker, amivel a pontosan meg nem határozható sejteket jelöljük meg. A jelölés két szinten hajtódik végre. Az első szint a bemeneti kép replikáján vagy egy azonos méretű transzparens képen rajzoljuk fel a jelölőket. Illetve a feldolgozáskor az objektumokat tartalmazó kis képen történik megjelölésre a meghatározott sejt.

4.4.2. Második verzió

A célunk először intenzitás értékek segítségével, a sejtek pontos elválasztása illetve elemszámának meghatározása. A 17. ábrán jól látható, hogy a neuronok alacsonyabb intenzitás értékekkel rendelkeznek, mint a környezetük. Ezt a jellemzőt szeretnénk kihasználni.

Első lépésként a kontraszt széthúzás hatékonyságának növelése volt a cél. Ennek érdekében a maszk és a kontúr pontok segítségével az alakzatot feldaraboljuk. Felhasználva a defektpontok közötti legkisebb távolságot. Itt egy tapasztalati úton előre meghatározott minimális távolságot definiálok. Ebben a folyamatban feltételezem, hogy például két közeli sejt közötti terület színértéke nem sokban tér el a magától a sejt színértékétől. Ebből kifolyólag a binarizálás folyamán egy objektumként történt a sejtek detektálása. Ha sikerül az objektumok szétválasztása, akkor a szétválasztott objektumok számával megegyező szegmentálandó képet és maszkot kapok. (19. ábra)



19. ábra Minimális terület vágást hajtottunk végre (bal), majd előfeldolgozást (közép) és konvex görbe meghatározást (jobb) végzünk

Következő lépésként az intenzitás értékek segítségével, a sejtek elemszámának meghatározása történik. A 17. ábrán látható neuronok alacsonyabb intenzitás értékekkel rendelkeznek, mint a környezetük. Ezt a jellemzőt kihasználva, a feldolgozás első lépéseként egyszerű kontrasztszéthúzást hajtottunk végre a maszkolt területen. Ennek a következtében, az objektumok között található magasabb intenzitás tartalmú területek a fehér értékhez jóval közelebbi vagy fehér értékeket vesznek fel. Esetünkben a feldolgozandó területek mérete kicsire redukálódott. Adaptív [21] szűrés segítségével csökkentjük az intenzitás értékek halmazát. A szűrt objektumon Otsu [14] binarizálást hajtottunk végre. Ha több elemet találunk a maszkon, a feldolgozás következő fázisba lép. Ellenkező esetben először egy módosított Otsu binarizálást végzünk az adaptívan szűrt képen. Ekkor csak a maszkolt területen található értékek alapján határozzuk meg a küszöb értéket, aminek a 0.7 szeresét vesszük a tényleges küszöbnek.

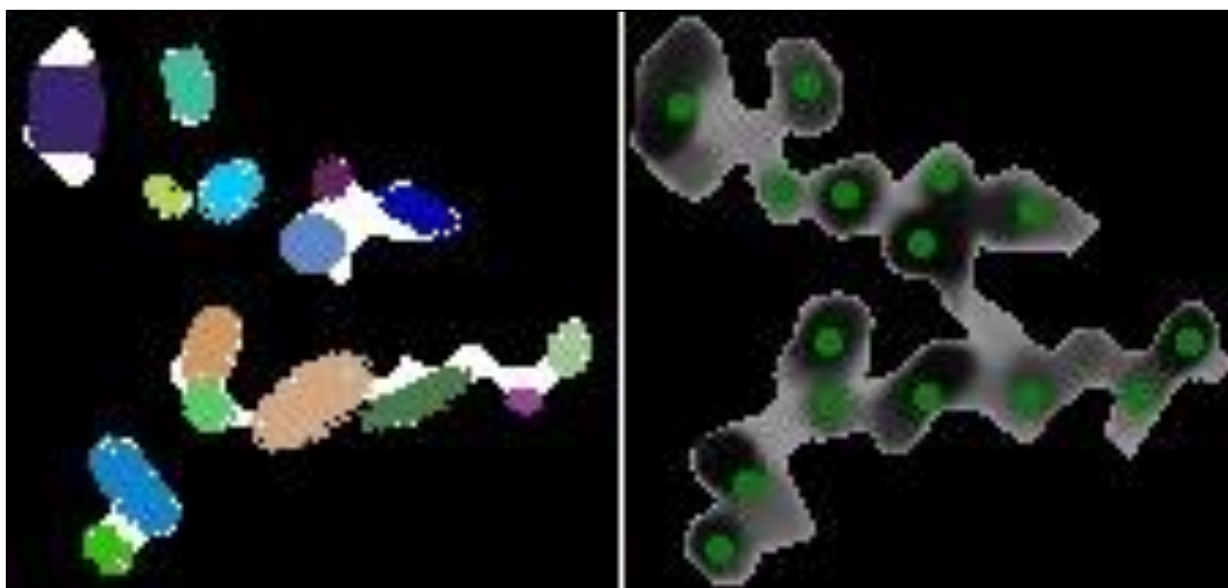
Következő lépésként meghatározzuk a módosított maszkon az objektumok elemszámát, meghatározott minimális objektum kritérium szerint. Ha több mint egy elemet detektáltunk, akkor a megtalált elemeket külön-külön tartalmazó új maszkképet és feldolgozandó képet hozunk létre és a későbbiekben ezeket a képeket processzáljuk. (19. ábra)



