

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertchnológia Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

TEXTÚRA ALAPÚ KÉPI ADATBÁZIS ÉPÍTÉSE NAGYFELBONTÁSÚ ORVOSI KÉPEK KLINIKAI VIZSGÁLATÁHOZ

Szerzők: Altsach Ádám

mérnök informatikus szak (BSc), IV.évf.

Bándi István Imre

mérnök informatikus szak (MSc), II.évf.

Kiss Gábor

mérnök informatikus szak (BSc), IV.évf.

Cseri Orsolya Eszter

mérnök informatikus szak (MSc), II.évf.

Konzulensek: Dr. Kozlovsky Miklós

egyetemi docens

Dr. Vámosy Zoltán

egyetemi docens

Budapest, 2012.

Köszönetnyilvánítás

Ezúton szeretnénk köszönetet mondani azoknak, akik hozzájárultak a TDK dolgozat elkészültéhez.
Külön köszönet Karóczkai Krisztiánnak és a 3DHistech Kft dolgozóinak a szakmai segítségnyújtásért.

Tartalomjegyzék

Köszönetnyilvánítás.....	1
Rövidítések	4
Bevezetés.....	5
1. A meglévő rendszer bemutatása.....	6
2. Orvosi háttér.....	7
2.1. Vastagbél részei és betegségei.....	7
2.1.1. A vastagbél szöveti felépítése	7
2.1.2. A vastagbél nyálkahártya.....	8
2.1.3. A vastagbél betegségei.....	9
2.2. Biológiai anyagok előkészítésének általános menete	9
2.2.1. Hematoxin festés.....	9
2.3. Digitalizált tárgylemez	10
2.4. A szöveti minták textúráinak sokszínűsége	11
2.5. DICOM szabvány és PACS.....	11
3. Képi adatbázisok.....	15
3.1. Az orvosi képek archiválása informatikai eszközökkel	15
4. Hasonló projektek	16
4.1. Összehasonlítási szempontok.....	16
4.2. ASH Image Bank.....	17
4.3. ImageAfter.....	17
4.4. TinEye	18
4.5. RevIMG	19
4.6. Windsurf	20
4.7. mdcm.....	21
4.8. ClearCanvas	21
4.9. DICOM is Easy.....	22
4.10. dcm4chee	22
4.11. Következtetések, értékelés	23
5. Textúra alapú módszerek	23
5.1. Textúradarabok definiálása	23
5.1.1. Turbopixels	24
5.1.2. SLIC superpixels	24
5.2. Textúravizsgálat.....	25

5.2.1.	Local Binary Patterns	25
5.2.2.	GLCM	25
5.3.	Következtetések, értékelés	26
6.	Célpontosítás	26
7.	A rendszerrel szemben támasztott követelmények	29
8.	A rendszer felépítése	30
8.1.	A rendszer moduljai	30
8.1.1.	Munkafolyamat alapú működés	31
8.1.2.	Annotáció	32
8.1.2.1.	DICOM modul	32
8.1.2.2.	PACS modul	33
8.1.2.3.	Lekérdezés	33
8.2.	Grafikus felület felépítése	34
8.2.1.	Textúra alapú algoritmusokat támogató eszközök	34
8.2.2.	Annotálást és adatbázissal való kommunikációt támogató eszközök	35
8.2.3.	Manuális korrekció	35
8.3.	Textúra alapú kereső algoritmusok	36
8.3.1.	Feature Extraction algoritmus	36
8.3.2.	LBP alapú algoritmus	36
9.	Tesztelési terv	41
9.1.	Hatékonyságvizsgálat	41
9.2.	Használhatósági vizsgálat	44
9.2.1.	A megfelelő minták kiválasztása és feltöltése	44
10.	Elért eredmények és értékelés	46
	Összegzés	49
	Irodalomjegyzék	50
	Függelék	54
F01.	Függelék: Értékrepresentációk	54
F02.	Függelék: A felhasznált DICOM tagok	55
F03.	Függelék: Hatékonyságvizsgálat szöveti eredményei	56
F04.	Függelék: Hatékonyságvizsgálat adatsorai	58
F05.	Függelék: Hatékonyságvizsgálat eredményei	60
F06.	Függelék: Használhatósági teszt során felhasznált minták	62

Rövidítések

ISO	International Organization for Standardization – Nemzetközi szabványügyi szervezet
PACS	Picture Archiving and Communication Systems – Kép archíváló és kommunikációs rendszer
ACR-NEMA	American College of Radiology and National Electrical Manufacturers Association – Amerikai radiológiai főiskola és a nemzeti elektromos gyártók egyesülete
GPU	Graphics processing unit – Grafikus feldolgozó egység
DICOM	Digital Imaging and Communications in Medicine - Digitális képkezelés és kommunikáció az orvostudományban
URI	Uniform resource identifier – Egységes erőforrás azonosító
SDK	Software development kit – Szoftver fejlesztői készlet
DLL	Dynamic-link library – Dinamikus függvénykönyvtár
GUI	Graphical user interface – Grafikus felhasználói felület
TCP/IP	Transmission Control Protocol/Internet Protocol
JPEG	Joint Photographic Experts Group – Állóképek szabványaival foglalkozó munkacsoport
VR	Value Representation – Érték reprezentáció
RIS	Radiology information system – Radiológiai információs rendszer
HIS	Hospital information system – Kórházi információs rendszer
SOA	Service oriented architecture – Szolgáltatás orientált architektúra
ROI	Region of Interest – Valamilyen szempontból fontos terület

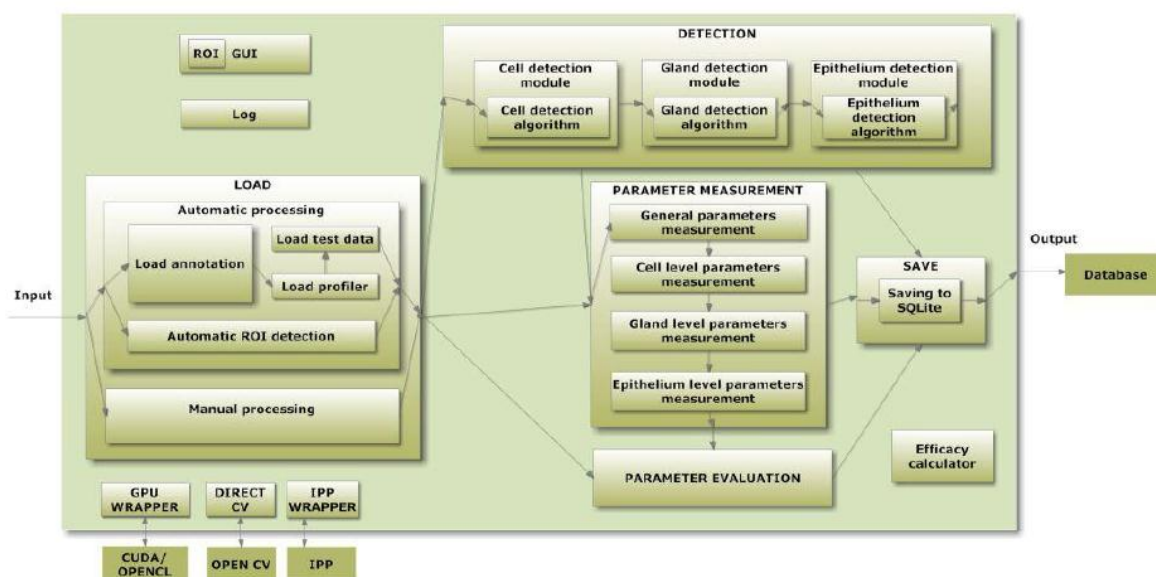
Bevezetés

A digitális fényképezés és az informatika fejlődésével megváltozott az orvosok munkafolyamata. A fejlődésnek köszönhetően lehetővé vált a mikroszkóppal vizsgált szöveti képek digitalizálása és az interneten történő továbbítása, valamint a számítógépes megjelenítés és tárolás. A folyamat változása sajnos nem hozta magával a munka egyszerűsödését. Egy orvosnak, akár a hagyományos módszerrel dolgozik, akár az új vívmányokat alkalmazza, nagyon nehéz eldöntenie egy szöveti mintáról, hogy a páciensnek mi lenne a hatékony kezelés, ha egyáltalán szüksége van rá. Manapság egy diagnózis felállítása több hetet is igénybe vesz a gyanú felmerülésétől a kezelés megkezdéséig. Ez alatt az idő alatt az esetleges betegség csak súlyosbodhat a páciens szervezetében. Ennek a TDK dolgozatnak célja a hosszas várakozási idő csökkentése és az ehhez kapcsolódó rendszer bemutatása.

A dolgozatban bemutatott fejlesztések egy meglévő rendszerben a használhatóság bizonyítása érdekében beépítésre kerültek, amelynek felépítését az első fejezetben ismertetjük. Ez egy olyan immundiagnosztikai rendszer, ami speciálisan a vastagbél-daganatok diagnosztikájára lett tervezve és megvalósítva. A jelenlegi képességei lehetővé teszik a digitális szöveti szkennerek által készített nagyméretű szöveti képek betöltését, a szövetet alkotó alapvető struktúrák (sejtmagok, mirigyek) detektálását egyszerű morfológiai paraméterek alapján, alapvető kvantitatív vizsgálatokat. Fejlesztésünk célja a program használhatóságának kiterjesztése. Pontosabb méréseket lehetővé tevő textúra alapú szegmentálási módszereket implementálunk, mivel egyéb algoritmikus eljárásokkal tapasztalataink szerint nehéz szegmentálni, amelyek az orvosnak segítségére vannak abban, hogy a különböző szöveti részeket (laza és tömött rostos kötőszöveti részek, zsírszövet, hámszövet, mirigyeket alkotó részek) azonosítsa és a méréseit csak a kívánt területen végezhesse el. Másik fő fejlesztési irányunk a DICOM/PACS alapú orvosi adatbázis-rendszer támogatása. A textúra alapú keresés erre az adatbázisra támaszkodva úgy lesz elvégezhető, hogy ehhez manuális kijelölés a felhasználó részéről nem lesz szükséges. A DICOM és PACS támogatás szabványos tárolási lehetőséget biztosít orvosi célú képeinknek, és később a szoftver képességeinek további kiterjesztését is lehetővé teszi.

1. A meglévő rendszer bemutatása

A meglévő MorphCheck rendszer egy speciálisan a vastagbél daganatok diagnosztikájára létrehozott szoftver, amelynek felépítését és moduljainak (1. ábra) működését tartalmazza ez a fejezet.



1. ábra: A MorphCheck rendszer általános felépítése [1].

A metszeti képek betöltéséért a „Load” modul felel. A vizsgálandó képeket szöveti szkennerekkel készítik, amelynek az eredménye igen nagy felbontású, akár 10.000 x 10.000 pixeles orvosi képek. Ezek gyakran elérhetik a több gigabájtos méretet is, emiatt nem lehet egészben betölteni őket a memóriába.

A betöltést követően a rendszer alapszinten megvizsgálja, hogy kiértékelhető-e a betöltött minta (formátum megfelelő-e, háttértől eltérő rész van-e rajta). Ha lehetséges, akkor detektálnia kell a sejtmagokat, a mirigyeket és a hámréteget. Lehetőség van manuális feldolgozásra, ezen opció választásánál a felhasználó által kiválasztott területen történhet a keresési és detektálási folyamat, itt az alkalmazott algoritmus paramétereinek a kiválasztására is van lehetőség. A másik lehetőség az automatikus feldolgozás, a program képes egy könyvtárban lévő összes mintát analízálni. Ilyenkor előre meghatározott annotáció által definiált területen történik a feldolgozás, ha nincs annotáció, akkor az Automatikus ROI detektáló meghatározza azt a részt, amin dolgozni érdemes. A „Load profiler” modul felel azért, hogy az automatikus feldolgozás az előre beállított sablonok alapján működjön.

A szöveti komponensek meghatározására és a morfológiai tulajdonságok meghatározására a „Detection” és a „Parameter Measurement” modulok alkalmazhatók. A meghatározott adatokból egy döntést kell hozni a páciens betegségére vonatkozólag, ezt a „Parameter Evaluation” modul feladata. A kiszámolt paramétereket és adatokat pedig helyi MySQL adatbázisban tárolja el a rendszer a későbbi kiértékelhetőség miatt. A rendszerbe bele került olyan modul is, ami a hatékonyság vizsgálatot segíti elő. Ezek a „Load test data” és az „Efficacy Calculator” modulok. A rendszer C# nyelven íródott, számos függvénykönyvtárt felhasználtak. A rendszer gyorsítása érdekében GPU-kon dolgoznak fel és futtatnak bizonyos nagy számítási igényű algoritmusokat [1].

2. Orvosi háttér

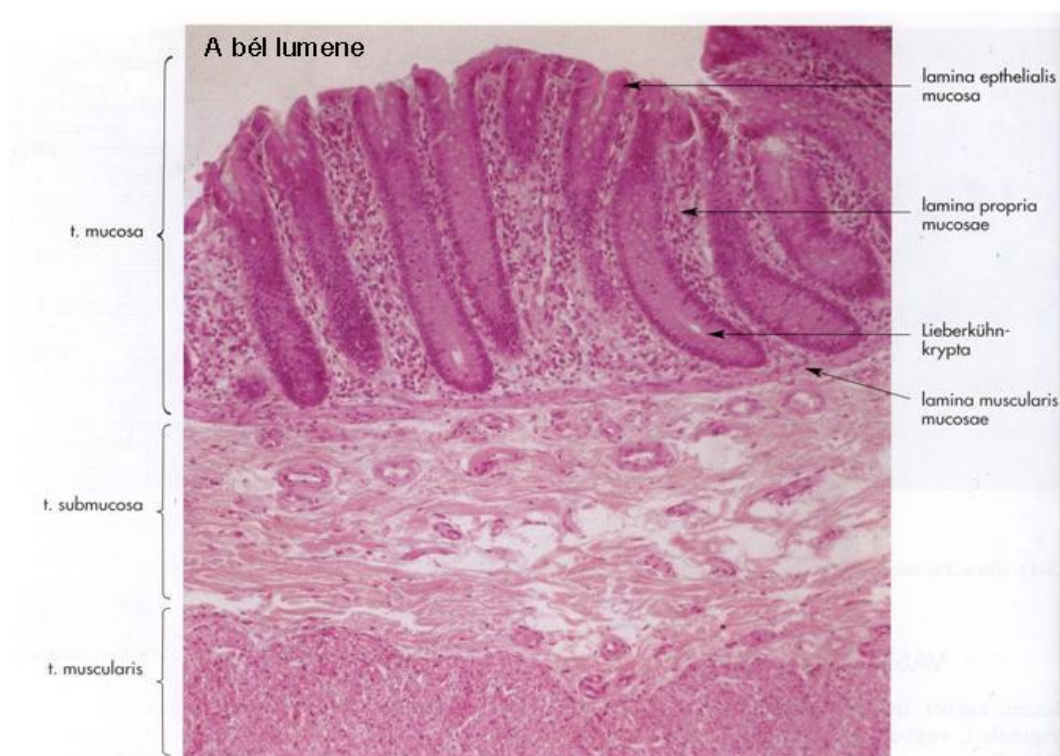
A dolgozat további részeinek megértéséhez számos témakört be kell mutatni. Az orvosi háttér rész ezeket a témákat öleli fel:

- Vastagbél részei és betegségei
- Biológiai anyagok előkészítésnek általános menete
- Szöveti minták textúráinak sokszínűsége
- DICOM szabvány és PACS

2.1. Vastagbél részei és betegségei

2.1.1. A vastagbél szöveti felépítése

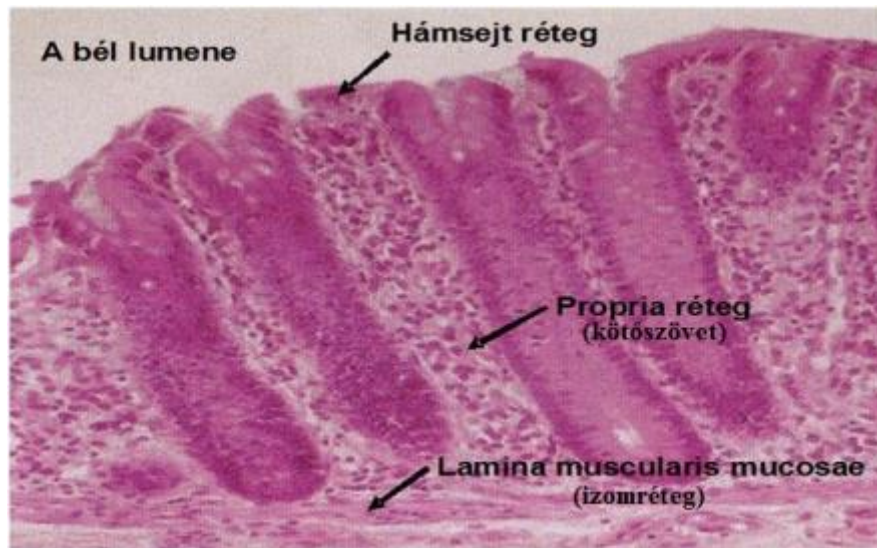
A vastagbél az emésztőrendszer utolsó szakasza. A bélrendszer feladata a táplálék emésztése, felvétele, felszívása és az emészthetetlen salakanyag külvilágra juttatása. A vastagbélrák az egyik leggyakoribb daganatfajta, világviszonylatban a második leggyakoribb megbetegedés, Magyarországon évente körülbelül 8000 új megbetegedést diagnosztizálnak. Az 2. ábra bemutatja a vastagbél felépítését [1].



2. ábra: Vastagbél szöveti szerkezete (Valcz, 2008).

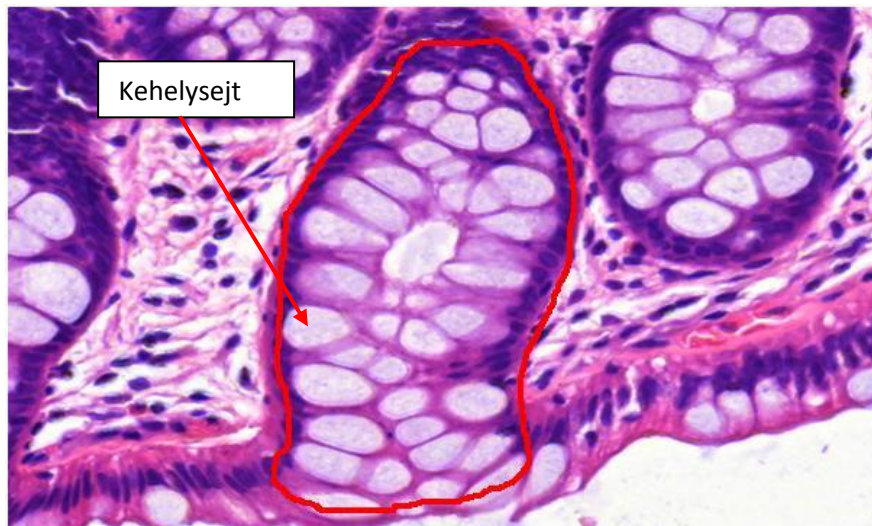
2.1.2. A vastagbél nyálkahártya

A vastagbél nyálkahártyája a hámsejt-, kötőszöveti és izomrétegekből épül fel (3. ábra) [17].



3. ábra: Vastagbél nyálkahártyájának szerkezete [17].

„A hámréteg jellemzője, hogy a sejtek közötti távolság kicsi, szorosan egymás mellett helyezkednek el, megnyúlt hasáb alakúak, és sejtmagjaik egyvonalban vannak [19]. A nyálkahártyába betüremkedő „képződmények” mirigyek, melyek feladata, hogy nyák termelésével síkosítsák a vastagbél falát. A mirigyek alsó részén egy osztódó sejtréteg, a germinatív vagy generatív réteg található. A mirigyek belsejében ún. kehelysejtek is találhatóak” [1] (4. ábra).



4. ábra: Hámréteg, és egy mirigy [17].