20. ábra a, konvex görbék, görbe vizsgálat nélkül b, hibásan meghatározott ellipszisek c, konvex görbék, görbevizsgálattal d, helyesen meghatározott ellipszisek

Következő lépésként a meghatározott maszkhoz defektpontokat keresünk és ellenőrizzük a görbület konvexitását. Ha az objektum határpontjai konvex görbét alkotnak, akkor az adott objektumot megjelöljük egy adott markerrel és befejezzük a feldolgozást. (19. ábra)

Ellenkező esetben folytatjuk a szegmentálást. A maszk által meghatározott görbületi pontokat feldaraboljuk a defektpontok mentén ponthalmazokká. Feltételezzük, hogy minden halmaz egy, konvex görbületet alkotó pontokat tartalmaz. A görbuleteket páronként megvizsgáljuk, meghatározunk görbénként matematikai tulajdonságokat. Egyező tulajdonságú és kritériumú görbéket összevonunk, ellenkező esetben fals eredményt kapunk. (20. ábra) Minden görbületre a hozzá legmegfelelőbb ellipszist illesszük, a lépéshez a Fitzgibbon és társai által publikált [19] eljárás került implementálásra. A meghatározott ellipsziseket megvizsgáljuk. Ellenőrizzük, hogy történt-e teljes vagy részleges átfedés. Ebben az esetben az egyik ellipszist eldobjuk. Az eljárás végén sikerül meghatároznunk a vizsgált képen található objektumok számát. (19. ábra, 20. ábra, 21. ábra)



21. ábra Az objektumon található görbületek alapján ellipsziseket határozok meg (bal), a meghatározott ellipszisek alapján detektálom az objektumokat (jobb)

A meghatározott neuronok megjelöléséhez egy marker kerül felhasználásra. Egy zöld marker, amivel a pontosan meghatározott sejtek kerülnek megjelölése (21. ábra). A jelöléskor két lehetőség áll rendelkezésünkre. Az első esetben a bemeneti kép replikájára rajzoljuk fel a markereket. Második lehetőségként létrehozásra kerül egy, a bemeneti kép méretével megegyező transzparens kép. A létrehozott replika alfa csatornával rendelkezik, ami az áttetszhetőségért felel. Ekkor a képen csak a markerek fognak látszódni és a későbbiekben az eredménykép külön réteggént hozzáadható az eredeti képhez. Ezzel a megtalált sejtek pontos beazonosítása illetve ellenőrzése könnyebbé válik.

5. Eredmények ismertetése

Ebben a fejezetben szeretném bemutatni azokat az eredményeket, amiket a megvalósított algoritmusok szolgáltatnak. Ugyancsak bemutatásra kerül az algoritmusok futási ideje is, bár a fejlesztés során elsődlegesen az volt a cél, hogy hatékony megoldást találjak a feladat megoldására. A feldolgozáskor használt tesztkörnyezet a következő volt. (1. táblázat)

Hardver	Hardver típusa
Architektúra	AMD
Processzor típus	Phenom II X6 1045T
Processzor órajel	2,7 GHz
Magok száma	6
Memória Mérete	2x4GB (DDR3)

1. táblázat Teszteléskor használt tesztkörnyezet

5.1. Eredmények

A tesztek során a fejlesztett rendszert és annak különböző verzióit használtam fel annak érdekében, hogy az egyes nagyobb modulok és a teljes rendszer hatékonyságát vizsgáljam. Az eredmények meghatározása két jól elkülöníthető blokkra bontható. Először meghatározásra kerül a feldolgozandó terület, melyet a kapott eredmények kiértékelése követ.

A második blokkban, előre, manuálisan meghatározott területe található sejtek megszámlálása történik. Kimentésre került a bemeneti kép replikája, amin a szegmentálás végén markerekkel megjelölésre kerültek a detektált értékek, illetve a meghatározott számértékek alapján a hatékonyság számszerűsítése.

A felhasznált tesztképek száma: 26. A képek mindegyikéhez, szakember által markerekkel megjelölt „gold standard” kép tartozik. A képek egységesen 4080×3072 pixelformátumúak, 24 bites színmélységgel. Emellett a képek rendelkezésre állnak PSD formátumban. Minden kép esetében, a heg bal, illetve jobb oldalán található feldolgozandó terület. A hatékonyság vizsgálat folyamán, a sejszámlálás képenként történt. Futási idő mérésekor a kép bal, illetve jobb területének mérése külön-külön történt.