Az egészséges nyálkahártya tulajdonságai [17]:

- A nyálkahártya felszíne sima
- A hosszanti tengelyük mentén metszett mirigyek sűrűsége a mintán 6—8 db / mm
- A mirigyek mérete normális, a nyálkahártyáig érnek, irányultságuk párhuzamos, és nem ágaznak el
- Csak a mirigy alsó részén található osztódó sejtréteg
- Sejtjeik száma és felépítése normális
- A kötőszövetben és az izomrétegben nincs elváltozás

2.1.3. A vastagbél betegségei

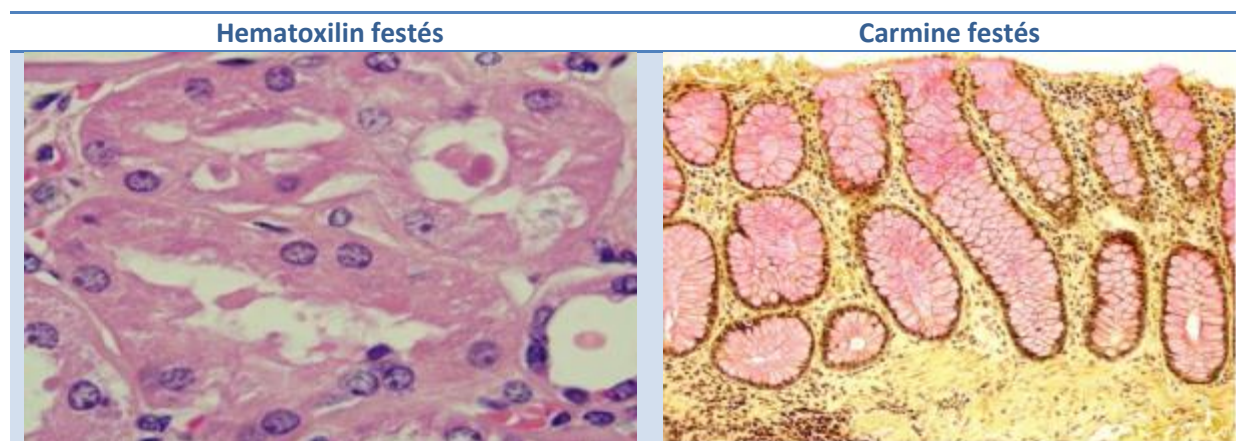
„Egy egészséges vastagbél nyálkahártya sejt daganatos elváltozását több ok is előidézhetheti. Az elváltozás hatására a sejt bizonyos tulajdonságai megváltoznak, így képes lesz magát kivonni a saját, és a környező sejtek növekedési kontrolja alól. Ezek a sejtek új populációt hoznak létre, és a túlnövik a környezetüket, ún. polip vagy hiperplázia jön létre. A hiperplázia azt jelenti, hogy a sejtek normálisnál ugyan gyorsabban szaporodnak, de felépítésük nem változik. Ha a folyamat itt nem áll meg, diszplázia alakul ki, amikor már a sejtek szerkezetében is változások történnek. Később ebből alakulhat ki a rák.” [1][19]

2.2. Biológiai anyagok előkészítésének általános menete

A biológiai anyagok, vagyis az állatból vagy emberből származó szöveti minták vizsgálatához egy előkészítési folyamat szükséges. A cél az előkészítés folyamán olyan jól metszhető ún. blokk készítése, amelyben a biológiai anyag a lehető legkevesebb torzulást mutatja az élőhöz képest. Az előkészítési folyamat első lépése a preparálás, vagyis a szerkezeti felépítés megőrzése, ezután a fixáló folyadék eltávolítása, víztelenítés, beágyazás, metszetkészítés és legvégül a metszet festése következik [9]. A metszet festésére azért van szükség, mert a preparátum körülbelül 5 mikron vastagságú szelet, ami önmagában nem tenné lehetővé a vizsgálatok elvégzését, ugyanis teljesen átlátszó. A jobb láthatóság érdekében ezért a vizsgálatokat megelőzően az előkészítési folyamat végén, a preparátumot megfestik különböző anyagokkal, amelyek hatására az adott anyagra jellemző színt veszik fel a szöveti részek. A festési eljárást követően az ember számára jól vizsgálható szöveti minták jönnek létre.

2.2.1. Hematoxilin festés

A festési folyamat során eozint, phloxint, alkohol és ecetsavat használnak fel a minták megfestésére, mivel a kivett preparátum olyan vékony (5 mikron), hogy mikroszkóppal sem látható, ezért a minta kivételét követően el kell végezni a festési folyamatot a további vizsgálatokhoz. A festés során alakulnak ki a hematoxilin-eozin festésre jellemző jegyek. Az eljárás után a sejtmagok kék, a citoplazma és az erithrocita a rózsaszín különböző árnyalataival festődnek.

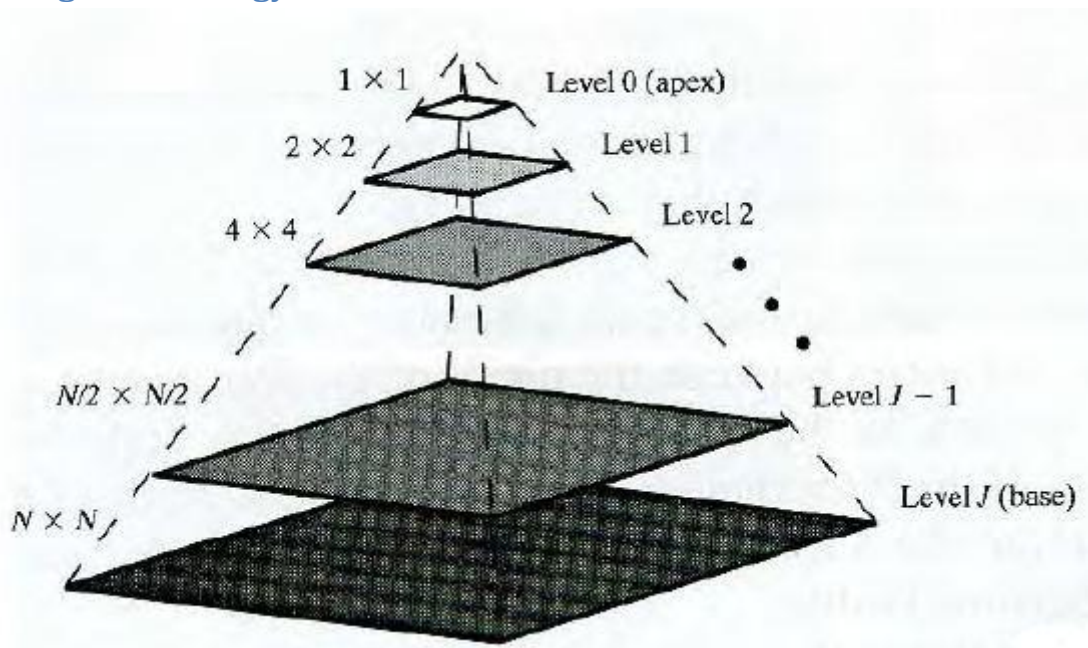


1. táblázat: Balra HE festés, jobbra Carmine festés [6].

Az 1. táblázatban látható a hematoxilin festésű szöveti minta képe (balra). Jól látható, hogy két különböző színnel vannak megjelölve az egyes szöveti részek. Kékkel láthatóak a sejtmagok és rózsaszínnel a citoplazma és az erithrocita. Ezt a festési technikát a hematoxilin és az eozin kezdőbetűiből „HE” festésnek is szokták nevezni. Ez egy nagyon elterjedt módszer, azonban rengeteg másfajta festési eljárást is alkalmaznak a szöveti részek minél jobb azonosítására. A táblázat jobb

oldali képen látható a carmine-festés eredménye, ami szintén egy gyakran használt festési típus. A festések során kialakuló jellemzők nagyon fontosak a textúra alapú kereső algoritmusok számára.

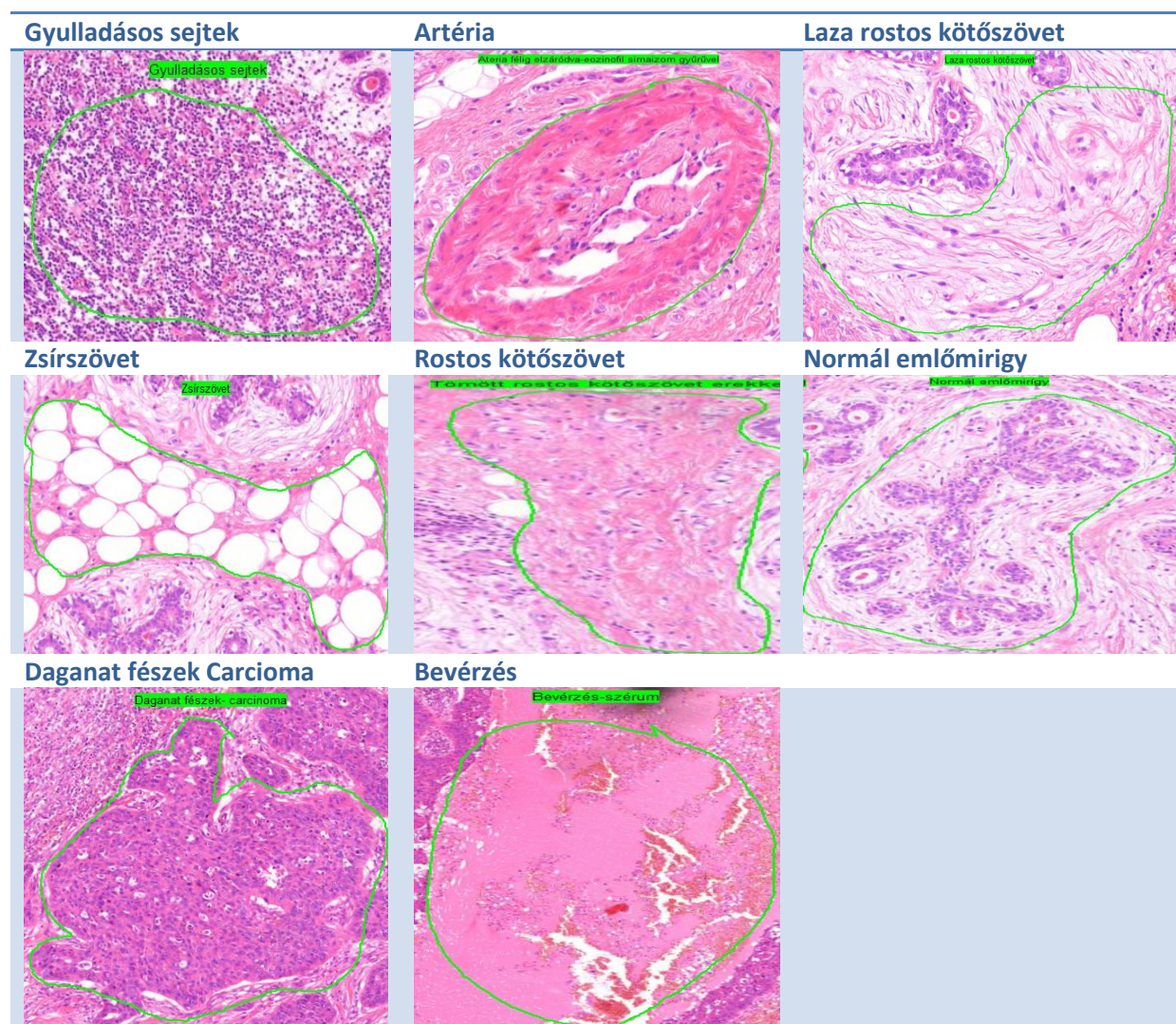
2.3. Digitalizált tárgylemez



5. ábra: A kép piramis [52].

A projekt során felhasznált szöveti mintákat az orvosok mikroszkóppal vizsgálják és szöveti szkennerek segítségével digitalizálják. Az elkészült képek mérete hatalmas lehet, akár több gigabájt. Ekkora méretű képek kezelése a memóriában nem lehetséges, ezért a képeket képpiramisokként kezeli a rendszer. A képpiramis több képet tartalmaz (5. ábra), a legalsó eleme a legnagyobb felbontású kép, minden felette lévő kép sorainak és oszlopainak a száma az alatta levőnek a fele. A képpiramis e tulajdonságából fakadóan nem kell mindig a képeket a legnagyobb felbontáson kezelni, ezzel jelentősen csökkenteni lehet a képek kezelési és feldolgozási sebességét [52].

2.4. A szöveti minták textúráinak sokszínűsége



2. táblázat: A digitalizált tárgylemezek vizsgálata során előforduló textúrák közül néhány.

A vastagbél szöveti minták textúrái igen sokszínűek lehetnek (2. táblázat). A képeken zöld vonallal vannak körbe határolva az egyes orvosi jellegzetességek. Az orvosi szempontból érdekes elváltozások lehetnek: gyulladásos sejtek, artéria, laza rostos kötőszövet, zsírszövet, emlőmirigy, bevérzés, daganat fészek stb. Látható, hogy egy betegség mennyi különböző textúrát képes létrehozni a szövetekben. Nincs lehetőség olyan képfeldolgozó algoritmus létrehozására, amely képes lenne minden textúrát megkülönböztetni. Az algoritmusok általában az egyes textúra jellemzőkhöz igazodva képesek elvégezni a feladataikat.

2.5. DICOM szabvány és PACS

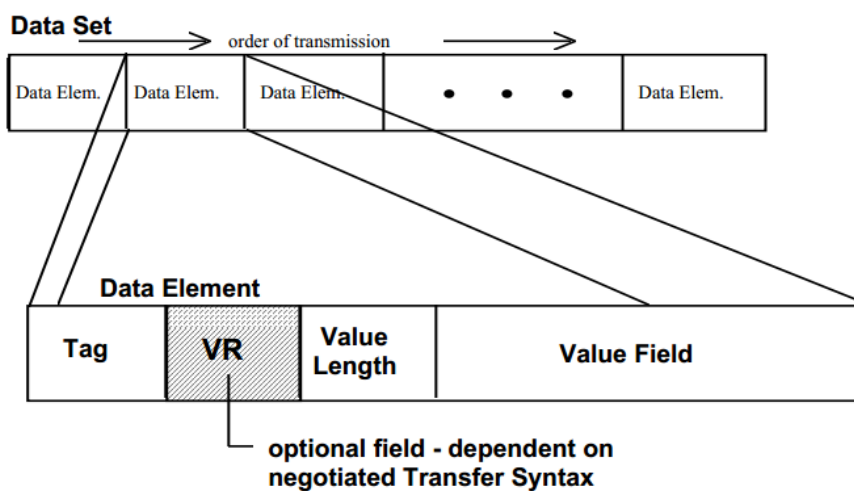
DICOM vagyis „Digital Imaging and Communications in Medicine” egy nemzetközi szabvány az orvosi képekhez és a hozzájuk tartozó információkhoz (ISO 12052). A szabvány meghatározza a kép és a mellé eltárolt adatok formátumát. A legtöbb radiológiai, kardiológiai és sugárkezelési képalkotó eszközben implementálásra került. Több száz berendezés használja a DICOM szabványt, mint formátumot, ezért az egészségügyi ellátásban az egyik legszélesebb körben bevezetett egységes irányelv [2].

A szabvány egy fájlformátum-definíciót és egy hálózati kommunikációs protokollt tartalmaz. A protokoll TCP/IP használatával teremt kapcsolatot a különböző gépek között. Az ACR-NEMA szabvány kiterjesztéseként tervezték, hogy a különböző gyártóktól származó orvosi berendezések, valamint az orvosi információs rendszerek egyes moduljai között megteremtsék a szabványos kommunikáció lehetőségét [3].

Data Element Tag	Data Element Length	Data Element Value								
(gggg, eeee) with VR of SQ	00000F00H	First Item			Second Item			Third Item		
		Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value	Item Tag	Item Length	Item Value
		(FFFE, E000)	0000 04F8H	Data Set	(FFFE, E000)	0000 04F8H	Data Set	(FFFE, E000)	0000 04F8H	Data Set
4 bytes	4 bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes	4 bytes	4 bytes	04F8H bytes

6. ábra: DICOM fájl példa [33].

A 6. ábra egy példán keresztül mutatja be a DICOM fájl felépítését. A Data Element Tag azonosítja a fájlt, ezen mező reprezentálja, hogy egy szabványos DICOM fájlról van szó. A Data Element Length a fájlnek a méretét ábrázolja 4 byte-on. Ezeket a részeket követi az úgymond hasznos adat a Data Element Value részben. Látszódik (7. ábra), hogy több elem is lehet egy fájlban First Item, Second Item, Third Item stb.



7. ábra: Adatelem felépítése [33].

Tehát a fájl több adat elemből áll, minden ilyen elem négy részre bontható:

- Tag: Tartalmaz egy pár 16 bites egész számot, az első a csoportazonosító, a második az aktuális adatelem sorszámát tartalmazza.
- VR: 2 bájtos karakter típus, ami azonosítja az érték reprezentációt.
- Value Legth: 32 bites mező, ami az adatelem teljes hosszát adja meg
- Value Field: Ez a mező tartalmazza a hasznos adatot.

Az F01. számú mellékletben megtekinthető a szabvány összes lehetséges érték reprezentációja.

A DICOM szabvány alapfeltevése, hogy egy fájl egy pácienshez tartozzon, ezért lehetséges, hogy több elemet is tartalmazhat egy fájl. A szabványt úgy alkották meg, hogy egy páciens minden adata egy fájlba legyen eltárolható, ennek köszönhetően a beteg nevét, életkorát, nemét, CT képét, MRI képét stb. egy DICOM fájl tartalmaz. Ebből a megoldásból több előny is származik, a beteg minden adata könnyedén hordozható az egyik egészségügyi intézményből a másikba, az adatok elkeveredése és felcserélése másik páciensével teljesen kizárt, a szabványos megoldásnak köszönhetően pedig a legtöbb orvosi képalkotó berendezés által létrehozott képet és adatot lehet tárolni a fájlban. Továbbá a szabványos fájlt bármelyik PACS rendszerben tárolni és menedzselni is lehet.

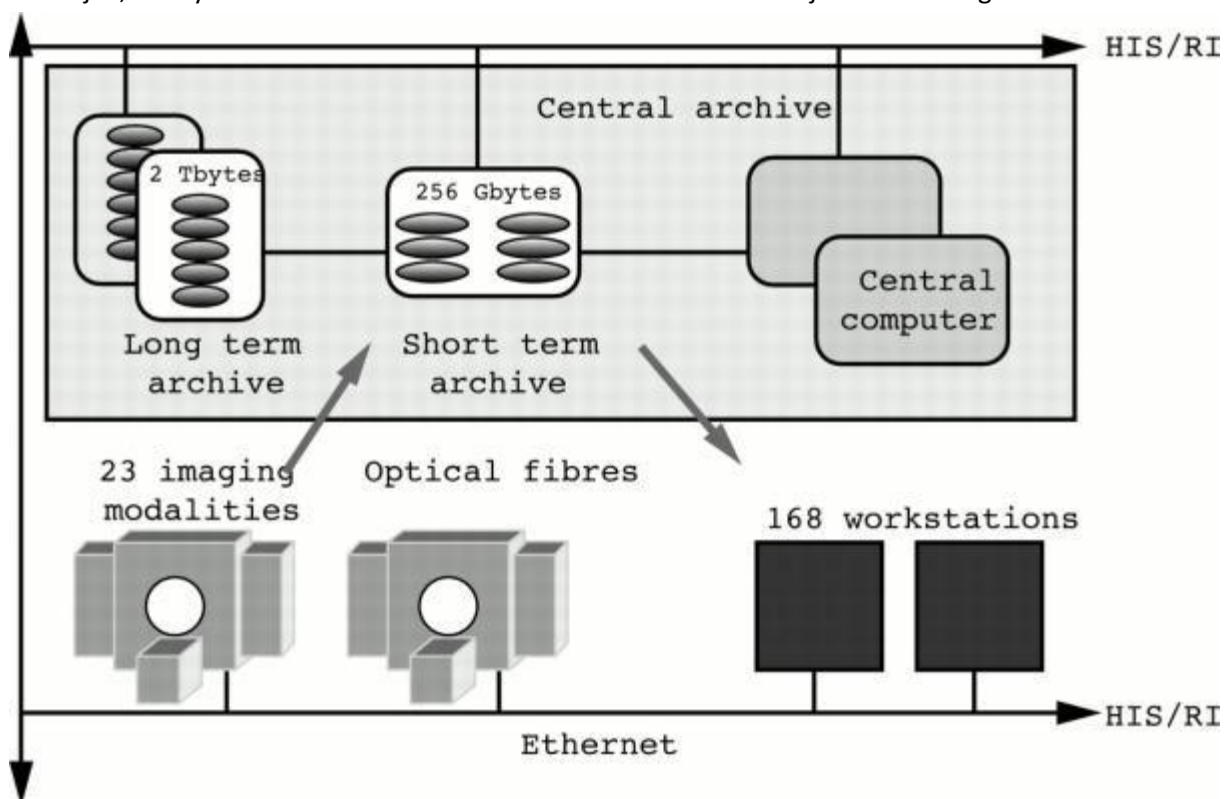
A Value Field mező tartalmazza tehát a fájlban a hasznos információt, tartalmazhat többféle formátumú képet (jpeg, bmp), MPEG-4 AVC/H.264 High Profile / Level 4.1 tömörítésű videót valamint meta adatokat. Bármilyen képalkotó orvosi berendezésből származó képet lehet tárolni ebben a részében a fájlban. Több mint 3000féle metaadatot lehet tárolni az egyes képek mellé. A metaadatok tartalmazhatnak információt a páciensről – név, kórelőzmény, magasság, terhességi státusz –, de lehet olyan adat is, ami a kép- vagy videóalkotó berendezés adatait tárolja, az eltárolható adatok skálája tehát igen nagy. Az egyes páciensekhez tartozó hatalmas információs halmazból kifolyólag kell egy olyan rendszer, ami ezeket tárolja és kezeli, erre szolgál a PACS szerver.

A PACS jelentése *„Picture Archiving and Communication System”*. Az orvosi képalkotásban a képalkotó eszközök, képi adatbázisok, képmegjelenítő kliensek, információs rendszerek, számítógépes hálózata, melynek feladata a megalkotott képek és a hozzájuk tartozó információ kezelése [4].