5.1.1. Hatékonyság

5.1.1.1. Előfeldolgozás

A 2. táblázat a tesztképek előfeldolgozásának hatékonyságát mutatja be. A táblázatból leolvasható hogy az előfeldolgozás a 26 képből 24 esetben találta meg pontosan a vizsgálandó területet. A folyamat 92.3%-os sikerességgel valósul meg. A pontatlanság a képen található abnormalitásból következik. Bizonyos képek esetében, az agy szeletelése közben előfordul az agyi ér hosszanti vágása, illetve a környezet tulajdonsága is teljesen eltérhet a megszokott képi jellemzőktől. Ilyen esetet figyelhetünk meg a 23. ábrán, míg egy standard jellegeket hordozó képet a 10. ábrán tekinthetünk meg.

Kép neve	Megtalált elektróda heg
image 003	Igen
image 005	Igen
image 044	Igen
image 045	Igen
image 046	Igen
image 047	Igen
image 050	Igen
image 051	Igen
image 052	Igen
image 053	Igen
image 077	Igen
image 087	Igen
image 099	Igen
image 117	Igen
image 118	Igen
image 119	Igen
image 120	Igen
image 128	Igen
image 186	Igen
image 188	Igen
image 195	Igen
image 197	Igen
image 203	Nem
image 204	Nem
image 205	Igen
image 212	Igen
Összesen	24

2. táblázat Előfeldolgozás hatékonyság tesztje

5.1.1.2. Szegmentálás

A szegmentálást két, implementált eljárással végeztem. A következőkben szeretném szemléltetni a kapott eredményeket, mindkét esetben.

A 3. táblázat szemlélteti az első verzió szegmentálásának eredményességét. Leolvasható a táblázatból, hogy a mintaképeken a legrosszabb hatékonyság 58,8%, míg a legjobb eredmény 82,6% lett. Az elért átlagos hatékonyság 70,2% volt.

Kép neve	Megtalált Sejtek	Mintaszám	Hatékonyság (%)
image 003	328	480	68,3
image 005	310	439	70,6
image 044	323	420	76,9
image 045	263	343	76,7
image 046	308	512	60,2
image 047	298	396	75,3
image 050	282	354	79,7
image 051	294	360	81,7
image 052	272	349	77,9
image 053	285	345	82,6
image 077	299	388	77,1
image 087	286	438	65,3
image 099	255	380	67,1
image 117	259	422	61,4
image 118	258	354	72,9
image 119	260	359	72,4
image 120	275	376	73,1
image 128	287	437	65,7
image 186	287	462	62,1
image 188	276	431	64,0
image 195	301	449	67,0
image 197	287	411	69,8
image 203	295	459	64,3
image 204	276	364	75,8
image 205	303	398	76,1
image 212	286	486	58,8
Átlag	7453	10612	70,2

3. táblázat Szegmentálás hatékonyság tesztje, első verzió esetén

A 4. táblázat szemlélteti a második verzió szegmentálásának eredményeit, vagyis a számszerűsített eredményeket és a program sikerességének százalékos értékét mutatja meg nekünk. A táblázatból leolvasható, hogy a legrosszabb hatékonyság 82,7% volt, míg a legeredményesebb esetben sikerült az összes sejtet megtalálni, vagyis 100% hatékonyságot elérni. A teszt során elért átlagos hatékonyság 90,4% volt.

Kép neve	Megtalált Sejtek	Mintaszám	Hatékonyság (%)
image 003	397	480	82,7
image 005	405	439	92,3
image 044	380	420	90,5
image 045	316	343	92,1
image 046	388	512	75,8
image 047	368	396	92,9
image 050	354	354	100,0
image 051	357	360	99,2
image 052	339	349	97,1
image 053	344	345	99,7
image 077	383	388	98,7
image 087	379	438	86,5
image 099	342	380	90,0
image 117	363	422	86,0
image 118	339	354	95,8
image 119	359	359	100,0
image 120	375	376	99,7
image 128	388	437	88,8
image 186	393	462	85,1
image 188	363	431	84,2
image 195	386	449	86,0
image 197	357	411	86,9
image 203	367	459	80,0
image 204	363	364	99,7
image 205	381	398	95,7
image 212	411	486	84,6
Átlag	9597	10612	90,4

4. táblázat Szegmentálás hatékonyság tesztje, második verzió esetén

Az előfeldolgozás hatékonysági teszten mutatott eredményei alapján elmondható, hogy sikerült egy meglepően jó hatékonysággal rendelkező megközelítést implementálni, amely legjobb esetben 100%-os míg átlagosan 90,4%-os hatékonyságot ért el.

A szegmentálás tesztelésének az eredménye a következő. A 10612 meghatározott sejtől, első verzió esetén 7453 sejtet sikerült megtalálni, ami 70,2%-os hatékonyságot jelent. A második verzió esetén ugyanennyi mintaszám közül már 9591 sejtet sikerült megtalálni, ami átlagosan 90,4%-os hatékonyságot jelent. A második verzióval sikerült 2138 sejttel többet megtalálni, vagyis 20,2%-os hatékonyságnövelést elérni. Az elért hatékonyság feltehetőleg kellő pontosságot jelent. (5. táblázat)

	Megtalált Sejtek	Mintaszám	Hatékonyság
Első verzió	7453	10612	70,2
Második Verzió	9591	10612	90,4
Növekedés	2138	0	20,2

5. táblázat Szegmentálási eljárások hatékonyságának összevetése

5.1.2. Futási idő

Ebben a pontban, a teszt során mért futási eredmények kerülnek ismertetésre. Az előfeldolgozás esetében, a program első futásakor a futási idő az átlagos futási idő többszöröse volt. Feltételezhető, hogy a változók allokálása okozza a magas futási időt. Előfeldolgozás esetén a leglassabb futási időt a legelső minta eredménye adta, ami 6,9 szekundum. A leggyorsabb futási idő 1,0 szekundum volt, átlagos futási idő pedig 1,6 másodperc (melléklet 1. táblázat).