A PACS rendszert a DICOM szabvány alapján eltárolt képi és szöveges információ tárolására, lekérdezésére használják, a szerver általában össze van kötve egy kórházi információs rendszerrel is. A szabványos formátumú „.dcm” kiterjesztésű fájlokból kinyert információkat a PACS valamilyen elterjedt adatbázis-kezelő rendszer segítségével tárolja. A rendszert elsődlegesen lokális hálózatokra tervezték, ezért semmilyen védelem nincs a hálózati kommunikáció során. A biztonsági kritériumokat a lokális hálózatot üzemeltető rendszergazdáknak kell biztosítani.

2.5.3. Esettanulmány: Hammersmith kórház PACS rendszere

A nyugat Londonban működő Hammersmith kórház 1996 óta átállt a teljesen digitális formában történő adattárolásra [45]. A filmeket csak akkor nyomtatják ki, ha más intézményeknek is szükségük van rájuk, amelyek nem rendelkeznek PACS rendszerrel és nem tudják kezelni még ezen adatokat.



8. ábra: Hammersmith kórház PACS rendszere [46].

A 8. ábra szemlélteti a kórházban használt centralizált tárolási rendszer felépítését. A hosszú idejű tárolásra a rendszer egy 2 terabájtos tárolót (Long term archive), rövid 10 napos periódusokra (a vizsgálatok idejéig – Short term archive) egy 256 gigabájt méretű RAID rendszert használ. A hálózati architektúrája a rendszernek szintén centralizált, a kétméretű tároló rendszerrel, 168 munkaállomással és 23 képalkotó berendezéssel (modalitások). Minden berendezés optikai kábel segítségével pont-pont kapcsolaton keresztül kapcsolódik a központi részhez. Az adatok visszanyerése ezért nagyon gyors, a rövid idejű tárolóból 2 másodperc alatt nyerik ki a képeket és a hosszú idejű tárolóból is mindössze 2 percig tart a lekérdezés. A kórház orvosai a rövid idejű tárolón tudnak képeket elemezni, ehhez egy előlehívási technikát alkalmaznak, vagyis minden másnapra szükséges adatot az éjszaka folyamán lehívnak annak érdekében, hogy a vizsgálatokkor rendelkezésre álljanak. A rendszert integrálták a kórházban működő radiológiai információs rendszerrel (RIS) és a kórházi információs rendszerrel. A rendszer előnyei közé tartozik, hogy a páciensekhez tartozó leletek azonnal rendelkezésre állnak, nem keverednek. Továbbá a telepatológiai megoldások is igen erőteljesen alkalmazhatók a rendszer segítségével, hiszen a gyermekgyógyász és az onkológus akár a kórház különböző szárnyában is lehet, miközben ugyanazon páciens leletét beszél meg telefonon [46].

A kórház számára a PACS rendszer telepítése egy igen költséges megoldás, a megvalósítás költsége 1-2 millió font egy 500 ágygal rendelkező intézménynek [46][47][48][49].

3. Képi adatbázisok

A digitális képalkotás fejlődésével egyre több digitális tartalom létrehozása vált lehetővé. A létrehozás után természetesen az eltárolásról is kell gondoskodni, ugyanúgy, mint az eltárolt tartalom hatékony kereshetőségéről. Erre a célra a legjobb a képeket is adatbázisban tárolni, itt azonban többféle keresési lehetőségre nyílik lehetőség a hagyományos relációs adatbázisokkal szemben.

A képi adatbázisokban való keresésre két módszert alkalmaznak. Az egyik megközelítés a szöveg alapú, a másik a tartalom alapú keresési módszer. A szöveg alapú keresési megoldások manapság nagyon elterjedtek, a legtöbb kereső rendszer ezt használja. Ennél a módszernél az egyes képekhez hozzárendelnek szöveges leírókat, amik alapján majd az adatbázisban keresni lehet. Ennek a módszernek két nagy hibája van. Az első, hogy nagymértékű manuális emberi munkát kell végezni, a másik, hogy a szöveges hozzáfűzés az emberi szubjektivitáson múlik. A szöveg alapú keresési módszer hibáinak kiküszöbölésére sok más módszert találtak ki, amelyek közül elterjedtté váltak a tartalmi alapú keresési technikát [10,11]. A tartalmi alapú visszanyerési technikát az irodalom sokféleképpen említi, „Content-Based Image Retrieval” (CBIR), de ismertek a „Query By Image Content” (QBIC) és a „Content-Based Visual Information Retrieval” (CBVIR) kifejezések. Ezek mind ugyanazt takarják, de a legismertebb a CBIR [12]. A tartalmi alapú keresési módszer gépi algoritmusok segítségével állítanak elő a képről jellemzőket.

A CBIR jellemzői [10]:

1. szint: Ez a szint a kép primitív jellemzői alapján történő keresést deklarálja. Ilyen jellemzők a szín, textúra, forma, vagy a kép objektumainak térbeli elhelyezkedése. Erre a keresésre az egyik példa, ha olyan képet keresünk, amin sárga csillagok találhatók és ezek egy gyűrűbe rendezettek. A lekérdezések e szintje tehát jellemzőket használ, mint például a sárga színárnyalatai, ezek a jellemzők közvetlenül levezethetők a képekből anélkül, hogy valamilyen külső ismeretbázist felhasználtak volna a kereséshez.

2. szint: Ez a szint a képen található objektumokból logikai következtetéssel meghatározott jellemzői alapján teszi lehetővé a keresést. Ezen a szinten a következő keresési példa képzelhető el: olyan képet keressünk, amin az Eiffel torony látható.

3. szint: Elvont sajátosságok alapján történő keresési módszer ez a szint. Erre a keresésre a legjobb példa az, ha olyan képet keresünk, ami például szenvedést ábrázol [10].

A 2. és a 3. szinteket együttesen szemantikai képviszanyerésnek nevezzük. Az 1. és 2. szint közötti különbséget vagy hézagot pedig szemantikai hézagnak nevezzük [13,14].

3.1. Az orvosi képek archiválása informatikai eszközökkel

Az orvosi tudás nagyon különböző formákból ered. Például vegyük a két fontos osztályát az orvosi tudásnak: anatómiai és élettani tudás. Az anatómiai információ a vizuális megjelenésen múlik, lehet egy kinagyított szív vagy különböző morfológiai paraméterek. Az élettani információk a biológiai folyamatokból erednek, amelyek nem feltétlenül vizuálisak. A fiziológiai adatok származhatnak például a metabolizmusból, vagy lehetnek numerikus adatok, melyek a fiziológiás tesztekben származnak, mint például a vérnyomás szisztolés és diasztolés értékei stb. Gyakran az anatómiai és

fiziológiai információk egyidejűleg vannak érvényben, de csak a szöveg- vagy számszerű információkat, mint például a vér kémiai értékét tároljuk adatbázisban [15].

Orvosi képek származhatnak orvosi képalkotó modalitásokból, többek között: CT, MRI, mikroszkópia, röntgenfelvételek. A képek diagnosztikus célokból történő elemzése komplex folyamat. Az orvosi felhasználásnál fontos lehet a kép tartalma, valamint azok az információk, amik a képhez tartozhatnak, de nincsenek rajta a képen. Az egészségügyi rendszerben a képek tárolására a DICOM szabványt és a PACS rendszereket használják, e megoldásokban a képalkotó berendezések által készített képek mellé az orvos páciensadatokat is eltárol. Az így előállított DICOM fájlt a PACS rendszer eltárolja, a hozzá csatlakoztatott kórházi információs rendszerben pedig megtekinthető a pácienshez tartozó állomány.

A DICOM szabványra és a PACS-ra épülő rendszer nem a hagyományos értelemben vett képi adatbázis rendszer. A szöveges tartalom alapján indexelt képi adatbázisok hátránya, hogy az emberi szubjektivitáson múlik az indexelés minősége, és ez befolyásolja a későbbi kereshetőséget, előnyei közé tartozik az egyszerűsége. A CBIR rendszerek hátránya, hogy nincs olyan algoritmus, ami pontosan képes modellezni a képeken található tartalmi információkat. A tárolt képek olyan sokfélék lehetnek, hogy nem lehetséges olyan algoritmust létrehozni, amely minden képből pontosan képes meghatározni a tartalmi jellemzőket. A DICOM megoldás hátrányai közé tartozik, hogy szükséges hozzá a PACS rendszer és a szabványnak meg kell felelnie. Ugyanezen tulajdonságból kifolyólag, azonban ennek a megoldásnak az a nagy előnye, hogy gyakorlatilag minden DICOM fájl egységes és az összes képalkotó berendezés képes adatot küldeni a szabványos kommunikációs csatornán a PACS szervereknek.

4. Hasonló projektek

4.1. Összehasonlítási szempontok

A hasonló projektek fejezetben mások által elkészített rendszerek (3. táblázat) elemzésére kerül sor több irányvonal mentén. Vizsgálat alá kerülnek képi adatbázisok és ezek keresési lehetőségei, annotációs lehetőségek, és olyan függvénykönyvtárak, melyek valamilyen módon hasznosak lehetnek a projekt szempontjából.

Név	Típus
ASH Image Bank	Textúra alapú képi adatbázis
ImageAfter	Textúra alapú képi adatbázis
TInEye	Textúra alapú képi adatbázis
RevIMG	CBIR rendszer
Windsurf	CBIR rendszer
mdcm	DICOM fejlesztői környezet
ClearCanvas	DICOM fejlesztői környezet
DISOM is Easy	DICOM fejlesztői környezet
dcm4chee	PACS szoftver

3. táblázat: Az áttekintett projektek összefoglalása.

4.2. ASH Image Bank

Az ASH Image Bank a világ egyik legnagyobb professzionális egyesülete, ami a vérzavarok okaival és bánásmódjával foglalkozik. Az egyesület célkitűzése, hogy elősegítse a vérzavarok megértését, diagnosztizálását, valamint komoly előrelépést érjenek el a megelőzésük területén [20]. A társaság létrehozott egy webalapú képi adatbázist, ami több mint 2100 hematológiai képet tartalmaz. Ehhez az adatbázishoz mindenki szabadon hozzáférhet, kereshet benne és ingyenesen letöltheti a találatokat [21].

Keresési lehetőségek:

Az adatbázisban kulcsszavak megadásával vagy kategóriákban lehet keresni. Az első megoldás nem célravezető akkor, ha nem szakavatott személy próbálja használni a keresőmotort. Ebben az esetben nem lehet ugyanis tudni, hogy milyen kulcsszavak adhatnak releváns eredményt. A másik keresési lehetőség ebből a szempontból sokkal jobb választás. Ebben a keresési módban a weboldal kilistázza az adatbázisban fellelhető képek kategóriáit úgy, hogy melléjük van írva, mennyi releváns találat létezik. Így minden egyes kategóriára biztosan lesz eredmény. A visszakapott találatok rendezett rácsos megjelenítésben láthatók. Minden egyes elem külön megtekinthető, a kép mellett megjelenik a szerző neve, a publikálás dátuma, a kategória neve, a kép mérete, megtekintések száma, valamint egy egysoros leírás arról, mi látható a képen. A feltöltés mindenki számára engedélyezett egy rövid regisztráció után. Feltöltéskor néhány adat megadása szükséges, mint például a szerző neve, kép címe, rövid leírás.

Értékelés:

Mindent összevetve, az ASH Image Bank egy kezdetleges orvosi képi adatbázis. Egy ilyen adatbázis sokkal jobban kereshető kellene hogy legyen az olyan felhasználók számára is, akik nem szakavatottak a témában. Jelenleg csak a kategória alapján lehet hatékonyan keresni. Nincs rendes annotálhatóság, nincs lehetőség részletes adatok megadására a feltöltendő képről. Ebből kifolyólag a keresés is igen szegényes lehetőségeket nyújt. További hátránya ennek a rendszernek, hogy mindenki szabadon feltölthet az adatbázisba, nincsen semmilyen ellenőrzési rendszer beépítve, a hibás adatok kiszűrésére nincs mód. Ugyanakkor előnyei is vannak a rendszernek. Az egyik az, hogy egy találati kép több kategóriába is tartozhat. Ez azért pozitív, mert egy kép többfajta betegséget is tartalmazhat, és így több keresésnél is releváns találat lehet. Másik előny, hogy a keresésnél a kategóriák kiválasztásánál csak olyan kategóriák jelennek meg, melyeknél releváns találatot is kaphatunk.

Az ASH Image Bank a fentiek szerint szövegalapú keresési módszert ajánl a képi adatbázisában, nincs semmilyen tartalmi alapú visszanyerési vagy keresési lehetőség.

4.3. ImageAfter

Az ImageAfter egy nagy online textúra alapú képi adatbázis. A keresések eredményei szabadon letölthetőek és szabadon fel is használhatók bármilyen tevékenység során [22].

A lekérdezést webes felület segítségével tették lehetővé. Itt is lehetséges előre definiált kategóriákban keresni, ennek az opciónak a választása során nem lehet tovább szűkíteni a releváns találatok számát további alkategóriák kiválasztásával, hanem az adatbázis minden, a kijelölt kategóriára illő elemét visszaadja az algoritmus. Ez sajnos sokáig is eltarthat, ami ronthatja a felhasználói élményt. Másik lehetőség, hogy textúrára illő kulcsszó alapján kereshetünk. Ezt egy

beviteli mezőbe kell beírni, amint néhány karakter bekerül a beviteli mezőbe, egyből felugrik egy lista a lehetséges kulcsszavakról. A találati ablakban a kép mellett a méretét mutatja a táblázat megfelelő sora. A kép fölé helyezve a kurzort megtekinthetjük a hozzá tartozó leírást, ami alapján a lekérdezéseket végrehajtja a keresőmotor. A találati listában is van egy további szűrési lehetőség, egy lenyíló lista, amiben színkódok jelennek meg hexadecimálisan.

Értékelés:

Sajnos ez a rendszer elég lassú. A kétféle módon kipróbált keresés során a „woods” kulcsszavas keresés és a legördülő listából szintén e kulcsszóra vonatkozó keresés eredményhalmaza nem volt konzisztens.

4.4. TinEye

Az eddigiek során bemutatott rendszerekben az volt a közös, hogy mindegyik csak szövegalapú keresési lehetőséget nyújtott. A TinEye-ban a képek tartalma alapján lehet kereséseket végrehajtani, ez alapján a fordított képkeresési szolgáltatások közé tartozik [23].

Használatkor egy képet kell bemenetül adni, és a megadotthoz hasonló képeket fog visszaadni releváns találatként. A keresés tárgya többféle módon is megadható, fel lehet tölteni fájlt a felhasználó saját gépéről vagy beilleszthető egy képre mutató URI, vagy böngészőplugin is létezik az összes elterjedt böngésző (Firefox, Internet Explorer, Chrome, Opera, Safari) alá a bemenet megadásához.

A rendszert nem lehetett tesztelni a projektben használatos szöveti képekkel, mert azokra nem adott találatot. A 4. táblázatban bemutatott példa ezért egy egyszerű keresés eredményét tartalmazza. Az első oszlopban a bemeneti kép látható, ami egy ezüstmetál színű BMW e46 típusú személygépkocsit ábrázol előlről, oldalról.

A TinEye képkeresés eredményei		
		
Bemeneti kép	Kimenet	Kimenet

4. táblázat: TinEye keresési eredményi.

A táblázat második és harmadik oszlopában a kimenetek, vagyis a releváns találatok találhatók. Amint látható, ugyanazon típusú autót talált meg ugyanazon színben és ugyanabból a nézetből.

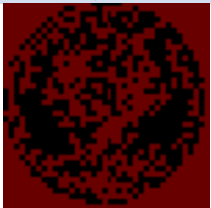
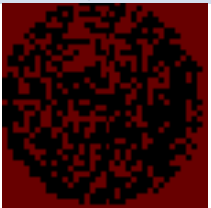
Értékelés:

Az oldal tesztelése során rengeteg kép kipróbálásra került. Sajnos a legtöbb próbálkozás során nem vagy kevés eredményt szolgáltatott a rendszer. Sok esetben egyáltalán nem adott eredményt, vagy csak magát a beadott képet, amit megtalált egy másik weboldalon. A fent bemutatott példa a legjobb esetek közé tartozik. Például egy körtét ábrázoló képre sem adott releváns találatot, pedig körtét

ábrázoló képből (egy elterjedt kereső szöveges keresési lehetőségének eredményei alapján) az interneten elég sok található. Egyelőre tehát elég kiforratlannak találjuk.

4.5. RevIMG

A RevIMG is egy fordított képkeresési rendszer, ami online használható. Az oldalon leírtak szerint az általuk használt algoritmusok szín, méretek és formák alapján keresnek [24]. A keresési lehetőségek hasonlítanak a TinEye rendszeréhez. Itt is fel kell tölteni egy képet, vagy megadni az URI-t, amin a kép található. Ezek után sajnos csak megadott kategóriákban lehetséges a keresés. A kategóriák között azonban nem szerepel orvosi rész, ezért szöveti mintákkal ezt a rendszert sem lehetett tesztelni. A találati eredmények mellett látható, hogy hány százalékos egyezést talált a bemeneti képpel a rendszer.

A RevIMG képkeresés eredményei				
				
Bemeneti kép	Kimenet	Kimenet	Kimenet	Kimenet
	59,07%	50,28 %	46,09%	43,84%
				

5. táblázat: RevIMG keresőmotor eredményei.

Tekintsük azt a keresési példát (5. táblázat), amelyben egy George Washingtont ábrázoló aranyszínű érmére kerestünk. A táblázat első oszlopában látható maga az érme, a többi oszlopban a találatok listája. Az eredményképek alatt szerepel az egyezés százalékos értéke, valamint az oldal által generált kép, amin az eredetivel megegyező részek szerepelnek fekete színnel jelölve. Az legjobban egyező találat 59,07%-os, ez maga a kép, amit az algoritmus bemenetként megkapott. Jól látható, hogy a többi találat is érme, de nem azonos háttér- és előtérzínnel. Az oldal nyújt egy másik szolgáltatást is, az online rajzolást, amelynek a kipróbálására nincs lehetőség, de a leírás alapján a következő módon működik: egy kép megrajzolását követően a képen ki kell jelölni egy ROI-t, azaz egy számunkra érdekes részt, amely a keresőmotornak átadva keresési bemenetként használható [24].

Értékelés:

A RevIMG egész jó keresési lehetőséget biztosít, azonban a kategóriák szűkossége miatt igen kevés dologra lehet használni. Ha kategórián kívüli képre kell keresni, akkor fals találatokra számíthat a felhasználó. A rendszer tesztelése során (az összehasonlíthatóság miatt) többek között a TinEye-nak adott bemeneti képeket is kipróbáltuk. Meglepő módon erre is adott a rendszer releváns találatokat, amik átlagban 10%-ban egyeztek. A 4. táblázatban bemutatott példában az ugyanolyan képekre sem volt 100%-os az egyezés, ez az algoritmus hatékonyságát minősíti, ugyanúgy, mint a fals találatok száma.

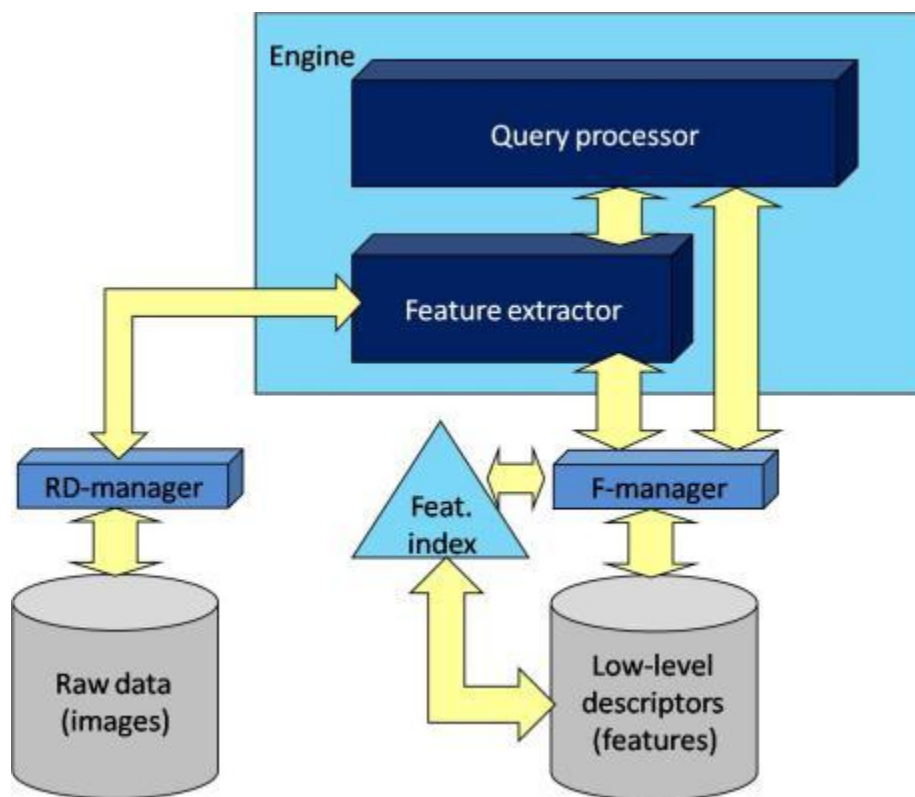
4.6. Windsurf

A Windsurf (**W**avelet-based **IN**dexing of Image**S** Using **R**egion **F**ragmentation) egy Java alapú függvénykönyvtár, aminek a célja egy általános keretrendszer létrehozása azok számára, akik képviszanyeréses keresést szeretnének alkalmazni. A rendszert „QPL” licenz alatt adták ki, tehát saját használatra, oktatási és kutatási célokra használható fel ingyenesen [25].