A következőkben a feldolgozás futási ideje kerül bemutatásra. A feldolgozás mind a két verzió esetén minimális eltérést mutat. A szegmentálás második verziójánál, alapvetően a feldolgozásban nem történt érdemi változtatás. Ennek ellenére a feldolgozás és a szegmentálás esetén is történik közös függvényhívás, így a módosítások közvetve és minimálisan, de kihatottak a feldolgozás futási idejére. A feldolgozás első verziójánál a leggyorsabb idő 6,6 másodperc, míg a leglassabb idő 8,6 másodperc, az átlagos futási idő pedig 7,4 másodperc. A második verzió esetén a legrosszabb futási idő 7,3 másodperc, míg a legjobb 6,8 másodperc, átlagosan pedig 7,0 másodperc alatt futott le a feldolgozás (melléklet 2. táblázat, 3. táblázat).

A melléklet 4. és a 5. táblázata a szegmentálás futási idejét mutatja be, az implementált első és második verzió esetében. Az első esetben a legrosszabb futási idő 1,2 másodperc, amíg a legjobb futási idő 0,7 másodperc, az átlagos futási idő pedig 0,9 másodperc. A második esetben a mért legrosszabb futási idő 2 másodperc, a legjobb futási idő pedig 1,2 másodperc,

az átlagos futási idő pedig 1,5 másodperc. Jól látható hogy a másod verzió esetén a többlet művelet következtében az átlagos futási idő majdnem megkétszereződött.

A futási idő esetén a szegmentálás és a feldolgozás a következő eredménnyel zárult. A feldolgozás esetén a második verzió az elsőhöz képest átlagosan 0,4 másodperccel gyorsabban fut le. A szegmentálás esetén a második verzió az elsőhöz képest átlagosan 0,6 másodperccel lassult (6. táblázat).

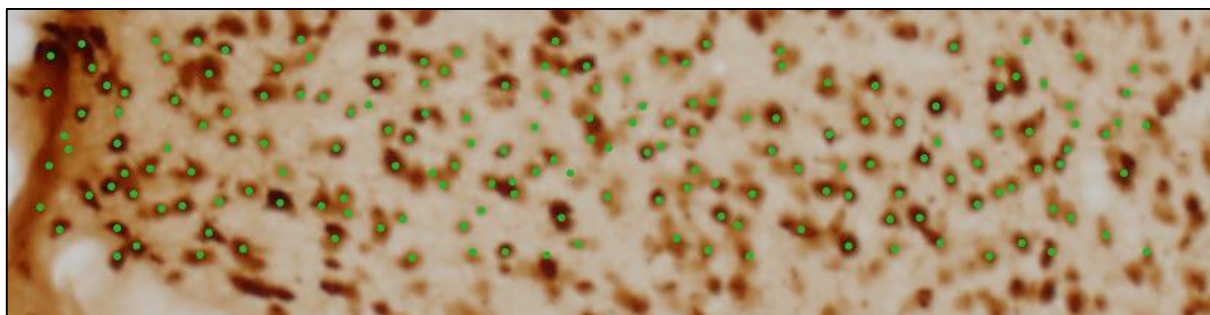
	Feldolgozás	Szegmentálás	Összesen
Első verzió	7,4	0,9	8,3
Második Verzió	7,0	1,5	8,5
Futási idő Növekedés	-0,4	0,6	0,2