Összességében a Windsurf rendszert a következők jellemzik:

- Diszkrét Wavelet transzformációs filter használata HSV színtérben,
- Régiószegmentáción alapuló k-means algoritmus,
- A régióösszehasonlításra a Bhattacharyya távolságfüggvényt alkalmazzák [5][26].

A 9. ábra szemlélteti a rendszer felépítését és a moduljait.



9. ábra: A Windsurf keretrendszer architektúrája [27].

Négy fő részből áll a rendszer:

- *Query Processor*, ez a modul lekérdezések készítésére alkalmas.
- A *Feature Extractor* az alacsony szintű jellemzőkkel és a képek szegmentációjával foglalkozik.
- Az *RD* és *F manager* modulok a képek nyers adatainak és a jellemzőinek a meghatározásához szükségesek.
- Végezetül az alacsony szintű jellemzők hatékony hozzáférése érdekében a *Feature Index* modult alkalmazza a rendszer. A keretrendszer többfajta indexelés használatát támogatja.

A *Query Processor* szekvenciális és hatékony indexalapú algoritmusokat tartalmaz a különböző típusú lekérdezésekhez (teljes lekérdezés, részlekérdezés, k-NN vagy Skyline lekérdezések), és három különböző paraméterezése lehetséges:

- A távolságfüggvény, ezzel összehasonlítja a képrégiókat,
- Az egyezési típus, ez meghatározza azt, hogy hogyan párosítsák össze azokat a részeket a képeken, amiket majd összehasonlítanak,
- Aggregáló függvény, amely egyesíti a távolságértékeket az összehasonlított régiókban, mindenhol a kép távolságértékének környezetében.

A *Feature Extractor* egy absztrakt része a keretrendszernek, a feladata, hogy megossza a rendszer osztályai között az alacsonyszintű képreprezentációt, a képszegmentációt és a régió jellemzőinek a meghatározását [27].

Értékelés:

A rendszer csak a jellemzők meghatározására és adatbázisban való tárolásra alkalmas. Ez a függvénykönyvtár egy jó kiindulási alapja lehetne számos tartalmi alapú képkeresési megoldásnak, azonban csak „QPL” licenc alatt lehet használni.

4.7. mdcm

Az mdcm egy Colby Dillion által fejlesztett függvénykönyvtár, aminek a 0.9.5 verziója lett tesztelve. Az SDK teljesen ingyenes LGPL licenc mellett [28].

Az mdcm fejlesztői környezet az orvosi képek és hozzájuk tartozó információk DICOM szabvány szerinti tárolására, „.dcm” fájlok létrehozására és a PACS szerverrel való kommunikációra lett kifejlesztve. A fejlesztők weblapja mindössze néhány linkből álló oldal, a közepén egy linkkel, ahonnan letölthetőek a függvénykönyvtár fájllai. A letöltött fájlokat a fejlesztőkörnyezetben a saját programhoz hozzáadva máris elkezdhető a kódolás. A szabványos fájl létrehozása a mellékelt példaprogram alapján nem nehéz. Az elkészített fájlhoz hozzá lehet adni saját tagokat, amelyek bekerülnek a metaadatok közé, az elmentett fájlt be lehet tölteni a megfelelő metódusok használatával. A DICOM fájlok képeket is tartalmazhatnak, sajnos ez a funkció nem működik megfelelően, ugyanis a betöltött kép byte tömbbé alakítása után a meghívott függvények nem mentették el az átadott képet. Az elmentett fájl egy ingyenes DICOM nézővel megtekintve, sajnos csak egy üres fekete képet tartalmaz. A fejlesztők oldalán lévő fórum ide tartozó részei szerint ez a függvénykönyvtár hibája.

Értékelés:

Mindent összevetve az mdcm nehezen lenne a kitűzött feladathoz alkalmazható. Szegényes a rendelkezésre álló dokumentáció is. A fejlesztők az évekkel ezelőtt jelzett hibákat a mai napig sem javították ki.

4.8. ClearCanvas

A ClearCanvas cég az orvosi képek manipulálására és a képek tárolására szakosodott. Az ő általuk fejlesztett Workstation 7.0 (32-bit) szoftver került tesztelésre. Ezt a fejlesztői környezetet a DICOM formátumú képek létrehozására és a PACS szerverrel való kommunikációra hozták létre [29]. Regisztráció után tölthető le a nyílt forráskódú csomag. Az általunk használt fejlesztőkörnyezetben elhelyezve a függvénykönyvtár nem fordult le. Rengeteg hiba volt a forráskódban, hiányzó DLL-ek, fájlok stb. A hiba felderítésében semmilyen eszköz nem állt rendelkezésünkre, még a hivatalos oldalon sem. A problémák megoldása után elkezdtük a forrás tesztelését, de sajnos a letöltött

csomagban lévő példamegoldás sem fordult le, ezért ezt nem volt értelme a fejlesztői környezet használatának, mivel e nélkül nem lehet értelmezni, melyik függvény milyen funkciót lát el.

Értékelés:

Mivel a letöltött forráskód le sem fordult a fejlesztői környezetben, ezért az egyébként rengeteg mindent magában foglaló kód felhasználása nem lehetséges a projektnél.

4.9. DICOM is Easy

A „DICOM is Easy” megoldása egy fórumon található. A fejlesztői környezet licence ingyenes, tehát szabadon felhasználható bármilyen projekthez, néhány kisebb megkötéssel. 32 bites és 64 bites verzió is van belőle. Két részből áll a rendszer, van egy kliens és egy szerver rész [30].

A kliens részben lehetőséget kap a fejlesztő elkészíteni a saját DICOM fájljait, feltöltve a képekkel és a megfelelő tagokkal. Ezeket a fájlokat lehet menteni, olvasni és szabványos módon elküldeni egy PACS szerver felé. Mindezeket a műveleteket néhány sor írásával könnyedén el lehet végezni. A PACS szerver rész egy MySQL szerverhez csatlakozik, amiben tárolja az adatokat. A szerver megkapja a DICOM fájlt és a benne lévő adatokat – a képet és adattagokat –, majd a MySQL adattábláiban tárolja. Ha lekérdezést kap a PACS, akkor szintén ebből az adatbázisból szedi ki az információkat, majd becsomagolja egy „.dcm” kiterjesztésű fájlá, és azt adja vissza a lekérdezés eredményeként. A már említett fórumon található egy 12 bejegyzésből álló útmutató sorozat, ami nagy segítséget nyújt a rendszer kipróbálásához.

Értékelés:

A rendszer nagy előnye a könnyű kezelhetőség és a jó dokumentáció. Ezekkel a tulajdonságokkal egy nagyon jól felhasználható rendszert szolgáltatnak a fejlesztők számára, amivel könnyedén kezelhetők az orvosi adatok. Ezek a jó tulajdonságok azonban csak a SDK kliensoldali részére mondhatóak el. A tesztelés során a PACS szerver rész folyamatosan hibákat generált, ezért ezt nem lehetett működésre bírni.

4.10. dcm4chee

A dcm4chee egy nyílt forráskódú PACS szoftver, amit az orvosi képek tárolására, menedzselésére fejlesztettek ki. Java programnyelven készítették el a szoftvert, ami egészében a DICOM szabvány szerint működik. A kifejlesztett szerver szoftver képes futni Windowson, Linuxon és Mac OS-en is, mindemellett támogatja a PostgreSQL, MySQL, Oracle, SQL Server adatbázisokat [31].

A tesztelt szoftver telepítése igen egyszerű, mint ahogy a webalapú kezelőfelület is. A webes kezelőfelületen csak az alapvetően szükséges funkciók találhatóak meg. A szerver működése a DICOM is easy SDK-val készített tesztszoftverrel került kipróbálásra. Amilyen egyszerű telepíteni, olyan nehéz használni a rendszert. A DICOM fájlok tartalmaznak számos olyan metaadatot, amiket az adatbázisban megfelelően inkrementálni kell a kereshetőség érdekében, például ilyen a képhez tartozó tanulmánynak a sorszáma. Ezeket az információkat csak tapasztalati úton tudja megszerezni az SDK-t használó fejlesztő.

Értékelés:

A szoftver telepítése egyszerű volt, de használata nem, mivel nem áll rendelkezésre sok információ a működésről. Meglepő módon a rendszer engedélyezi több megegyező PatientID mező meglétét az adatbázisban, de nem enged ugyanolyan nevű páciens felvinni az adatbázisba. Mindent összevetve azonban a szoftver jól használható egyszerű többplatformos rendszer.

4.11. Következtetések, értékelés

A megismert rendszerek között számos olyan található, amely nagyon jó alapot képezne a projektnak. Kiemelendő a „DICOM is Easy” megoldása, amely egyszerűségével és használhatóságával kitűnik a többi közül. Mivel elsődlegesen orvosi célokra használható textúra alapú képi adatbázis tervezése a cél, ezért ez a fejlesztői környezet jó választás lehet, mivel az egészségügyi ellátásban leggyakrabban használt szabványon alapul. A DICOM szabvány használata mellett szól az az érv, hogy ha ebben lenne megvalósítva a rendszer, akkor minden e szabvány mentén képet előállító készülék adatát tudná fogadni a rendszer. Ez a lehetőség egy nagyon széleskörű használhatóságot jelentene a rendszer szempontjából, ugyanis egy olyan textúra adatbázist lehetne létrehozni, amelyben minden orvosi képalkotó berendezésből származó textúra jelen lehetne. Így mások által implementált textúra alapú kereső algoritmusnak is szolgáltatnánk adatokat. Mivel azonban a tesztelések során csak a kliensoldali része működött a „DICOM is Easy” fejlesztői környezetnek, ezért a PACS részt nem lehet használni a projekt során. Lévéen a DICOM szabvány szerint elkészített fájlokat bármilyen más gyártó vagy fejlesztő által készített PACS is képes fogadni, ezért a dcm4chee fejlesztése is alkalmazható, mivel az jól működött a tesztelések alapján.

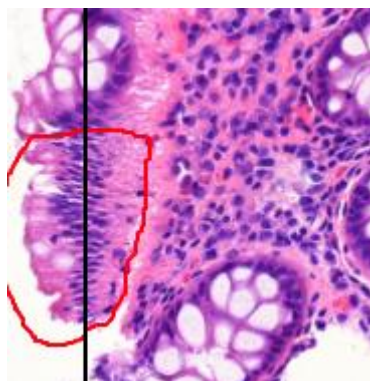
5. Textúra alapú módszerek

A rendszerbe beépítésre került egy általunk tervezett és megvalósított textúraklasszifikáló algoritmus is, amelynek feladata az orvosok által értékelendő szöveti struktúrák különválasztása. A szöveti struktúrákon az orvos méréseket végezhet, a textúraminták pedig feltölthetők a DICOM adatbázisba a keresés megkönnyítése és automatizálása érdekében. A következőkben az algoritmus által felhasznált elméleti alapokról írunk.

5.1. Textúradarabok definiálása

Az analízis során mindenképpen sok pixelt kell együtt vizsgálni ahhoz, hogy a textúrára számított tulajdonságok értelmet kapjanak.

Az együtt vizsgált pixelek kiválasztása alapvetően befolyásolja az algoritmus által képzett végső eredményt. „Textúradarabok” képezhetők egyszerű rácsos felbontással, de így egy rácselembe több textúrából eshetnének darabok. Ilyenkor ugyan adódik a továbbbontás lehetősége (split and merge megközelítés), de elképzelhető, hogy sok bontás után egy jelentősen kisebb textúradarabon már nem mutatkoznak vagy nem mérhetők ugyanazok a jellegzetességek. Másrészt az, hogy egy adott rácselemnek a továbbbontása szükséges-e, szintén nem egyszerűen eldönthető a természetes textúrák esetében. Egy egyszerű esetben pl. él közelében bonthatnánk, de egy hámszövet-részben az az él a textúra szerves részét képezi (10. ábra).



10. ábra: Pirossal: hámszövet-darab. Feketével egy olyan lehetséges rácsvonal látható, ami nagyon nehezen lenne kezelhető.

A hagyományos rácsalapú megközelítés helyett jobb eredményt adhat az ún. superpixel alapú megközelítés, amelyben a cél a kép szegmentálása olyan kis részekre, amelyek kompaktak, közel azonos méretűek, és megközelítőleg homogének [36]. Egy ilyen felbontás szintén egy rácsszerű struktúrát definiál, de úgy, hogy a rácselemek/superpixelek határai illeszkednek a kép jellegzetességeihez (élek). Ilyen superpixeleket generáló algoritmusok több más mellett a Turbopixels [40], az N-Cuts [41] és a SIFT [36] ([37]-ben fellelhető válogatás alapján). A fentiek közül az N-Cutshoz nem találtunk publikusan elérhető cikket, emiatt a másik kettőt vizsgáltuk elsősorban.

5.1.1. Turbopixels

Ez az algoritmus úgynevezett „geometric flow” alapú megközelítést biztosít superpixelek generálására. Az ilyen algoritmusok valamilyen kezdeti határokat egy alkalmasan definiált „flow” segítségével úgy frissítik folyamatosan, hogy azok a megkívánt superpixel-határok felé mozogjanak. A Turbopixels-nek paraméterül megadható a kívánt superpixelek száma. Az algoritmus kezdetben szabályos rácshálóban helyezi el körkörös formájú superpixel-kezdeményeket, amelyeket a kisebb gradiensű részek felé mozgat el, annak az elkerülése érdekében, hogy a superpixel a kezdet kezdetén erős élre essen. Ezután iteratíván „evolválja” a superpixel-kezdemények határait, úgy, hogy a superpixelek egymásba folyását váz-számítás segítségével megakadályozza (az aktuális superpixelek által meghatározott váz határain nem terjedhet túl a superpixel területe a határokat módosító lépésben) [40].

5.1.2. SLIC superpixels

A SLIC egy K-közép-jellegű megközelítés superpixelek generálására. Két paramétere van: a kívánt superpixelek mérete (pl. 10 megadására körülbelül 10x10-es superpixeleket generál), és egy „kompaktság” paraméter, amely azt befolyásolja, hogy mennyire törekszik az algoritmus a superpixel belső homogeneitására a kompaktság rovására. A méret helyett egyes implementációkban előfordul, hogy a superpixelek számát kell megadni. Az algoritmust egy rácshálóra inicializálják (felvesznek egy-egy superpixel-középpontot a megadott régió méretnek megfelelő szabályos rácshálóban), majd a kezdeti superpixel-középpontokat itt is a kisebb gradiensű részek felé viszik egy 3x3-as ablakban. Ezután minden pixelt a legközelebbi superpixel-középponthoz rendelnek. A „legközelebbi” fogalmát a saját távolságmértékükkel mérik, amely a pixel LAB színtérben vett közelségét jutalmazza, a térbeli (síkbeli) távolságot pedig (a kompaktság paraméter értékétől függő mértékben) bünteti. Végül minden superpixelhez új középpontot számolnak, s e műveletet iteratíván folytatják a kívánt eredmény eléréséig [36].

A szerzők bemutatták a SLICO nevű egyparaméteres verziót is, amelynek az inputja csak annyi, hogy hány darab szuperpixelt szeretnénk [36,37]. Ehhez azonban sem forráskód, sem leírás nem elérhető egyelőre.

5.2. Textúravizsgálat

A textúradarabok azonosítása után a feladat szempontjából megfelelő, mérhető paramétereket (jellemzőket, tulajdonságokat) kell találni a textúrák leírásához. [41] (Tuceryan és Jain cikkére hivatkozva) az ismert módszerek között négy nagy csoportot azonosít: a statisztikai, a geometriai, a modell-alapú és a jelfeldolgozást használó technikákét. A statisztikai módszerek a szürke értékek síkbeli eloszlását analizálják, és az eloszlás alapján statisztikai információkat generálnak. Ezek közül a leggyakrabban az „együttes előfordulási” (cooccurrence) jellemzőket [42] és a szürkességiérték-különbséget (gray level differences [43]) használják. A geometriai módszerek a textúrát mint egyszerű primitívek által felépített egységet vizsgálják, ahol a primitívek például élek, régiók lehetnek, és ezekre számítanak statisztikákat (méreteket, irányokat, intenzitást, kompaktságot stb.) vagy elhelyezkedési szabályokat. A modell-alapú módszerek parametrikus modell becslésével próbálkoznak a textúra leírására, a jelfeldolgozáson alapuló módszerek pedig a képben lévő frekvencia-információ felhasználására törekednek, ezek között is a legegyszerűbb élapapú módszerektől (élsűrűség per egységnyi terület, Rosenfeld és Thurston) a Fourier- vagy wavelet-transzformáción alapuló módszerekig sokféle technika van. Említésre érdemes a Local Binary Patterns, amely elég egyszerű elven alapul, és tulajdonképpen két osztály között (a statisztikai és geometriai modellek határán) helyezkedik el. Az LBP gyakran használt módszer, mivel egyszerű metodika segítségével meglepően sok információt képes kifejezni.

5.2.1. Local Binary Patterns

Az LBP működési elve a következő: minden pixelt a képen egy bináris számmal jelöl a pixelnek a szomszédaihoz való viszonya szerint. A legegyszerűbb variációban szürkességi értékekkel dolgozik, és a pixel értékével küszöböli a közvetlen szomszédait. Az eredményt mint bináris számot reprezentálja. Az LBP fontos tulajdonsága, hogy implicit módon invariáns a szürkességiértékben bekövetkező monoton változásokra. Könnyen alkotható forgatásra invariáns változat is ebből (biteltolással megkeressük a legkisebb előállítható számot), és számtalan variációja is létezik (pl. adott rádiusszal, vagy csigavonalban vizsgálva a pixeleket [44]). Az így kapott bináris számokból hisztogramokat szoktak építeni, így megkaphatók egy adott képben vagy képdarabban a pixelekre vonatkozó lokális viszonyok [38,43,44].

5.2.2. GLCM

Az úgynevezett másodrendű statisztikai módszerek közül (ezek pixelpárokra vonatkozó információkat vizsgálnak, [41]) talán a leggyakrabban használt módszer. Haralick cikke szerint a képben a textúra tulajdonságai jórészt leírhatók egy készletnyi olyan mátrixszal, amelyek szürkességi értékek együttes előfordulását írják le (Gray Level Cooccurrence Matrix, GLCM). Az általa reprezentált értékek segítségével számtalan statisztikai jellemző számolható: Haralick cikke 14-et említ (pl. kontraszt, korreláció, variancia, inverz differencia-nyomaték, entrópia stb.). A GLCM mátrix forgatásra invariáns [42].

5.3. Következtetések, értékelés

A szuperpixel-generáláshoz vizsgálat algoritmusok a mi képeinkre meglehetősen hasonló outputot generál jelentős hibák nélkül (SLIC output: 20. ábra). Az algoritmusok bonyolultsága azonban egyértelműen eldönti a kettő közötti választás kérdését. [36] szerint azonban a SIFT mintegy 10-szer gyorsabb a Turbopixels-nél és 500-szor a Normalized Cuts-nál, és ezt az információt még ha némi fenntartással kezeljük is, akkor is igaz, hogy a Turbopixels képletei is bonyolultabbak, és ráadásul a megállási feltétele miatt minden lépésben szkeletont számít, ehhez képest a SIFT számítása egyszerű és gyors. A SIFT-hez a VLFeat nevű Matlab és C nyelvű függvénykönyvtárban [40] egy könnyen használható implementáció van.

A textúravizsgálathoz az LBP eredményét találtuk elég jónak, ezért az általunk tervezett textúradetektáló algoritmust erre építettük. A későbbiekben a fejlesztés során a GLCM beépítését is tervezzük. A jelenlegi működés leírása és részletes indoklás az algoritmust bemutató fejezetben olvasható.

6. Célpontosítás

A megismert hasonló rendszerek alapján és a levont következtetések után kialakult a megvalósítandó rendszer működése. A rendszernek DICOM szabvány szerint kell működni, vagyis az adatbázisnak tudnia kell fogadni a DICOM fájlokat és képesnek kell lennie a lekérdezésekre választ adni. A PACS szerver felé irányuló kommunikációknak is a szabványban leírtak alapján kell működni. Azért ez a megoldás a legjobb a projekt szempontjából, mert ennek köszönhetően más gyártó által készített eszközök is tudnak kapcsolódni és sikeresen kommunikálni a rendszerrel. A megvalósításra a DICOM is EASY sdk kliens oldali része és a dcm4chee PACS szoftver csomag kerül alkalmazásra. A rendszerben lévő üzleti logika annyiban változik meg ennek hatására, hogy a kép kijelölése és az annotáció kitöltése után a rendszernek el kell végeznie a szabványosítást, valamint a lekérdezéseknél ugyanezt a folyamatot kell végrehajtani fordítva. Az annotáció elkészítése során az 6. táblázatban bemutatott adatokat kell tudnia tárolni a rendszernek.