6. táblázat Az első és a második verzió futási idejének összevetése

5.2. Értékelés

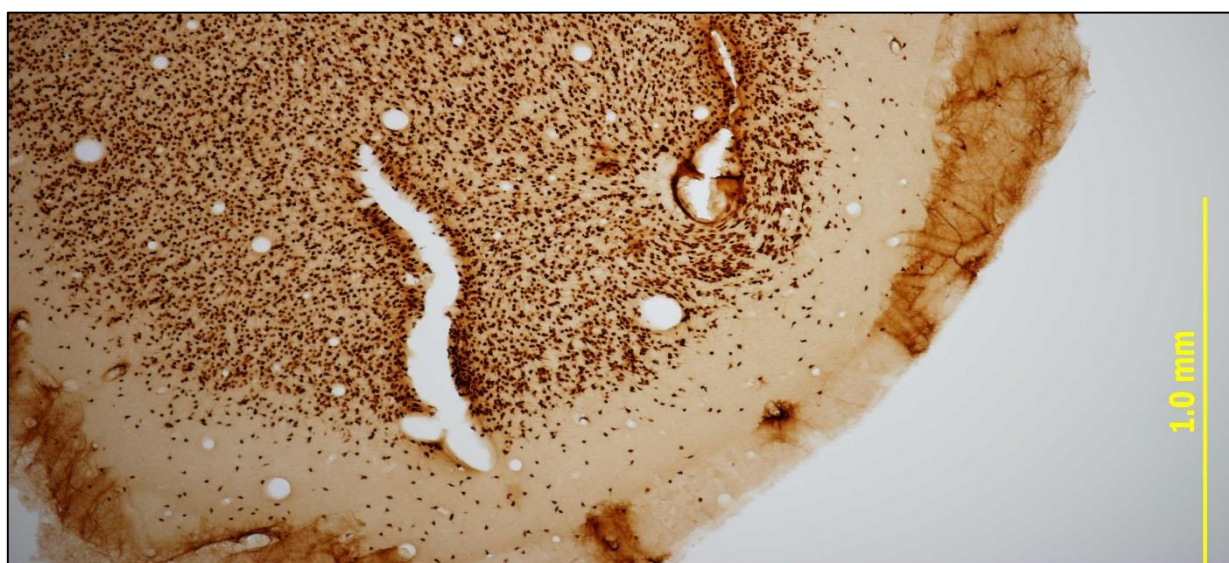
A teszt eredményekben jól látható, hogy az előfeldolgozás szembetűnően jó 92,3%-os sikerességgel zárult. A feltételezett abnormalitást tartalmazó képek esetén a felhasználónak lehetősége van a terület kézi kijelölésére. Ezzel elkerülhető a hibás előfeldolgozás.

A szegmentálás folyamán lehetőségem nyílt két implementált verzió tesztelésére. Az első verzió esetében a szegmentálás az objektumok intenzitás értékeiből fakadó eltérésre alapszik, kiegészítve konvex objektum határgöbe vizsgálattal. Ezen nézőpont alapján implementált eljárás 70% hatékonyságot volt képes elérni átlagosan 8,3 másodperc alatt. A második verzió esetén, a szín alapú szegmentálás finomításra került kiegészítve geometriai alapú vizsgálattal. Az eljárás segítségével az összeragadt sejtekre ellipszist próbáltunk illeszteni. Aminek következtében az összeragadt sejtekre egy elég jó eredményt kaptunk. A második verzióval sikerült elérni 90,4%-os hatékonyságot, átlagosan 8,5 másodperc alatt, mindamelllett hogy volt olyan eset is mikor sikerült az összes sejtet megtalálni. A második verzióval sikerült 20,2%-os hatékonyság növelést produkálni, mindamelllett a többletkód átlagosan 0,2 másodperc többlet futási időt eredményezett. A program képenként 9,9 – 10,1 másodperc alatt fut le. (22. ábra)



22. ábra Megjelölt agysejtek az eredmény képen

A program jelenleg nem használ párhuzamos feldolgozást, eltekintve a felhasznált külső modulokban (Emgu CV, Aforge.NET, Accord.NET). A program továbbfejlesztése során felhasználóbarát keretrendszer kerül megvalósításra mivel a jelenleg használt tesztkörnyezet kevésbé felhasználó barát, de kellő felhasználási ismeret birtokában használható a terület megtalálására és a sejtek megszámlálására. A modul a két meghatározott objektum feldolgozása helyett, a bevezetésben vázolt eljárás szerint dolgozza fel a képet. Aminek folytán megfelelő arányszám képződik és meghatározhatóvá válik a sejtpusztulás mértéke. A program kellő hatékonysággal működik a használathoz, de a tényleges felhasználás előtt szükségesek még tesztek illetve az eljárások finomítása. A 23. ábrán az átlagos mintaképekhez képest, egy abnormálisnak tekinthető mintakép látható. Jól megfigyelhető baloldalt a hosszanti irányban ketté vágott agyi ér. A kép több mint felén a feldolgozás szempontjából érdemi információ nem található. A heg környezetének a jobb oldalán feldolgozandó terület nem határozható meg. Ezekben az esetekben általában egy feldolgozási terület határozható meg. Előfeldolgozásuk jelenleg nem minden esetben lehetséges a szélsőséges jelegek miatt.



23. ábra Abnormális jelegeket tartalmazó mintakép

6. Továbbfejlesztési lehetőségek

A dolgozat elkészülésének időpontjában a projekt még számos továbbfejlesztési lehetőséget mutat. A jövőben a következő modulok implementálását tűztem ki célul, amelyek főként a speciális (hibás metszet) esetek lekezelését segítik.

- Felhasználóbarát keretrendszer
- Felhasználó által meghatározható minimális és maximális detektálható objektum méret
- Felhasználó által meghatározható, hogy adott képen automatikus vagy manuális feldolgozandó terület meghatározása
- Felhasználó általi visszacsatolás a megjelölt sejtek esetében, vagyis három marker használata. Egy, amivel a program jelöl, egy, amivel a felhasználó a rosszul jelölt esetet törli, és egy utolsó, amivel a felhasználó meg nem jelölt sejtet jelöl meg.
- Felhasználó által meghatározható, hogy egy mikron az adott felbontásnál mekkora pixelméretet jelent.
- Statisztikai modul valós idejű implementálása, aminek segítségével meghatározható az arányszám és a sejtpusztulás mértéke.
- Eljárások párhuzamosítása a gyorsabb feldolgozás érdekében.
- A szegmentálás mélyrehatóbb tesztelése, eljárás finomítása.