Kategória	Adat	Kategória	Adat
Szerzői adatok		Festési adatok	
	Szerző neve		Típusa
	DOI Digitális objektumazonosító		Festő neve
	Státusz		Festés dátuma
	Intézmény neve		Festés helye
	Intézmény címe		Felhasznált anyagok
Készítési adatok			Festési idő
	Feltöltés dátuma		Koncentráció
	Hardver neve		Tömeg
	Készítési protokoll		Rögzítő anyag
Kép adatok		Szöveti adatok	
	Kép szélesség		Szövet név
	Képfelbontás		Szerv
	Kép magasság	Páciens adatok	
	Szintér		Páciens születési dátuma
	Nagyítási szint		Páciens neve

	Kép megjegyzések	Páciens neve
	Alfa csatorna	Páciensazonosító
Szoftver adatok		Páciens kora
	Alkalmazás neve	Páciens súlya
	Alkalmazás verziója	Allergiák
	Alkalmazás készítője	Dohányos státusz
Orvosi adatok		Páciens kórtörténet
	Kategória	Etnikai csoport
	Kép típusa	Terhességi státusz
	Kép leírása	Megvizsgált testrész
	Annotáció	Kezelési adatok
	Betegség	Leírás
	Betegség mértéke	Megjegyzés adatok
	Diagnózis leírása	Megjegyzés 1
	Ki adta a diagnózist	Megjegyzés 2
	Hol adta a diagnózist	
Feldolgozási megjegyzések		
	Alfa csatorna	

6. táblázat: Bemutatja a rendszerben tárolt adatokat.

A tárolt adatok nem csak diagnosztikai célokat szolgálnak. Számos adat önmagában nem hordoz hasznos információt az orvosok számára, de más adatokkal együtt véve hasznos statisztika készíthető.

A tárolandó adatok 11 különböző csoportba sorolhatóak, ezek a következők:

1. Szerzői adatok
2. Készítési adatok
3. Szoftver adatok
4. Festési adatok
5. Kép adatok
6. Szöveti adatok
7. Orvosi adatok
8. Páciens adatok
9. Kezelési adatok
10. Feldolgozási megjegyzések adatai
11. Megjegyzés adatok

Szerzői adatok

A szerzői adatok a képet feltöltő személyről adnak információt. A szerző nevén, intézményének nevén és intézményének címén kívül két fontos mezőt tárolunk, mégpedig a DOI-t és a státuszt. A DOI vagyis digitális objektumazonosító segítségével szintén egyértelműen azonosítható a feltöltött tartalom. A státusz egyfajta kategória besorolásra alkalmas, ezt a mezőt egy későbbi feldolgozási fázis tölti fel, a feltöltött tartalom ellenőrzése után. A lehetséges státuszok a következők lehetnek: „ellenőrzött” és „nem ellenőrzött”.

Készítési adatok

A készítési adatok az annotáció feltöltésének idejét, a készítés során használt hardver nevét és a készítési protokollt tartalmazza.

Szoftver adatok

A szoftver adatok annak az alkalmazásnak az adatai, amivel elkészítették az annotációt és feltöltötték a képi adatbázisba. A szoftver nevét, verzióját és a fejlesztőjének a nevét tároljuk.

Festési adatok

A festési adatok a feltöltendő kép festési eljárásáról és a festő nevééről szolgáltat információt. A festési eljárás főbb lépéseinek az adatai kerülnek tárolásra, a festés neve, a festés típusa, a festés során felhasznált anyagok stb.

Kép adatok

A képadatok a feltöltött képről adnak információt. Tárolásra kerül a kép szélessége, magassága, felbontása, színtere, nagyítási szintje és az alfa csatorna használata. Ezeket az adatokat a textúra alapú keresések során az algoritmusok használják fel.

Szöveti adatok

A szöveti adatok a szövetminta származásáról szolgáltatnak információt. Ez alapján eldönthető melyik szervből nyerték ki a mintát.

Orvosi adatok

Az orvosi adatok a mintához tartozó diagnosztikai célú információkat tartalmaznak. Tárolásra kerül a betegség kategóriája, neve, mértéke, a diagnózis leírása, ki és hol adta a diagnózist. Továbbá az adatbázis tartalmazza a kép típusát (MRI, röntgen), és egy annotációt a képről.

Páciens adatok

A páciens adatok a betegről szolgáltatnak információt. Azonban az adatvédelmi törvénynek megfelelően az ebben a kategóriában tárolt adatok nagy része nem lehet valós, ezzel az adatok nyilvánosságra kerülése esetén is biztosítva van a beteg anonimitása. A páciens neve semmiképpen sem lehet kitöltve, helyette egy azonosító kód kerül eltárolásra. Az azonosító tárolása esetén a többi adat valós adatokkal feltölthető. A későbbiekben a páciens betegsége, életkora, neme, etnikai hovatartozása, dohányzási státusza, terhességi státusza stb. alapján bizonyos összefüggések kereshetők, amik elősegíthetik a betegségek kialakulását és megértését.

Kezelési adatok

A kezelési adatok egy leírás a páciens utólagos kezeléséről, ami tartalmazza az alkalmazott kezelés nevét, típusát, idejét és az elért eredményt.

Feldolgozási megjegyzések adatai

A feldolgozási megjegyzések adatai között csak az alfa csatorna szerepel. A kép alfacsatornájának használatát írja le. A felhasználói felületen kijelölt poligonokat el kell tudni tárolni képekben. A képek általában vagy négyzet, vagy téglalap alakúak. Valahogy jelezni kell a feldolgozó algoritmusoknak, hogy a négyzet alapú képen mi a háttér és mi az előtér, vagyis az eltárolandó poligon.

Megjegyzés adatok

A megjegyzés adatok kitöltése opcionális, ezek az adatok olyan plusz információk lehetnek, amelyek nem szerepelnek a felsorolt listában.

Az F02. számú függelékben megtekinthető a DICOM szabvány összes felhasznált tagja a kiegészítő paraméterekkel együtt.

7. A rendszerrel szemben támasztott követelmények

A rendszerrel szemben támasztott követelmények a következőkre vonatkoznak:

- a) GUI
- b) Kliens
- c) Szerver
- d) Hálózati kommunikáció
- e) Az egész rendszer

A GUI-nak lehetővé kell tennie a felhasználó számára azt, hogy a textúra-adatbázisba adatot tudjon feltölteni. Ennek keretében a felhasználó ki tudjon jelölni egy szövegrészt egyedi színnel, és a kijelölés elrontása esetén lennie kell módosítási lehetőségnek. Egy képre tetszőleges mennyiségű kijelölést lehessen felvinni, melyek külön-külön módosíthatóak. További elvárás a felhasználói felületről, hogy legyen lehetőség szövegtípusok felvitelére, beolvasására külső forrásból. A külső forrás lehet adatbázis vagy egy fájl is. Lehessen a felvett típusokból osztálypéldányokat létrehozni. Az így létrejött osztálypéldányokhoz korlátlan mennyiségű minta kijelölésére is legyen mód a felhasználó számára egy jól áttekinthető, ízléses felületről. A kijelöléshez lehessen annotációt fűzni. Az annotáció kitöltésének elősegítése érdekében a GUI-t úgy kell elkészíteni, hogy a felhasználónak minél kevesebb adatot kelljen kézzel kitöltenie, inkább csak kiválasztania kelljen a tartalmakat egy előre meghatározott listából. Ezek után legyen lehetőség a kijelölésekből származó kis képek és annotációk együttes mentésére vagy a lokális, vagy a globális szerverre. Továbbá a felhasználói felületnek keresési lehetőséget is biztosítania kell a felhasználó számára. Ezen felül különböző halmazműveletek elvégzését is elő kell segítenie a slideon.

A kliensalkalmazás üzleti logikájában kötelezően implementálva kell lennie a keresési, annotálási, kommunikációs részeknek. Az alkalmazásnak a felhasználói felületről begyűjtött adatokat ellenőriznie kell és a megfelelés esetén a szükséges műveletet elvégeznie. A keresések esetén az adatbázisból kinyert adatokat fel kell dolgoznia és a GUI számára átadni. Az alkalmazásnak az annotációból származó adatokat DICOM szabványú „.dcm” kiterjesztésű fájlban kell tudnia egyesíteni, majd a szabványnak megfelelő módon továbbítani a PACS szerver felé. Az elküldött fájl megérkezését nyugtázni és a felhasználó tudomására adni a GUI-n keresztül.

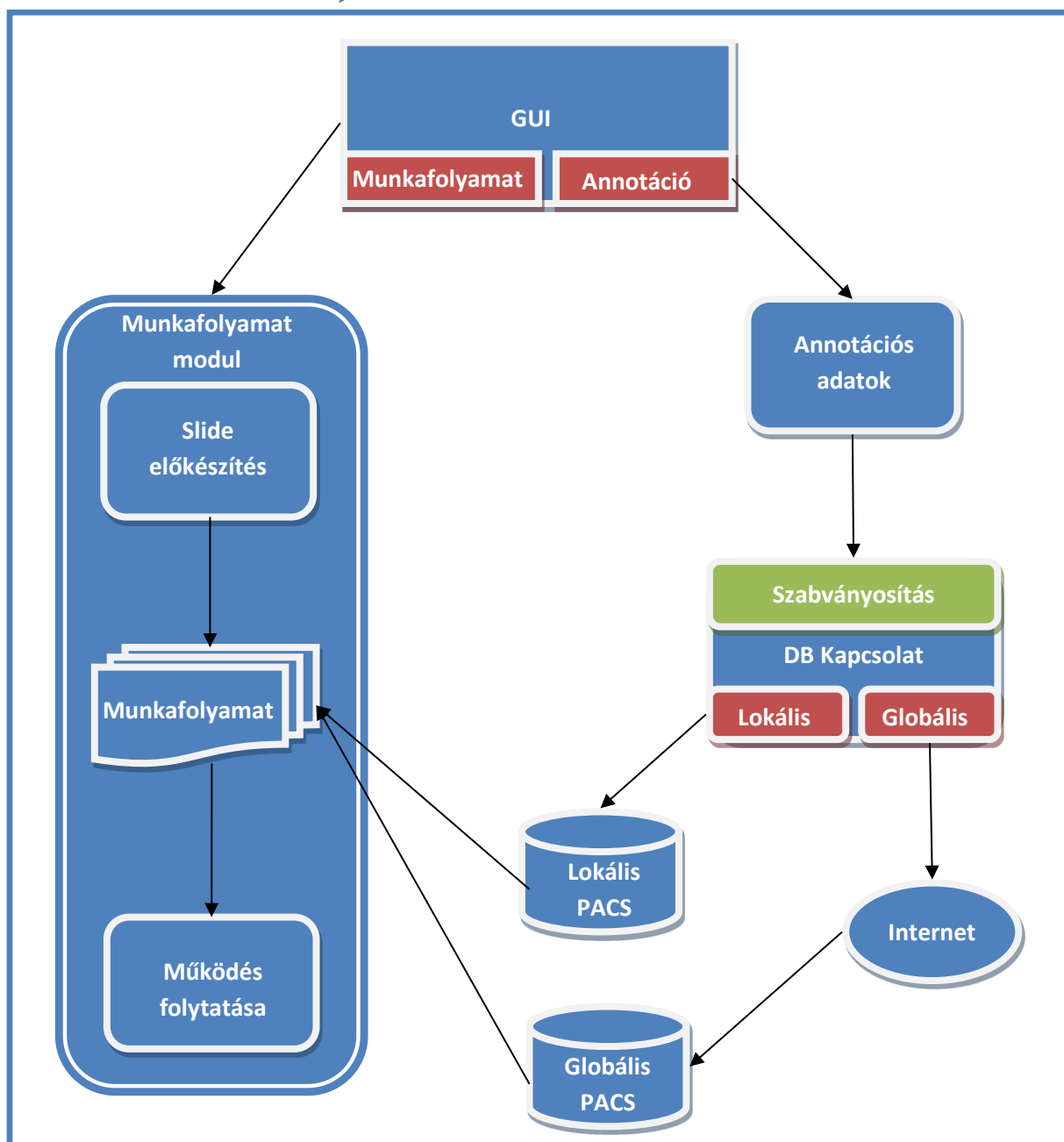
A szerver feladatai közé tartozik a hozzá érkező lekérdezések kiszolgálása, mindezt úgy, hogy szabványos módon kommunikáljon az őt megszólító alkalmazásokkal. Továbbá a hozzá küldött DICOM fájlokat tudja kezelni és a megfelelő adatbázis-struktúrába beilleszteni a későbbi lekérdezhetőség érdekében. Feladatai közé tartozik még a beérkező lekérdezések kiszolgálása, ezt is szabványos módon kell végrehajtania. Továbbá biztosítania kell valamilyen lehetőséget az adatbázis karbantartására, vagy a nem megfelelő adatok eltávolítására.

A hálózati kommunikációval szemben támasztott követelmény az, hogy a kliens és a PACS szerver vagy a PACS szerver és egyéb kliens alkalmazás között a kommunikációt a DICOM szabványban leírtak alapján valósuljanak meg. Ez a szabványos kommunikáció mind a lekérdezésekre, mind a feltöltésekre vonatkozik. Mivel a DICOM szabványt eredetileg a lokális hálózatokra tervezték, ezért

nincs benne a szabványban, hogy valamilyen titkosított csatornán kell kommunikálni az eszközöknek. A mi rendszerünknek azonban az interneten keresztül kell a PACS szerverhez kapcsolódnia, ezért VPN segítségével kell a kapcsolatot megteremteni. Az egész rendszernek olyan felépítésűnek kell lennie, hogy lehetőség legyen a DICOM szabvány követésére, vagyis a szervertől adatbázisnak, a kliensoldali GUI-nak és a kialakított kommunikációs csatornának könnyen bővíthetőnek és átalakíthatónak kell lennie. Ha lehetőség van rá, a tervezési fázisban az előre látható fejlődési irányt figyelembe kell venni.

8. A rendszer felépítése

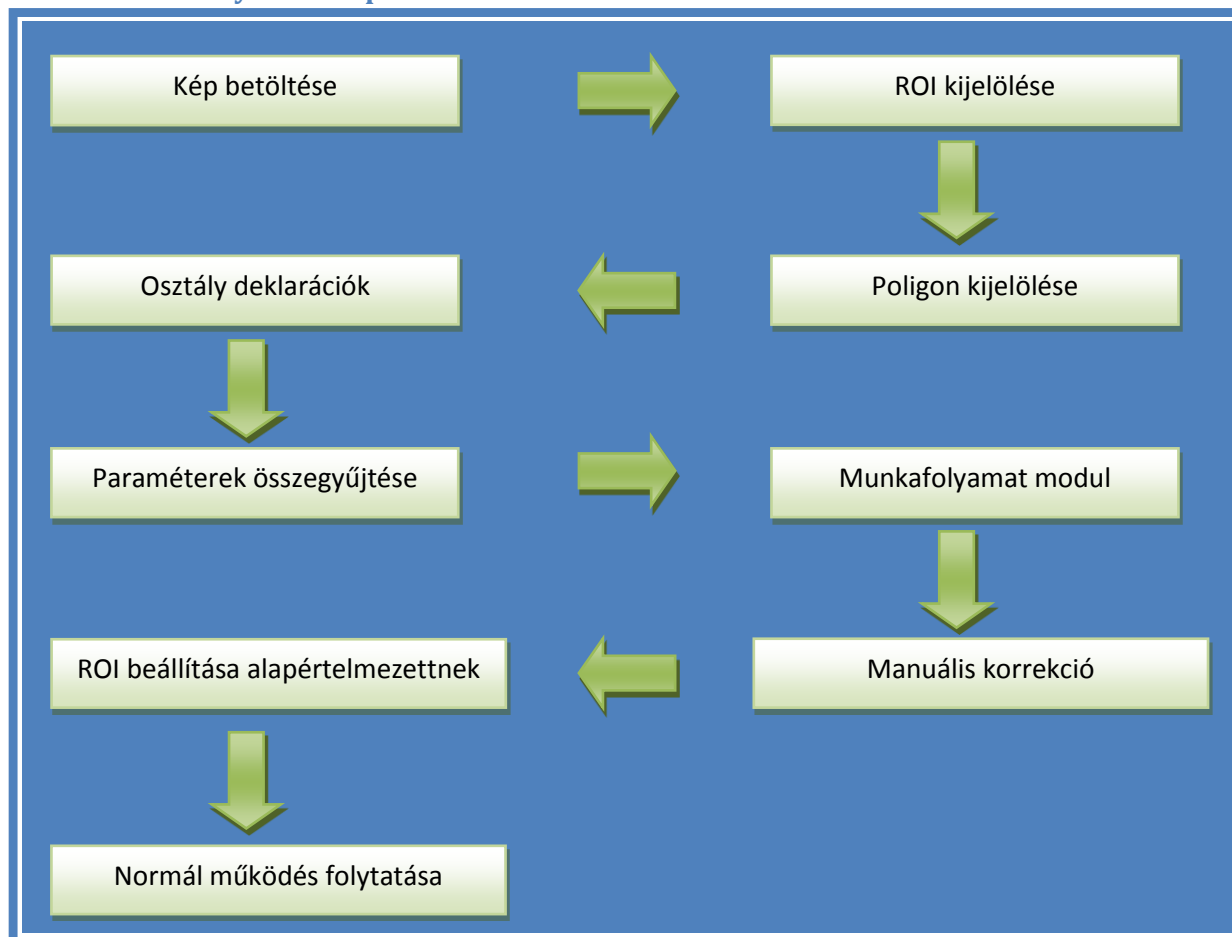
8.1. A rendszer moduljai



11. ábra: A rendszer általános felépítése.

A teljes rendszer általános felépítéséből (11. ábra) fakadóan a felhasználó alapvetően két jól elkülönülő feladatra használhatja a rendszert. Az egyik lehetséges működési lehetőség, amikor a felhasználó egy adott munkafolyamatot hajt végre. Egy ilyen munkafolyamat több algoritmus előre definiált, megfelelő célú egymás után kapcsolását, láncolását jelenti. A másik lehetséges működési mód az annotációkészítés, amely során a felhasználó szöveti mintákat annotálhat és tölthet fel az adatbázisok valamelyikébe.

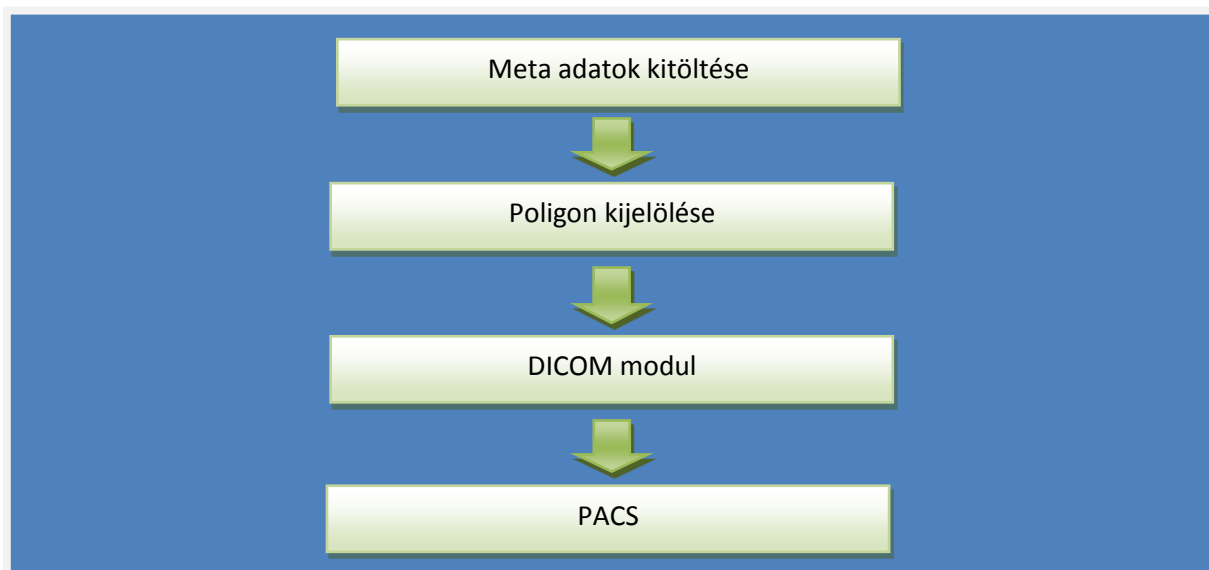
8.1.1. Munkafolyamat alapú működés



12. ábra: A munkafolyamat alapú működés.