7. Összefoglalás

A projektmunka folyamán lehetőségem nyílt egy izgalmas és rengeteg lehetőséget rejtő témával foglalkozni, ami a biológia és az informatika határmezsgyéjéhez kicsit közelebb áll, kapcsolatot teremtve a két terület között. A kutatás, amibe sikerült becsatlakoznom a Magyar Tudományos Akadémia Természet Tudományi Kutatóközpontjában jelenleg is folyik. A kutatás célja egy újonnan kifejlesztett titán-nitrit elektróda élő szervezetre gyakorolt hatásának vizsgálata. A fejlesztendő alkalmazás célja a digitalizált agyi metszeteken található elektróda által okozott sérülés meghatározása, a meghatározott heg környezetében pedig a sejtek szegmentálása. A projekt célja olyan applikáció fejlesztése, amivel a fáradtságos munkával elvégezhető sejtyszámolást automatizálható, könnyítve ezzel a kutatók munkáját.

Irodalomkutatásom folyamán több publikációval is találkoztam. Első megközelítéskor az újabb, komplexebb eljárások helyett törekedtem a már megismert algoritmusok használatára. Az implementálás és a tesztek folyamán világossá vált, hogy az effajta megközelítés nem vezet hatásos eredményre. Ennek okai feltételezhetően a következők voltak. Az algoritmusok globálisan végeztek a képen módosításokat. Az eljárások, az intenzitás értékeket durván módosították, eltüntetve a finom határvonalakat. A kép fényerőssége, a neuronok eltérő sűrűsége és mérete illetve a háttér inhomogenitása miatt ez a megközelítés kudarcba fulladt.

Szükséges volt új, komplex eljárások használata specifikus problémákra szabva, amelyekkel finomabb módosítások végrehajthatása megvalósíthatóvá vált. Ezzel megőrizve a fontos objektum jellegeket. A sejt számolás elvégzését nehezítő problémák számbavétele után, friss módszerekkel próbálkoztam. Ilyen probléma a közeli sejtek határainak elmosódása vagy a véraláfutásos terület. A problémára jó eredményt nyújtó eljárás a Szelektív kontraszt kiegyenlítés Poisson egyenlet segítségével [6]. Az eljárás jellegzetes lokális területenként végzi a feldolgozást, Használatával az összetapadt sejteknél, illetve a heg körüli véraláfutásos területeknél jelentősen megnöveli a szeparálhatóság esélyét, ezzel is növelve a szín alapú szegmentálás hatékonyságát.

Az implementáció folyamán első körben a hatékony előprocesszálas eredményét felhasználva, szín alapú feldolgozás került előtérbe, amely hozzávetőlegesen 70%-os hatékonysággal képes a szegmentálásra. Itt a fő problémát az összetapadt sejtek jelentették. A probléma enyhítésére extra módszerek kerültek megvalósításra, olyan, mint a maszk objektum minimális területeinek az átvágása, vagy módosított Otsu eljárás. Az extra módszerek számottevő javulást azonban nem eredményeztek. A szükséges előrelépést a probléma egy új megközelítése hozta meg. Elvetésre került a színalapú megoldás továbbfejlesztése, helyette

csak a szegmentálás első fázisát képezte, amelyet egy második fázis követett, az ellipszis illesztés. Az alakzat detektálás geometriai alapjai pontosabb meghatározást nyújtottak, mint a csak szín alapú meghatározás esetében. Erre a problémára Fitzgibbon és társai által publikált [19] ellipszis illesztés került felhasználásra. Amely egy kellően rugalmas és pontos eljárásnak bizonyult, segítségével sikerült 90,4%-os átlagos hatékonyságot elérni úgy, hogy a legjobb eredmény 100% hatékonysággal zárult. A szín alapú szegmentálás kiegészítése geometriai alapú processzálassal meghozta a várt eredményt, átlagosan 20%-os növekedést sikerült elérni.

Sikerült egy olyan keretrendszert létrehozni, amely képes 92,3%-os hatékonysággal a vizsgálandó területet megtalálni és 90,4%-os hatékonysággal a sejteket megszámlálni. A hatékonyságnak köszönhetően véleményem szerint sikerült elérni egy elfogadható mértékű szintet, aminek köszönhetően a sejtszámlálás automatizálható.