A munkafolyamat végrehajtása során (12. ábra) a felhasználó betölt egy metszeti mintát a GUI-n. Beállítja a ROI-t, vagyis azt a részt, amit feldolgoztatni szeretne a munkafolyamattal. A felhasználói felületen kijelölhet bizonyos poligonokat, ezekhez a poligonokhoz hozzáfűzhet osztálydeklarációkat, amik majd segítenek az algoritmusoknak a feldolgozási folyamat során. Ezek után egy algoritmus összegyűjti a megadott poligonok és osztályok adatait, majd elindít egy új munkafolyamatot, aminek átadja az összeszedett adatokat. A munkafolyamat működését a felhasználó befolyásolhatja. Amennyiben valamelyik textúra alapú keresési munkafolyamatot választotta a felhasználó, akkor az algoritmusok a lokális vagy a globális PACS-ból lekérhetnek eltárolt adatokat a működésükhöz. A munkafolyamat végeztével a megkapott eredményt a rendszer a GUI-n megjeleníti. Ekkor lehetőséget kap a felhasználó az eredmények manuális korrekciójára, amely segítségével a kereső algoritmusok esetleges hibáit javíthatja. A szegmentált részt végül a felhasználó beállíthatja alapértelmezett ROI-nak, amelyen újabb feladatokat végezhet el vagy elvetheti az eredményeket és folytathatja a munkamenetet az előző állapotról.

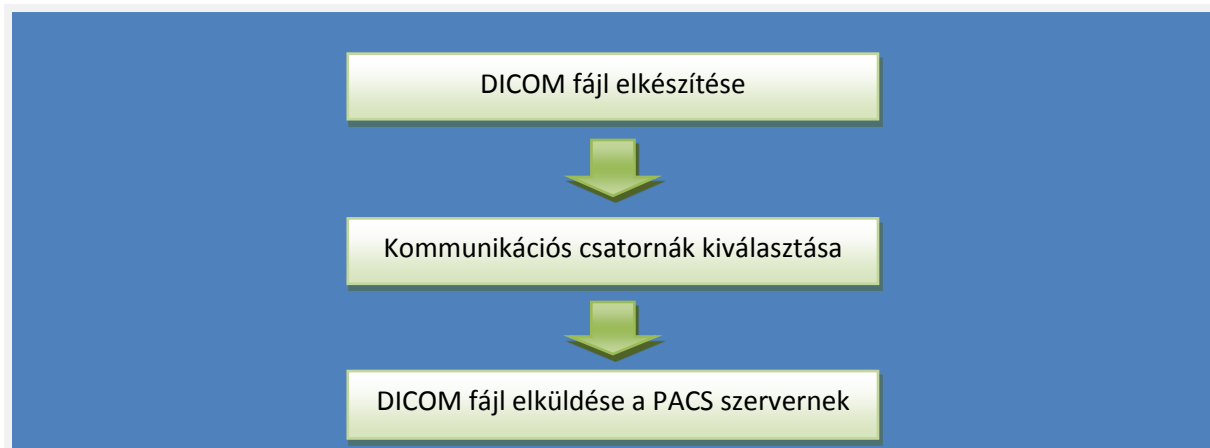
8.1.2. Annotáció



13. ábra: Az annotáció folyamata.

Az annotáció során (13. ábra) a felhasználó, aki tipikusan egy orvos, betölthet a GUI segítségével egy vagy több digitális mintát, majd ezen kijelölve egy ROI-t elkezdheti az annotációs adatok összegyűjtését. Ehhez kijelöli a tárolni kívánt szövetdarabot, utána az adatmezőkben megadja az adott mintához tartozó adatokat, amelyek a Célpontosítás, konkretizálás szekcióban bemutatott 11 kategóriába sorolhatóak (Szerzői adatok, Készítési adatok, Szoftver adatok, Festési adatok, Kép adatok, Szöveti adatok, Orvosi adatok, Páciens adatok, Kezelési adatok, Feldolgozási megjegyzések adatai, Megjegyzés adatok). Számos adatot a rendszer kitölt előre, például a kijelölt képek paramétereit. Az adatmegadás után a rendszer elvégzi a szabványosítást, ennek során egy olyan fájlformátumot kapunk, amit bármelyik PACS rendszer képes fogadni. Az elkészült DICOM fájlt a felhasználó választása szerint a lokális vagy a globális PACS-ba küldi el a rendszer.

8.1.2.1. DICOM modul

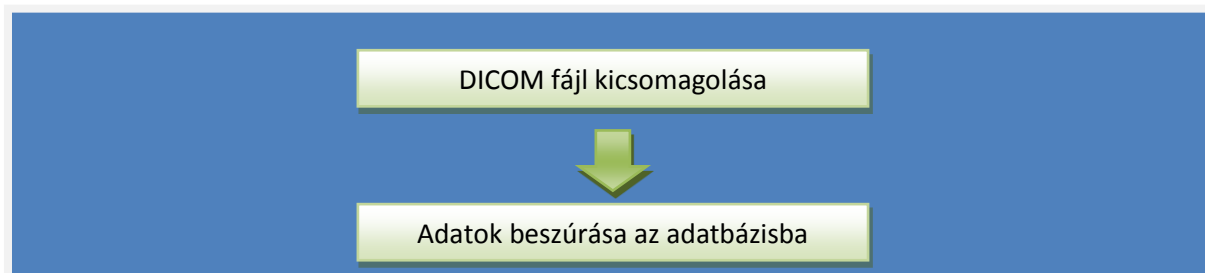


14. ábra: A DICOM modul folyamata.

A DICOM modul feladata a szabványosítás végrehajtása. A folyamata (14. ábra) első lépés a modulnak átadott poligon- és metaadatok alapján elkészíteni a DICOM fájlt. Az átadott adatok között szerepel a felhasználó által kiválasztott adatbázis neve is, ennek megfelelően történik meg a

kommunikációs csatorna kiválasztása. Ezek után a kiválasztott csatornán a modul kiküldi az elkészült fájlt a szervernek.

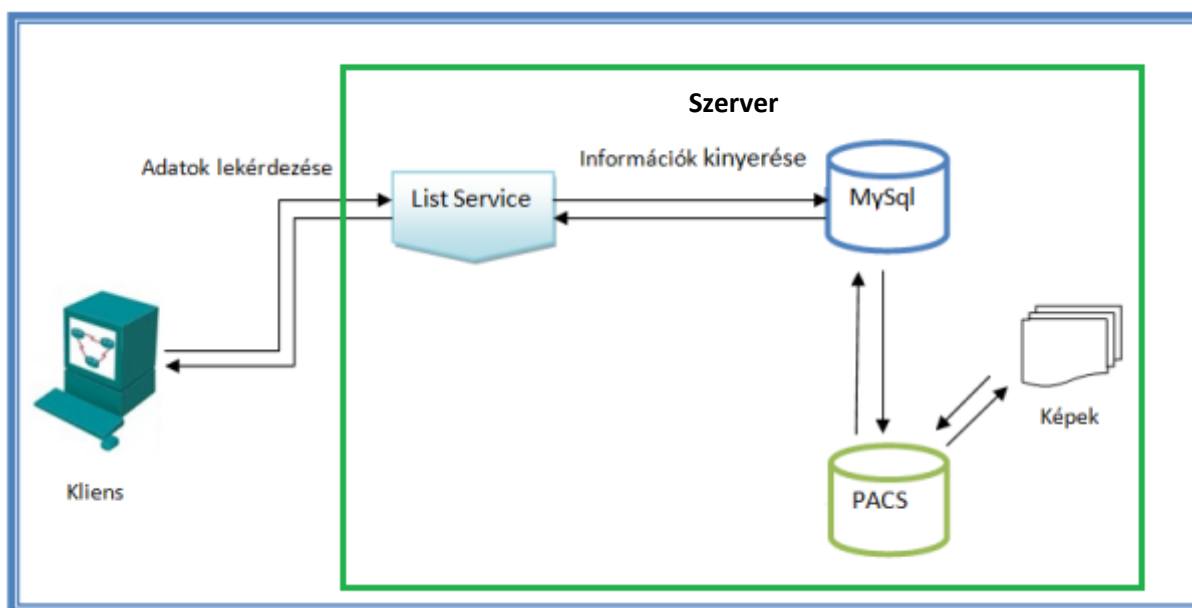
8.1.2.2. PACS modul



15. ábra: A PACS modul folyamata.

A PACS modul (15. ábra) feladata a megkapott DICOM fájlok eltárolása a későbbi felhasználhatóság érdekében. A modul a szervernek elküldött DICOM fájlt kicsomagolja, az így megkapott képet lementi a merevlemezre, a metaadatokból kinyert információkat pedig beszurja a MySQL adatbázis megfelelő tábláiba.

8.1.2.3. Lekérdezés



16. ábra: A lekérdezés modul felépítése.

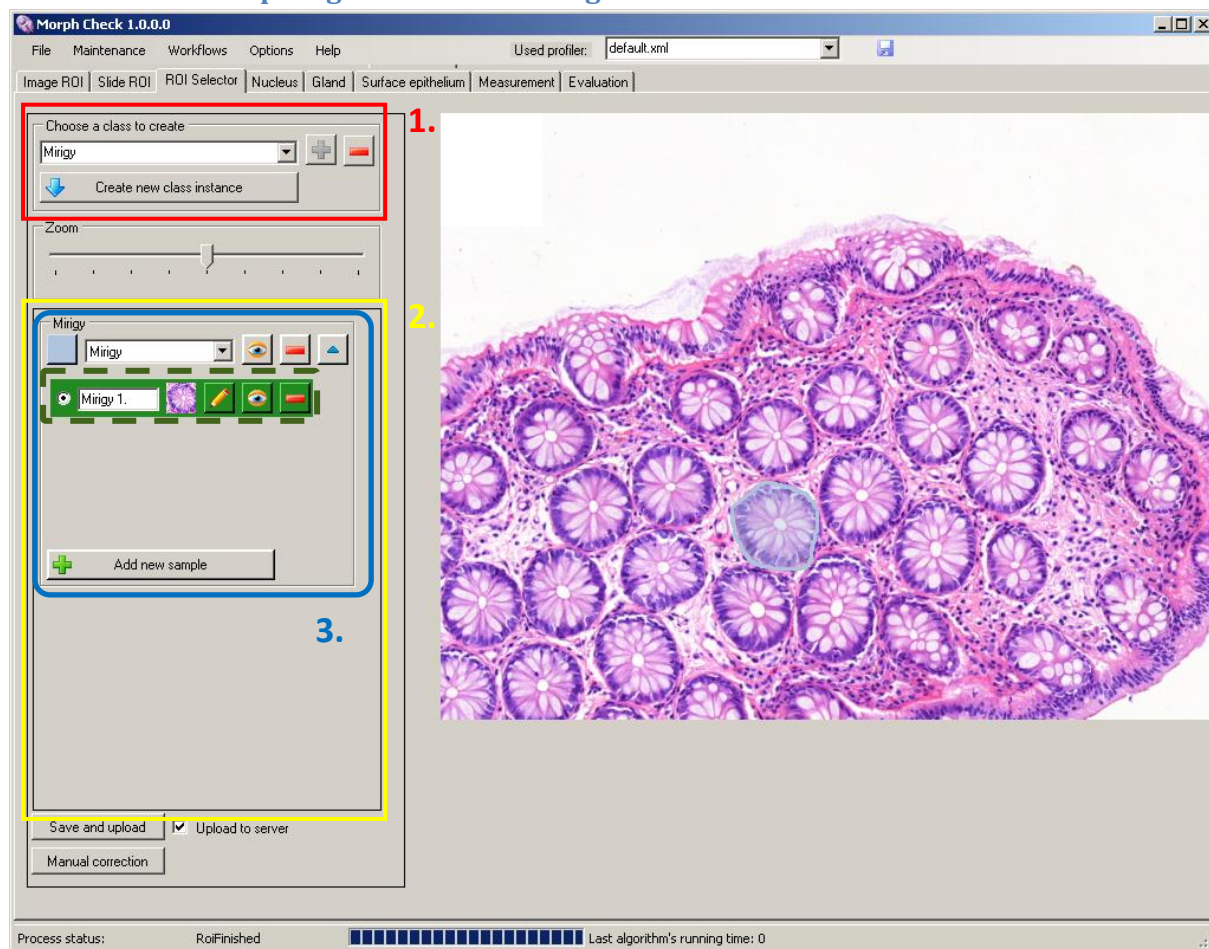
A lekérdező modul (16. ábra) a textúra alapú kereséseknél fontos. A PACS rendszer úgy működik, hogy alapesetben egy kórházi információs rendszerhez kapcsolódik, ahol az orvos a pácienshez tartozó egyetlen DICOM fájlt kérdezheti le, ebből kifolyólag a PACS-ból az adatokat nem lehet úgy kinyerni, mint egy relációs adatbázisból. A projekt szempontjából ez azért probléma, mert az eltárolt orvosi adatokból nem kérhető le például minden RGB színterű kép. A textúra alapú kereső algoritmusoknak pedig ilyen szűrési lehetőségekre szüksége lehet. Ennek a problémának a megoldására készült el a lekérdező modul. A modul a szolgáltatás orientált architektúrára (SOA) épül. A PACS szerver mellett fut egy szolgáltatás, ami JAVA alapú, aminek az előnye a több platformon való használhatóság. A klienssel csatlakozva a szolgáltatáshoz a szűrő feltételeknek megfelelő választ kapunk a szervertől, amelyek segítségével az adatbázisban eltárolt adatokból lehet kinyerni a

megfelelő fájlokat. A DICOM fájlokat ezután bármely más szabvány szerint működő kliensalkalmazás felhasználhatja.

8.2. Grafikus felület felépítése

A bemutatott működés megvalósításához a felhasználói felületet kibővítettük. Röviden bemutatjuk az itt történt változásokat.

8.2.1. Textúra alapú algoritmusokat támogató eszközök



17. ábra: A felhasználói felület.

A 17. ábra több kijelölés látható, melyek a felhasználói felület főbb részeit jelölik.

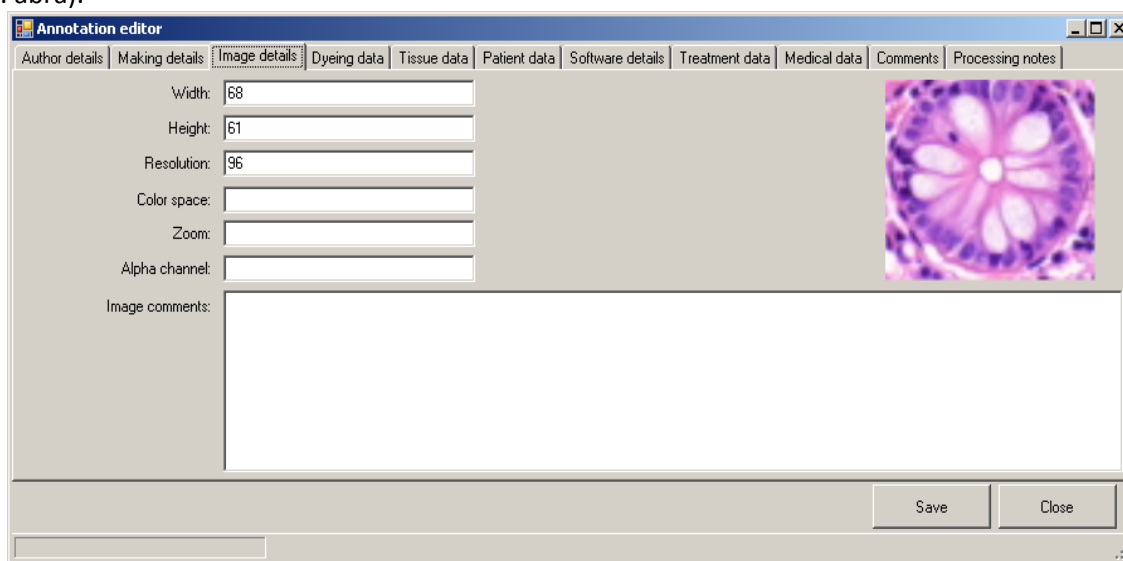
1. A szövettípus választó, amely egy külső forrásból betölti a meglevő szövettípusokat, ha vannak. A külső forrás lehet egy JSON [51] szerkezetű fájl vagy akár adatbázis is.
2. A kiválasztott szövettípusból létrejövő osztályok példányait látható itt. Minden osztály a létrejövetelekor zárt státuszban van. Az osztályok láthatóságát egyesével lehet állítani, illetve a szükségtelenek törölhetők.
3. Ha a felhasználó kinyit egy osztályt, akkor láthatóak az osztályon belüli minták és új felvételére is lehetőség nyílik.
4. A minták láthatósága külön ki/bekapcsolható, valamint egyesével adható meg a hozzájuk tartozó annotáció és egyesével törölhető a már hozzá rendelt annotáció.

A felhasználó számára egy képen lehetőség van kijelölni kézzel kis részeket, amelyeket el akar tárolni vagy küldeni későbbi feldolgozásra. Minden kijelölt alakzatot más rétegen (*Layer*) tárol a rendszer.

A szövettípus választó segítségével választhatja ki a felhasználó, hogy milyen típusú osztályt szeretne felvenni. A típusát egy osztálynak a későbbiekben is lehet módosítani, így nem okoz problémát, ha elsőre rosszat választ a felhasználó. A lenyíló lista feltöltése automatikusan megtörténik, hogy ne a felhasználónak kelljen kitalálnia az értékeket. Ezeket az információkat egy JSON fájlban tárolja a rendszer. A szövettípus választó felületen ezen kívül lehet új típust felvenni illetve meglévőt törölni. A felület úgy van megtervezve, hogy figyelje, ha ugyanazt a típust szeretné a felhasználó még egyszer felvenni. Ilyenkor letiltódik az új osztálytípust felvevő gomb. Ha törölni szeretne a felhasználó egy típust, akkor az törli az összes felvett osztálypéldányt, ami ilyen típusú. Erre egy felugró ablak is figyelmeztet, a véletlen mellékattintások okozta kellemetlenség elkerülése végett.

8.2.2. Annotálást és adatbázissal való kommunikációt támogató eszközök

Ezen a felületen van lehetőség az egyes mintákhoz annotációt felvenni. Egy nagyon egyszerű felhasználói felületen lehet megadni szükséges információkat, majd a végén elmenteni azokat (lásd 18. ábra).



18. ábra: Annotációs adatok szerkesztő felülete.

A mezők jelentős részét a felhasználónak kell kitöltenie, de e mellett vannak olyan mezők, amelyek automatikusan töltődnek ki. Ilyenek például a kép dimenziói és felbontása.

8.2.3. Manuális korrekció

A digitális képeken való keresés nem mindig ad jó eredményeket, mivel ez nem egy triviális feladat. A rendszer tartalmaz egy manuális korrekció funkciót, amely az egyes keresések után végezhető. A célja az, hogy a felhasználónak lehetősége van módosítani a kapott eredményt. Erre azért van szükség, mert például egy textúra alapú keresésnél rosszul talál meg egy adott mintázatot és ezután van lehetőség az eredmény módosítására manuálisan. A megfelelő működéshez szükség van egy eredeti és egy eredményképre, valamint egy listára az eredményképen megtalálható osztálytípusokról és azok jelölő színéről. A felület lehetőséget biztosít a felhasználó számára poligonos és egyszerű pont kijelölésre. Ezekkel az eszközökkel lehet a különböző osztályokba besorolni vagy törölni az adott képpontot vagy képponthalmazt. A törlés során osztály nélkülivé állítja a rendszer a kijelölt területet.

8.3. Textúra alapú kereső algoritmusok

8.3.1. Feature Extraction algoritmus

Ez a 3DHistech által fejlesztett textúra alapú kereső algoritmus. Lévé, hogy a 3DHistech egy piacvezető cég az egészségügyi szektorban, ezért az általuk implementált eljárás működése ipari titok, a csapat csak tesztelésre kapta meg egy DLL formájában.

Az algoritmus a wavelet transzformációra épül, ami jelfeldolgozáson alapuló, textúra analízisre szolgáló módszerek egyike (számtalan más alkalmazási területe is van). Waveletek segítségével a jelekben előforduló szabálytalanságok vagy hirtelen események jobban leírhatók, mint egyszerű szinuszoidokkal, emiatt jól alkalmazható természetes textúrák esetében [53]. Sajnos nincs ismeretünk a 3DHistech által alkalmazott wavelet implementáció konkrét részleteiről.

Az algoritmusnak a kiválasztott képen definiált szövetosztályokat kell bemenetként átadni. Ezen kívül egyetlen paraméter van, a wavelet transzformáció szintjeinek a száma. Ez az érték 1 és 10 között lehet. Jelentése: hány darab kicsinyítésen essen át a kép, mindegyiken wavelet együtthatókat mérve, melyek a jellemző-tér dimenzióját növelik. Ilyen módon a különböző zoomszinteken látható texturális információkat is beleveszi a tudás kialakításába. 1-es értéknél csak az aktuális nagyításra jellemző textúrát veszi figyelembe. 1-nél nagyobb érték átadása esetén biztosítani kell, hogy a bemeneti kép méreteinek a 2 alapú logaritmus a szintnél nagyobb vagy egyenlő legyen.

Értékelés

Az algoritmus egy szálon működik mindössze, ennek ellenére igen gyors, és bizonyos eredményei jónak is tűnnek (háttérelválasztás). Tesztelésének módszereiről és eredményeiről részletes információt közlünk a tesztelésről szóló fejezetben.

8.3.2. LBP alapú algoritmus

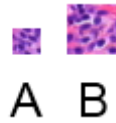
Az LBP alapú algoritmus általunk tervezett és elkészített módszer, így a működését részletesen ismerteti a dolgozat.

A feladata a szöveti képeken látható természetes textúrák elválasztása, illetve osztályokba való klasszifikálása. Hasonló problémákkal régóta foglalkoznak a képfeldolgozás területén, de az aktuális feladatban több nehezen leküzdhető nehézség van, amelyeket érdemes körvonalazni.

Egyrészt, az általános textúraszegmentáló módszerek többsége egyszerű mesterséges textúrákon mutat jó eredményt. Ez nyilván egyben esélyt ad arra, hogy bonyolult természetes textúrákon is jól működjenek ezek a módszerek, a gyakorlatban mégis azt tapasztaljuk, hogy az általuk mutatott eredmények nem egyeznek az elvárttal. Másrészt, a konkrét feladatban olyan textúrákkal dolgozunk, amelyekre a következők is igazak:

- Az általunk vizsgált bonyolult textúrák egy részénél az emberi szemlélő is az előzetes ismeretei alapján vagy bizonytalanul osztályoz, nagyon erősen felhasználva előzetes ismereteket (két hasonló formájú „darab” pl. elhelyezkedéstől függően hámszövet egy darabja, vagy kötőszövet egy darabja lehet – az 19. ábra az A jelű kép egy hámszövet-darab, a B jelű pedig egy kötőszövet-darab. Ezek az eredeti képen a környezet alapján az ember számára egyszerűen felismerhetők, de a szövet természetes textúrájában

számtalan ilyen igen hasonló darab előfordul). Ezenél csak a textúradarab környezete vagy elhelyezkedése lehet támaszkodási pont, mert önmagukban gyakorlatilag nem megkülönböztethetők.



19. ábra: A: hámszövet-darab, B: kötőszövet-darab.

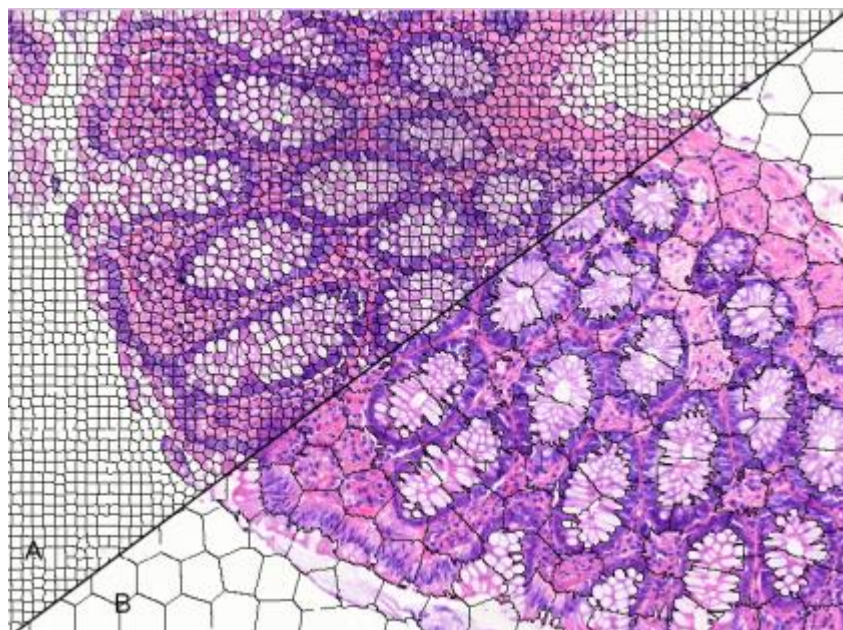
- Maga a hasonlóságfogalom is nehezen definiálható: hogy a felhasználó számára a „hasonlóság” azt jelenti-e, hogy „olyan kötőszövet-részek, amelyek nem véresek”, vagy azt, hogy „olyan kötőszövet-részek, ahol a sejtmagok hasonló sűrűségben helyezkednek el, még akkor is, ha véresek”, kizárólag az ő aktuális gondolkodása dönti el.

Mindkét probléma megoldható, vagy a megoldás közelíthető a felhasználó előzetes ismereteinek bevonásával (például úgy, ha tudjuk/megadhatjuk, hogy hámszövet csak a metszet szélén lehet, vagy milyen textúrák (szövegtípusok) milyen más textúrák mellett helyezkedhetnek el, vagy úgy, hogy a felhasználóval megadhatjuk, hogy milyen tulajdonság szerinti hasonlóságot szeretne). Ilyen szabályok könnyen definiálhatók. Az algoritmus fejlesztése során azonban a célunk egyelőre egy minél jobb módszer fejlesztése a fentiek nélkül. Később, ha már a fejlesztés késznek mondható, akkor további pontosítás érdekében az említettek közül implementálhatjuk ezeket a segédfunkciókat is.

Algoritmusleírás

Az algoritmus inputja a vizsgálandó terület, valamint az azonosítandó osztályokhoz tartozó képek (egy-egy osztályhoz több kép is tartozhat).

A vizsgálandó területet először „textúradarabokra” bontjuk a SLIC algoritmus segítségével. A SLIC-ben két paramétert kell megadni: egyrészt, hogy mekkora szuperpixelet szeretne látni az ember a felbontás eredményeként (regionsize paraméter (20. ábra)), másrészt, hogy mennyire törekszik arra, hogy a szuperpixelet egységes formájúak legyenek (regularizer paraméter). Utóbbi paraméter alacsonyra választásakor homogén színű, de véletlenszerű formájú és méretű, nem kompakt, magasra választásakor pedig esetleg inhomogénebb, de a régióméret és kompaktság megtartására törekvő szuperpixelet kapunk.



20. ábra: A SLIC algoritmus eredménye kétféle paraméterkészlettel. A: regionsize: 10, regularizer: 0.8. B: regionsize: 40, regularizer: 0.6 .

A paraméterek megválasztására a szempontok a következők voltak:

- Esetünkben a szín szerinti homogeneitás nyilván nemcsak nem fontos, de egy teljesen szín szerinti bontás nem is lenne használható (hiszen a vizsgálandó textúráink többszínűek). 0.8-as regularizer paramétert választottunk (0-1-es skálán), amely még biztosítja, hogy a superpixelek a fontos határokhoz igazodjanak, de már meglehetősen kompakt superpixeleket generál.
- Nagyobb méretű superpixelek kívánatosabbak lennének számunkra, hiszen ezekben nyilván pontosabb statisztikák lennének számíthatók. Viszont ha túl nagyra választjuk a superpixelek méretét, akkor a rácspontok véletlenszerű elhelyezkedéséből adódóan sok superpixelbe több textúrából fognak részek esni, ami a textúra vizsgálatát lehetetlenné teszi. Emiatt viszonylag kicsi, 10-es superpixel-méretet választottunk, ami kb. 10x10-es régiókra való törekvést jelent.

Ezek után minden superpixelt egyenként vizsgálunk osztályhoz való tartozás szempontjából. Mivel a 10x10-es régiók túl kicsik ehhez a vizsgálathoz, minden superpixelt a szomszédjaival együtt vizsgálunk, így egy átfedő elrendezést alkotva a képen. A vizsgált területek ezek szerint hozzávetőleg 30x30-asak, ami kb. kilencszer akkora vizsgálható területet jelent. Az átfedés jelentősége azért is nagy, mert lehetővé teszi a környezet információinak beszámítását a superpixelek besorolásakor.

Az osztályhoz tartozás vizsgálata egyelőre két paraméter segítségével történik: egyszerű hisztogram és egy Local Binary Pattern-hisztogram (forgatásra invariáns változat) alapján. Mindkét paraméter segítségével távolságot számítunk minden mintaképhez. (A hisztogram esetében az Earth Mover's Distance nevű metrikával számolunk különbséget. LBP-hisztogramnál az oszlopok egymásmellettiisége nem jelent hasonlóságot, ezért egyszerűen a két hisztogram különbségének az abszolút értékével számolunk.) A konkrét besorolást majd a pontosabb LBP-alapú osztályozás adja, a hisztogramot csak ennek megtámogatására használjuk, a következőképpen:

- Minden vizsgált superpixel esetében az összes mintaképhez tároljuk a (hisztogram szerinti) távolságot, majd ezeket növekvő sorba rendezzük, és a superpixelét abba az osztályba soroljuk, amelyikből először találjuk meg a *második* hivatkozást a sorrendezett távolságok között. Ha a superpixel egyértelműen az adott osztályba sorolható, akkor ugyanis a két legkisebb távolság úgyis ehhez az osztályhoz számolódik. Ha viszont kétséges a besorolása, akkor (mivel hasonlóság „véletlenül” is előfordulhat egy-egy kép esetén) azt szeretnénk jutalmazni, ha több képhez is hasonló egy adott osztályból. Ilyenkor több alacsony távolság is számolódik ehhez az osztályhoz, amik a lista elejére kerülnek.
- Végül, miután a superpixelhez megvan, hogy melyik osztályba tartozik, a szomszédaihoz is bejegyezzük ugyanazt az osztályt. Így minden superpixelhez az átfedések miatt több besorolás áll elő, és a végső besorolás az az osztály lesz, amelybe a legtöbbször besoroltuk a superpixelét.
- Ezzel a művelettel előállt, hogy a superpixelét a hisztogramja szerint melyik osztályba sorolnánk. A szomszédokhoz való bejegyzés oka, hogy a nagyobb homogén textúrarészek a legérdekesebbek számunkra, és ez a módszer bünteti a magányos, „véletlenül” előforduló besorolásokat, és jutalmazza azokat, amelyek környezetében több hasonló besorolás van.

LBP alapján minden superpixelhez egy darab besorolást számítunk csak, ezután pedig összevetjük a superpixelekre kapott besorolásokat. Amelyik superpixelét hisztogram és LBP alapján is azonos osztályba soroltuk, azokat stabilan besorolt superpixelnek tekintjük, amelyeket pedig nem, ott nem tekintjük biztosnak az eredményt. Jelenleg a fals pozitív találatok elkerülése érdekében ezeket a superpixeleket besorolatlanak tekintjük, de később a „stabil” szomszédok bevonásával valamilyen besorolást rájuk is számolunk majd.

Értékelés

Az algoritmus jelenleg még fejlesztés alatt áll. A módszert a Matlab környezetben prototipizáltuk, és a kipróbáláshoz olyan .NET-es DLL formában építettük be a rendszerbe, amelyet Matlab alól fordítunk a kód minden lényeges változtatása után. Ezzel nem valós .NET kód áll elő, hanem egy matlabos futtatókörnyezet, a Matlab Compiler Runtime által lefuttatható modul.

Az algoritmus C# nyelvű portja is elkészült már, jelenleg kísérleti fázisban van. Ennek a hatékonyságára igen nagy hangsúlyt fektettünk a minél jobb sebesség elérése érdekében. Ebben a SLIC algoritmus megvalósítására a VLFeat library C nyelvű API-ját használtuk, amelyhez wrapper készült a C# kódban való felhasználhatóság érdekében. A feltétlenül létrejövő változók ésszerű mértékű újrafelhasználásával, előkalkulációkkal, részszámítások mentésével, keretrendszeri sajátosságok kihasználásával nagyon jelentős sebességnövekedés volt elérhető a Matlab verzióhoz képest (még az egyébként kiválóan párhuzamosítható algoritmus többszálúsítása nélkül is).

A Matlab modul rosszabb teljesítményének az oka két dologban keresendő: egyrészt, a Matlab kód implementációja nem ideális (Matlabban kerülni kell a ciklusokat, helyette a kód „vektorizálásával” mátrix és vektorműveletekre kell visszavezetni a ciklikusan leírható műveleteket – mivel a Matlabot csak ideiglenesen, prototipizálásra használtuk, nem fordítottunk arra különösebb figyelmet, hogy ez tényleg mindenhol így legyen), másrészt a futtatókörnyezettel való kommunikáció jócskán megnyújtja a végrehajtási időt.

Továbbfejlesztési lehetőségei

- A szuperpixelek kívánatos mérete fix megadás helyett számolható a slide méretarányából, ugyanis az is természetesen ismert információ, hogy mekkora egy átlagos méretű sejtmag, átlagos vastagságú hámréteg (stb). A régióméret mesterséges meghatározása is úgy történt, hogy minden régió „néhány sejtmag” méretű legyen.
- A jelenlegi algoritmus minden egyes szuperpixel szomszédaival együtt vett kis környezetét megvizsgálja, így nyilván olyan részleteket is, amik textúrahátáron, több textúra találkozásánál helyezkednek el. Ezekre a textúrabesorolás bizonytalan vagy fals eredményt adhat. Valószínűleg lehetséges a textúradarab homogeneitásának ellenőrzése, mégpedig (a szuperpixel és szomszédai között számított) lokális szimmetriatulajdonságok vizsgálatával (az összes természetes textúránkra jellemző egyfajta szimmetria).
- Az algoritmus pontosságához a kulcsot a távolságszámítási metodika adja, ennek a folyamatos fejlesztése elsődleges prioritás. Van esély arra, hogy a jelenlegi hisztogram alapú előszűrést inkább GLCM-en alapuló módszerrel fogjuk leváltani, az LBP helyett vagy mellett pedig más módszerekkel is folyik a próbálkozás (Gabor-filter).
- Jelentősen pontosítható lenne az osztályozás, hogyha (ahogy a problémák részénél is kifejtettük) az általánosságot feladva olyan háttérinformációkat is felhasználnánk, mint pl. textúratípusok elhelyezkedése (hámszövet a metszet szélén) vagy szomszédságai (hámszövet mellett kötőszövet, hámszövet közelében zsírszövet). Ezt megtehetjük, hiszen valójában nem célunk általános textúraklasszifikáló algoritmus fejlesztése, de a kellő pontosságot nem ezek segítségével kell elérnünk, hanem ezeket „biztosítéknak” tekinthetjük a pontos eredmény eléréséhez. Ezeket a szabályokat a felhasználó közreműködése nélkül is felépíthetjük, hiszen az elérni kívánt cél része, hogy a felhasználó javíthatja a klasszifikációt, ha rossz lenne. Minden ilyen javított, vagyis biztosan jónak tekinthető klasszifikáció viszont megvizsgálható a fenti szempontok szerint.
- A hasonlósággal kapcsolatban felvetett probléma (hogy a hasonlóság a felhasználó gondolkodásától függ) megoldása érdekében esetleg felvethetőek lennének további paraméterek, amelyek segítségével a felhasználó megadhatja, milyen hasonlóságra gondolt. Ez a felhasználó dolgát jelentősen nehezítené.

9. Tesztelési terv

A tesztelési terv magába foglalja a hatékonyságvizsgálatot, vagyis az implementált algoritmusok vizsgálatát olyan szempontból, hogy mennyire eredményesen működnek az algoritmusok, és ebből kifolyólag mennyire alkalmas orvosi eszközökben történő felhasználásra. A hatékonyságvizsgálat előre definiált metrikák és tesztminták alapján történik. A teszt eredményeképp egy átfogó képet kapunk az implementált algoritmusok által elérhető eredményekről és a hiányosságairól. A 7. táblázat bemutatja a hatékonyságvizsgálat során használt számítógép konfigurációját.

Paraméterek	Leírás
Név:	Hawlett-Packard Z800 Workstation
Processzor szám:	2 db
Processzor típusa:	Intel® Xeon® E5645 (2.40 GHz, 12 MB cache, 6 cores)
Memória típusa:	4 GB 1333 MHz DDR3 ECC Unbuffered RAM
Merevlemez:	Hitachi® 500 GB 7200 rpm SATA NCQ
Operációs rendszer:	Windows® 7 Professional 64 bit

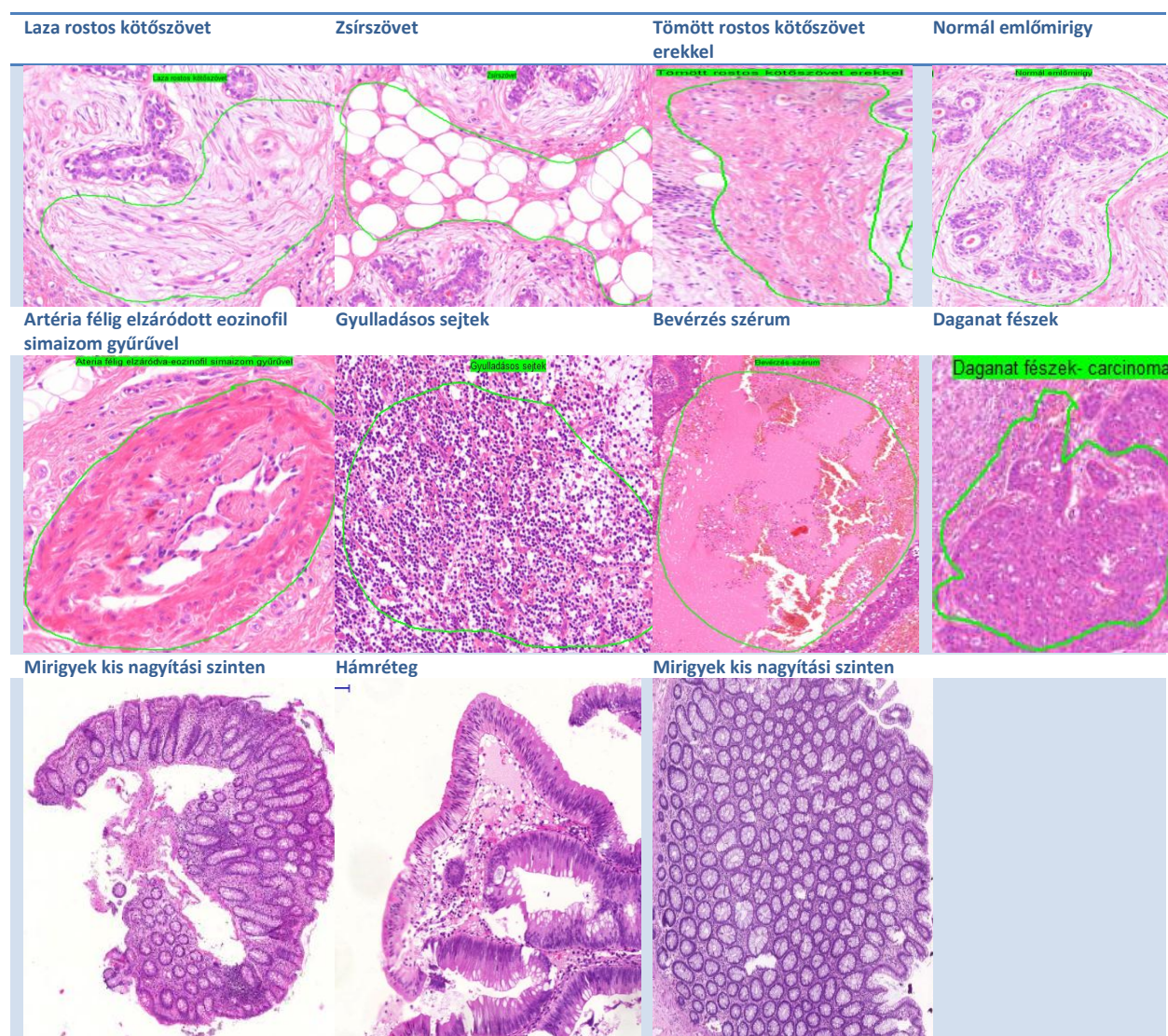
7. táblázat: A hatékonyságvizsgálat során használt konfiguráció.

9.1. Hatékonyságvizsgálat

A hatékonyságvizsgálat alatt az implementált textúra alapú osztályozó algoritmusok teljesítményének mérését értjük. Az irodalomban rengeteg metrikát, paramétert és eljárást lehet találni képfeldolgozó algoritmusok vizsgálatára [34][35].

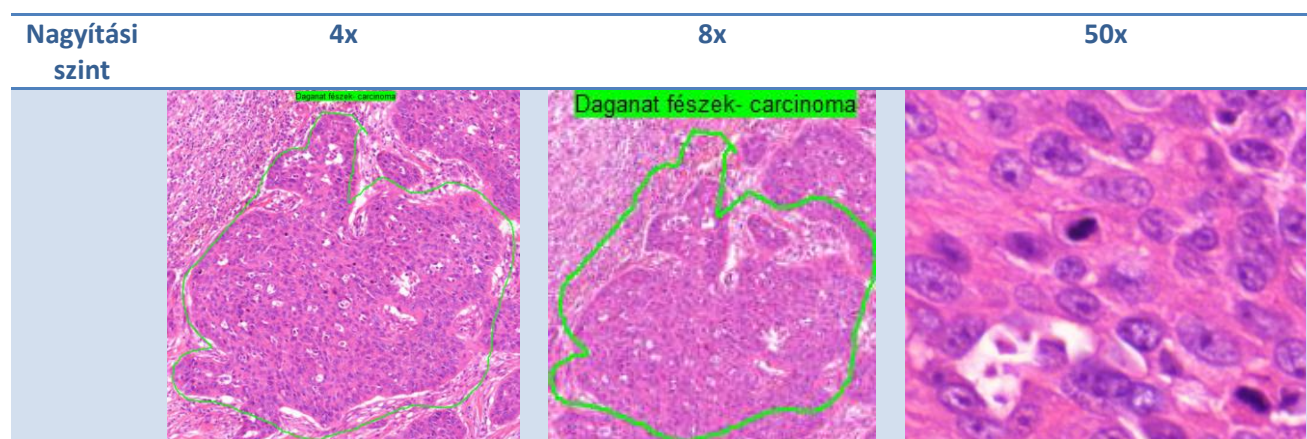
A textúra alapú osztályozó algoritmusoknak a következő felsorolásban található szöveti részeket kellene az orvosok számára automatikusan felismerni és bejelölni (a 7. táblázatban a képek is láthatóak):

- Betegségekre jellemző szövettípusok
 - Daganatfészek
 - Bevérzés szérum
 - Gyulladásos sejtek
 - Artéria félig elzáródott eozinofil simaizomgyűrűvel
- Általános szöveti részek
 - Normál emlőmirigy
 - Tömött rostos kötőszövet erekkel
 - Zsírszövet
 - Laza rostos kötőszövet
 - Mirigyek kisebb nagyítási szinten, ahol homogénebb a textúrájuk
 - Nyálkahártya kötőszöveti része
 - Hámsejtréteg
 - Kehelysejt



8. táblázat: A felismerendő szöveti mintázatok.