8. Irodalomjegyzék

- [1] Mukamel, R, Fried, I. Human Intracranial Recordings and Cognitive Neuroscience. *Annual Review of Psychology*. (2011)
- [2] Maria Irene Montenegro, Irene Montenegro, M. Arlete Queirós, John L. Daschbach, *Microelectrodes: Theory and Applications*, Kluwer Academic Publisher, Neatherland, (1990).
- [3] Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Belén Petro, Catalina Sbert, *Simplest Color Balance*, Image Processing On Line, 1 (2011).
[<http://dx.doi.org/10.5201/ipol.2011.llmps-scb>, utolsó látogatás: 2014. április 14.]
- [4] Nathan Moroney, Palo Alto, *Local Color Correction using Non-Linear Masking*, 8th Color and Imaging Conference, pp. 108-111., (2000).
- [5] Marc Lebrun, Arthur Leclaire, *An Implementation and Detailed Analysis of the K-SVD Image Denoising Algorithm*, Published in Image Processing On Line, pp. 99-133., (2012).
[<http://dx.doi.org/10.5201/ipol.2012.llm-ksvd>, utolsó látogatás: 2014. április 14.]
- [6] P. Pérez, M. Gangnet and A. Blake. Poisson Image Editing, *ACM Transactions on Image Processing*, Proceedings of ACM SIGGRAPH 2003, (22. kiadás, 3. kötet) pp. 313 - 318 (2003).
[<http://doi.acm.org/10.1145/882262.882269>, utolsó látogatás: 2014. április 12.]
- [7] Jean-Michel Morel, Ana Belen Petro, Catalina Sbert, *Fourier Implementation of Poisson Image Editing*, Pattern Recognition Letters, (33. kötet, 3. kiadás), pp. 342–348, (2012. február 1.),
[<http://dx.doi.org/10.1016/j.patrec.2011.10.010>, utolsó látogatás: 2014. április 12.]
- [8] Nicolas Limare, Ana Belén Petro, Catalina Sbert, and Jean-Michel Morel, *Retinex Poisson Equation: a Model for Color Perception*, Image Processing On Line, 1 (2011. április).
[http://dx.doi.org/10.5201/ipol.2011.lmps_rpe, utolsó látogatás: 2014. április 13.]
- [9] Palágyi Kálmán, *Képfeldolgozás haladóknak*, Szeged, Typotex kiadó 2009, ISBN 978-963-279-518-8.

- [10] OpenCV, [<http://opencv.org/> utolsó látogatás: 2014. november 10.]
- [11] AForge.NET, [<http://www.aforgenet.com/> utolsó látogatás: 2014. november 4.]
- [12] Shin Shi, *Emgu CV Essentials*,
Packt Publishing Ltd, Birmingham, UK, (2013. november),
ISBN 978-1-78355-952-7.
- [13] Karel Zuiderveld, *Contrast limited adaptive histogram equalization*,
Published in Graphics gems IV,
Academic Press Professional, San Diego, USA, pp. 474-485, (1994).
- [14] Nobuyuki Otsu, *A Threshold Selection Method from Gray-Level Histograms*,
Published in Man and Cybernetics,
IEEE Transactions (9. kötet, 1. kiadás), (1979. Január),
ISSN 0018-9472, DOI 10.1109/TSMC.1979.4310076.
- [15] Milan Sonka, Vaclac Hlavac, Roger Boyle,
Image Processing, analysis, and machine vision,
Cengage Learning, Stamford, Connecticut, USA, pp. 332-335, (2008),
ISBN 978-0-495-24438-7.
- [16] Ana-Belen Petro, Catalina Sbert, *Selective Contrast Adjustment by Poisson Equation*,
Published in Image Processing on line, pp. 208-222, (2013. szeptember),
[<http://dx.doi.org/10.5201/ipol.2013.41>, utolsó látogatás: 2014.11.14.]
- [17] Juan Gabriel Gomila Salas, Jose Luis Lisani, *Local Color Correction*,
Image Processing On Line, (2011. Szeptember),
[http://dx.doi.org/10.5201/ipol.2011.gl_lcc, utolsó látogatás: 2014.10.20.]
- [18] C. Tomasi, R. Manduchi, *Bilateral Filtering for Gray and Color Images*,
Sixth International Conference on Computer Vision, pp. 839-846,(1998. január),
ISBN 81-7319-221-9, DOI 10.1109/ICCV.1998.710815,
[<https://www.cs.duke.edu/~tomasi/papers/tomasi/tomasiIccv98.pdf>]
[utolsó látogatás: 2014.11.01]

- [19] Andrew W. Fitzgibbon, Maurizio Pilu, Robert B. Fisher,
Direct least-squares fitting of ellipses,
Published in Pattern Analysis and Machine Intelligence,
IEEE Transactions (21. kötet, 5. kiadás), pp. 476-480,(1999. május),
ISSN 0162-8828, DOI 10.1109/34.765658
- [20] Accord.NET, [<http://accord-framework.net/>, utolsó látogatás: 2014. szeptember 15.]
- [21] L. Matalas, R. Benjamin, R. Kitney,
An edge detection technique using the facet model and parameterized relaxation labeling,
Published in Pattern Analysis and Machine Intelligence,
IEEE Transactions (19. kötet, 4. kiadás), pp. 328-341,(1997. április),
ISSN 0162-8828, DOI 10.1109/34.588006

9. Melléklet

A melléklet a futási idők vizsgálatához kapcsolódó táblázatokat tartalmazza.

Kép neve	Előfeldolgozási idő (szekundum)
image 003	6,9
image 005	1,3
image 044	1,3
image 045	1,3
image 046	1,3
image 047	1,3
image 050	1,3
image 051	1,2
image 052	1,3
image 053	1,4
image 077	1,4
image 087	1,4
image 099	1,0
image 117	1,0
image 118	2,0
image 119	1,4
image 120	1,4
image 128	1,8
image 186	2,5
image 188	1,3
image 195	1,2
image 197	1,5
image 203	1,2
image 204	1,3
image 205	1,3
image 212	1,2
Átlag idő	1,6

1. táblázat Előfeldolgozás futási ideje

Kép neve	Feldolgozás - 1. verzió (szekundum)	Feldolgozás - 2. verzió (szekundum)
image 003 A	7,2	7,0
image 003 B	7,2	7,0
image 005 A	6,7	7,1
image 005 B	7,2	6,9
image 044 A	6,6	7,3
image 044 B	7,0	7,0
image 045 A	7,1	7,1
image 045 B	7,3	7,2
image 046 A	7,1	7,1
image 046 B	7,2	7,0
image 047 A	7,0	6,8
image 047 B	7,0	7,1
image 050 A	7,0	7,1
image 050 B	7,3	7,0
image 051 A	7,0	7,3
image 051 B	7,0	6,9
image 052 A	6,8	6,9
image 052 B	7,4	7,1
image 053 A	6,9	7,1
image 053 B	7,5	7,2
image 077 A	7,3	6,9
image 077 B	7,3	7,1
image 087 A	7,4	7,3
image 087 B	7,4	7,1
image 099 A	7,8	6,9
image 099 B	7,0	7,1

2. táblázat Feldolgozás futási ideje az első és a második verzió függvényében

Kép neve	Feldolgozás - 1. verzió (szekundum)	Feldolgozás - 2. verzió (szekundum)
image 117 A	7,0	6,9
image 117 B	7,3	7,1
image 118 A	7,9	7,0
image 118 B	7,3	7,2
image 119 A	8,4	7,0
image 119 B	7,2	7,0
image 120 A	7,4	7,0
image 120 B	7,2	7,1
image 128 A	7,5	7,2
image 129 B	7,2	7,0
image 186 A	8,2	7,2
image 186 B	7,2	7,0
image 188 A	8,6	6,9
image 188 B	7,1	7,0
image 195 A	7,6	7,1
image 195 B	7,0	7,1
image 197 A	8,3	7,2
image 197 B	7,0	7,0
image 203 A	7,9	7,1
image 203 B	7,1	7,2
image 204 A	8,4	7,1
image 204 B	7,2	6,8
image 205 A	7,9	7,1
image 205 B	7,1	7,1
image 212 A	8,3	7,0
image 212 B	7,2	6,9
Átlag	7,4	7,0

3. táblázat Feldolgozás futási ideje az első és a második verzió függvényében. (1. táblázat folytatása)

Kép neve	Szegmentálás - 1. verzió (szekundum)	Szegmentálás - 2. verzió (szekundum)
image 003 A	1,0	1,6
image 003 B	0,9	1,7
image 005 A	1,0	1,6
image 005 B	1,0	1,7
image 044 A	0,9	1,6
image 044 B	0,9	1,6
image 045 A	0,7	1,5
image 045 B	0,9	1,6
image 046 A	1,0	1,7
image 046 B	0,8	2,0
image 047 A	1,0	1,5
image 047 B	0,9	1,4
image 050 A	0,9	1,5
image 050 B	0,8	1,3
image 051 A	0,9	1,7
image 051 B	0,7	1,2
image 052 A	0,8	1,2
image 052 B	0,9	1,6
image 053 A	1,0	1,7
image 053 B	0,7	1,3
image 077 A	1,0	1,8
image 077 B	1,0	1,5
image 087 A	1,1	1,6
image 087 B	0,9	1,6
image 099 A	0,9	1,4
image 099 B	1,0	1,7

4. táblázat A szegmentálás futási ideje az első és a második verzió függvényében

Kép neve	Szegmentálás - 1. verzió (szekundum)	Szegmentálás - 2. verzió (szekundum)
image 117 A	1,1	1,5
image 117 B	1,1	1,5
image 118 A	0,8	1,4
image 118 B	0,9	1,3
image 119 A	1,0	1,6
image 119 B	0,9	1,4
image 120 A	0,8	1,4
image 120 B	1,0	1,6
image 128 A	1,1	1,6
image 129 B	1,0	1,7
image 186 A	1,0	1,8
image 186 B	1,0	1,6
image 188 A	0,9	1,4
image 188 B	0,9	1,6
image 195 A	1,1	1,6
image 195 B	1,0	1,2
image 197 A	1,0	1,3
image 197 B	0,9	1,4
image 203 A	1,1	1,4
image 203 B	1,0	1,3
image 204 A	0,9	1,6
image 204 B	1,0	1,9
image 205 A	1,0	1,6
image 205 B	0,9	1,8
image 212 A	1,2	1,9
image 212 B	1,0	1,4
Átlag	0,9	1,5

5. táblázat A szegmentálás futási ideje az első és a második verzió függvényében, (3. táblázat folytatása)