A 8. táblázatban szereplő szöveti képek mintáin látható, hogy azonos nagyítási szinten lettek elkészítve. A hatékonyságvizsgálat során azonban figyelembe kell venni azt, hogy a különböző nagyítási szinteken a textúrák jellemzői megváltozhatnak. Amint azt a 9. táblázat is szemlélteti, 4-szeres nagyítási szinten a daganatfészkek textúrája sokkal homogénebb, mint az 50-szeres nagyításban. Ebből a tulajdonságból következik, hogy az algoritmusokat különböző nagyítási szinteken is vizsgálni kell, hiszen az eltérő textúrajellemzőknek köszönhetően esetleg változhatnak a hatékonysági értékek.



9. táblázat: Daganat fészekről készült szöveti kép, különböző nagyítási szinteken.

A szöveti részek kiválasztása után a megfelelő metrikák kiválasztására kerül a sor. Ehhez be kell vezetni néhány fogalmat:

- „reference result set”(referencia eredmény készlet): Azon pixelek halmaza, ami a referenciaképen meg lett jelölve egy hozzáértő személy által (patológus, onkológus). Ezeket a képeket „gold standard” képeknek is szokták nevezni.
- “test result set”(teszt eredmény készlet): Azon pixelek halmaza, amit a tesztelt algoritmus produkál, vagyis találatként jelöl meg.

A referencia eredmény készletet és a teszt eredmény készletet kell felhasználni a paraméterek kiszámításához. Az irodalomból kiválasztott pixelszintű paraméterek a következők [35]:

- Fals negatív találatok (FN):
A referencia eredmény készlet tartalmazza, de a teszt eredmény készlet nem. Vagyis meg lett jelölve mint adott szövethez tartozó rész, de az algoritmus nem találta meg.
- Fals Pozitív találatok (FP):
A teszt eredmény készlet tartalmazza, de a referencia eredmény készlet nem tartalmazza. Vagyis az algoritmus megtalálta, de a referenciában nem tartozik az adott szöveti részhez.
- Igaz Pozitív találatok (TP):
A referencia eredmény készlet és a teszt eredmény készlet is tartalmazza. Vagyis meg lett jelölve és megtalálta az algoritmus.
- Igaz Negatív találatok (TN):
Sem a referencia eredmény készlet, sem a teszt eredmény készlet nem tartalmazza. Vagyis nem lett megjelölve és az algoritmus sem találta meg.
- Pontosság:
$$\frac{TP+TN}{TP+TN+FP+FN}$$
 Ez egy százalékos mérőszámot ad meg, ahol 100% azt jelenti, hogy a referencia eredmény készlet pontosan megegyezik a teszt eredmény készlettel. Vagyis az algoritmus pontosan ugyanazt adja vissza, amit az orvos megjelölt a referencia képen. A 0% pedig azt adja meg, hogy az algoritmus nem ad megfelelő eredményt.
- Felidézés:
$$\frac{TP}{TP+FN}$$
 Ez mérőszám azt jelzi, hogy mennyit talált meg az előre bejelöltek közül. Ha mindet megtalálta, akkor 1, amennyiben kevesebbet talál a megjelöltekből, akkor 0-hoz közelít ez a szám.

- Precízió:

$\frac{TP}{TP+FP}$ Találatok helyességének a száma. Ha csak helyeset talál meg akkor 1. Ha több helytelenet talál meg, akkor 0-hoz közelít ez az érték.

Az algoritmusok hatékonyságvizsgálatához hozzátartozik még a futásidő. A vizsgálat során ezek a paraméterek lesznek kiszámítva. A szöveti részek és a paraméterek sokasága miatt lehetetlen lenne ebben a dolgozatban mindet bemutatni és vizsgálni, ezért a legegyszerűbbek kerültek kiválasztásra és ezekkel lesznek tesztelve az implementált algoritmusok. A függelékben megtekinthetők a tesztelés során használt „referencia eredmény készlet” képek (F03. számú függelék).

9.2. Használhatósági vizsgálat

A helyes működés és az életben való felhasználhatóság bizonyításához szükség van egy stabil és jól átgondolt tesztadatbázisra, mely segítségével releváns tesztkörnyezetet lehet kialakítani a már ismertetett textúra alapú kereső algoritmusok valós környezetben való teszteléséhez. A vizsgálat során 42 mintából 1039 textúradarabot használtunk fel, minden minta HE festésű vastagbél preparátumból lett kiválasztva és feltöltve a használhatóság bizonyítására.

9.2.1. A megfelelő minták kiválasztása és feltöltése

A tesztek során olyan textúra adatbázist kellett kialakítani, amely segítségével minden szöveti komponens elkülöníthető a többitől. Ennek érdekében a textúrarészeket a megfelelő körülményekkel kellett megválasztani az algoritmusok teszteléséhez.

A digitalizált tárgylemezről készített minták csoportosítása:

- Szövetrészek szerint
 - Mirigy
 - Hám
 - Kötőszövet vagy stóma
- Épségüket tekintve
 - Egészséges
 - Adenómás
 - Karcinómás

A minták a 3DHitech Kft által fejlesztett Panoramic Viewer szoftverrel lettek annotálva a szövetrészeknek megfelelően. Az annotálás minden szöveti részre el lett végezve, az egészséges, adenómás és karcinómás esetekben is. A Panoramic Viewer szoftver segítségével az annotációk a digitalizált tárgylemezben is el lettek tárolva, ezzel lehetővé téve a későbbi megismételhetőséget.

Mirigy			
Típus:	Egészséges	Adenóma	Karcinóma
Darabszám összesen: 292 db	124 db	133 db	35 db

10. táblázat: Mirigy textúrák darabszáma az egyes esetekben.

A 10. táblázatban látható a mirigyeket tartalmazó textúrák száma, amelyek a tesztek során felhasználásra kerültek. Az annotáció során azért kellett mind a három típusból mintákat venni, mert az egyes típusokhoz tartozó textúra részek jelentős mértékben különböznek, nem csak formájukat tekintve, hanem a színüket is, amely alapján az algoritmusok elvégzik a megfelelő osztályzásokat.

Stróma			
Típus:	Egészséges	Adenóma	Karcióma
Darabszám összesen: 299 db	105 db	102	92 b

11. táblázat: Stróma textúrák darabszáma az egyes esetekben.

A tesztek során a stróma vagy kötőszöveti textúrák (11. táblázat) is feltöltésre kerültek az adatbázisba. Minden mintából és azon belül a típusokból azért kell nagyszámú mintát felhalmozni az adatbázisban, mert a minták készítése során alkalmazott festési eljárások nem feltétlenül produkálnak ugyanolyan színintenzitásokat és festettséget. Ezzel a módszerrel nem kell külön a festéseknek profilokat létrehozni.

Hám			
Típus:	Egészséges	Adenóma	Karcióma
Darabszám összesen: 233 db	80 db	101 db	52 db

12. táblázat: Hám textúrák darabszáma az egyes esetekben.

A harmadik szöveti rész a hám, amelynek a darabszámát a 12. táblázat szemlélteti. Az algoritmusok alapvetően képekkel dolgoznak, amely képeken megjelenik a preparátum, valamint egy fehér rész, ami gyakorlatilag a preparátumot övezi. Tehát a képeken az algoritmusoknak e szöveti komponenseken kívül a háttérnek definiált fehér részeket is fel kell ismernie és megkülönböztetni a többi textúra típustól. Ennek érdekében a textúra adatbázisba definiálásra került egy háttér típusú textúra is, amelyből két fajtát lehet megkülönböztetni (13. táblázat). Az egyik a homogén, nem szennyezett háttér, a másik pedig a szennyezett, ez utóbbi a preparátum előkészítése során szennyeződik a legkülönbözőbb módon. A homogén, nem szennyezett háttér elválasztására természetesen egyszerűbb algoritmus is alkalmas lehet, a tesztelések során azonban a kétfajta háttér egyfajta textúraként került vizsgálatra.

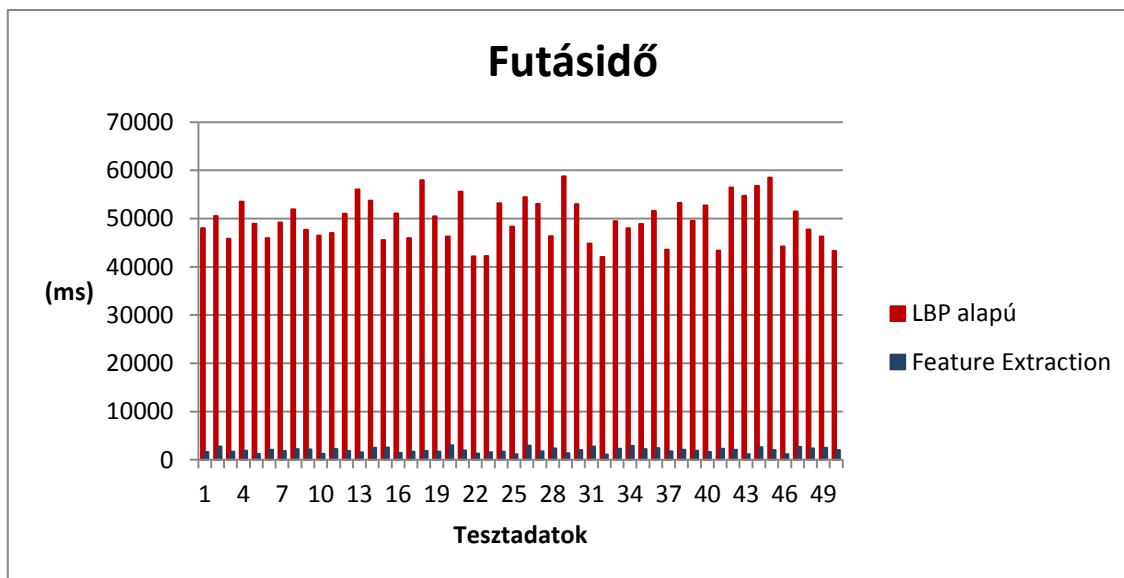
Háttér		
Típus:	Nem szennyezett	Szennyezett
Darabszám összesen: 215 db	108 db	107 db

13. táblázat: Háttér textúrák az egyes esetekben.

A minták kiválasztását követően tesztelési célból egy a helyi hálózaton elérhető gépre lett telepítve egy PACS szerver, alap beállítási paraméterekkel. A kliens program megfelelő konfigurálását követően a feltöltés sikeresen megtörtént a kiválasztott textúrákkal. A tesztek során használt minták részletes adatai az F06. Függelékben tekinthetők meg.

10. Elért eredmények és értékelés

A hatékonyságvizsgálathoz a 3DHsitech által adott anonim szöveti mintákat használtuk fel. A tesztek elkészítéséhez 20 digitális mintából vettünk adatokat, amelyből 50 ROI-t elemeztünk (F01. számú függelék). Ezek közül 32 ROI volt egészséges, 18 beteg. A dolgozat a futásidő, pontosság, felidézés és precízió eredményeket mutatja be részletesen. Az F04. számú függelékben az adatsorok egy része, az F05. számú függelékben az algoritmusok pontosságának összehasonlítása tekinthető meg.

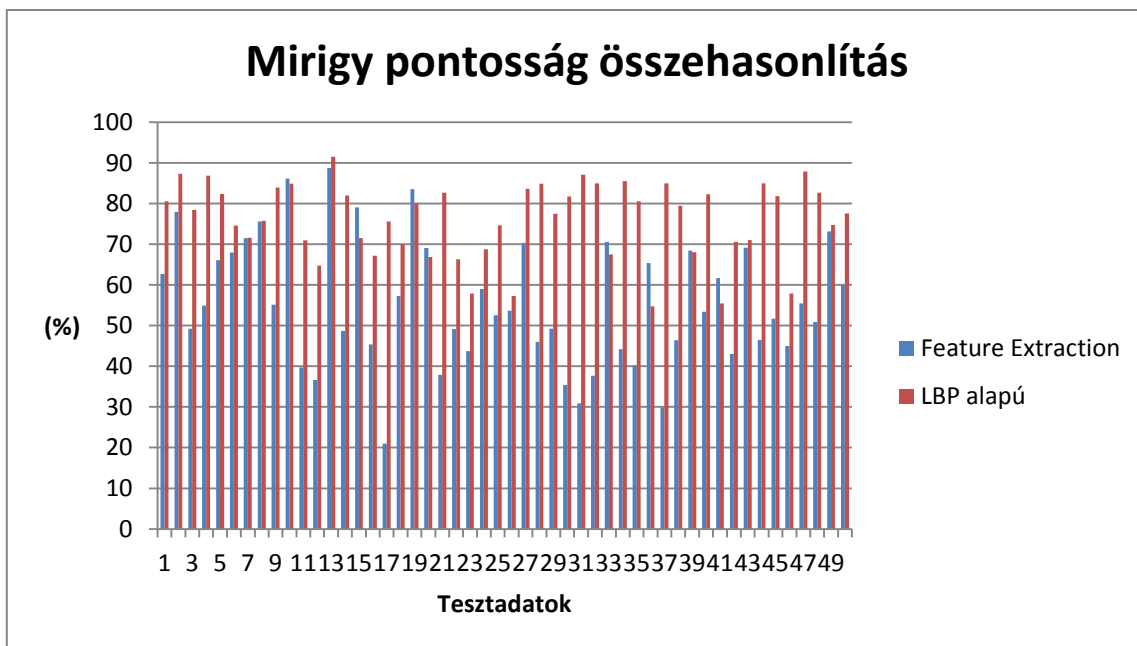


21. ábra: A futásidők összehasonlítása a saját LBP alapú és a 3DHsitech Feature Extraction algoritmusai között.

A 21. ábra szemlélteti a tesztek során kapott futási időket algoritmusonként. A piros színnel megjelölt adatsor a Feature Extraction textúraosztályozó algoritmus, amelyet a 3DHsitech Kft-től kaptunk meg tesztelésre. A kék színű adatsor az általunk implementált algoritmus. Az ábrán jól látható, hogy az LBP alapú algoritmus körülbelül 40-szer lassabb futási időket produkált a tesztek során.

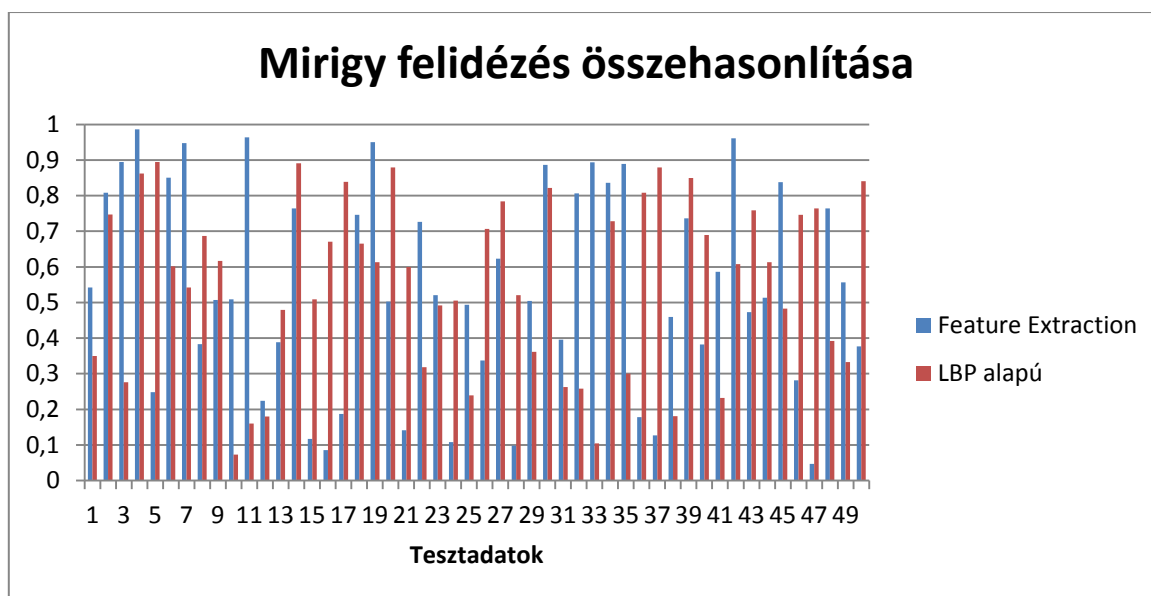
Ez a különbség a jelenleg stabilnak tekintett Matlab implementációval lett mérve. A rosszabb eredmény nagyrészt abból fakad, hogy az algoritmus Matlab runtime alatt fut, és az ezzel való kommunikáció rányomja a bélyegét az eredményre, csakúgy, mint a Matlab kód optimalizátlansága (sok minden iteratív módon megvalósított).

Az LBP kísérleti C# implementációjával kapcsolatban olyan tesztek végeztek, amelyekben ugyanazon tesztképekre mértük a Matlab és a C# modul teljesítményét. Ezen tesztek során a C# javára körülbelül 7x-es sebességelőnyt mértünk, ugyanolyan minőségű eredmény mellett. Ez az előny a C# implementáció soros működése mellett volt mérhető. Az implementáció emellett kifejezetten könnyen többszálúsítható (a megvalósítás során a jó párhuzamosíthatóság végig a legfontosabb szempontok között volt), és ez a futtató géptől függően még tovább növelheti a sebességet.



22. ábra: A hatékonyságvizsgálat során kapott pontossági adatok.

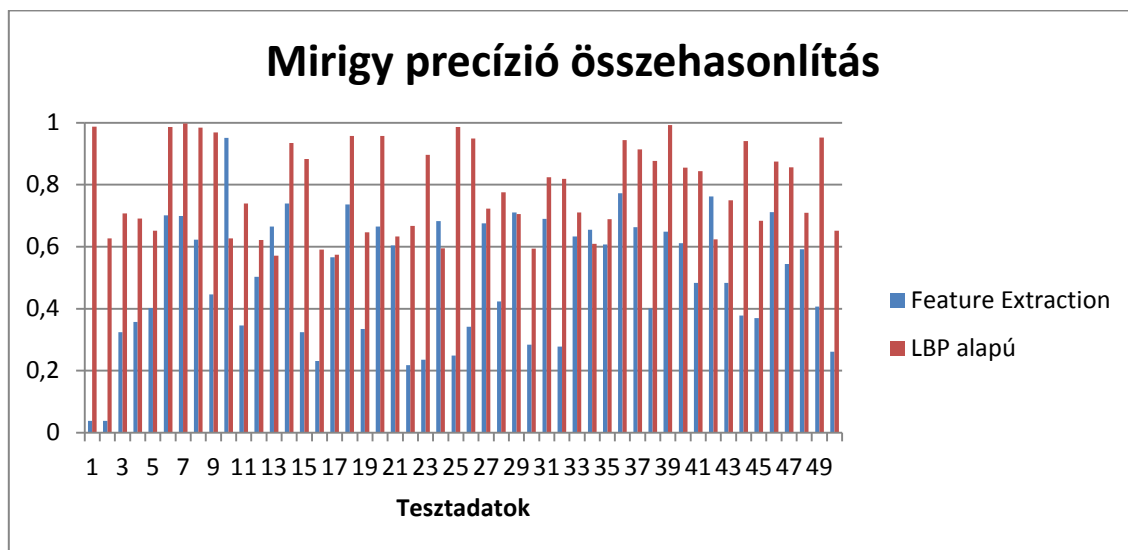
A tesztek során lefuttattuk a vizsgált algoritmusokat, majd a teszteredmény és a referenciakészlet vizsgálatával minden képre meghatároztuk az előre definiált paramétereket (TP, TN, FP, FN), ezek alapján pedig kiszámoltuk a származtatott eredményeket. A 22. ábrán láthatóak a pontosság értékei, ez azt mutatja meg, hogy hány százalékban egyezik a referencia eredmény készlet a teszt eredmény készlettel. Az általunk tervezett algoritmus pontossága a vizsgált teszteseteknél 2 alkalommal egyezett meg a Feature Extraction algoritmus pontosságával, ezt a kettő pontot kivéve azonban minden esetben meghaladta azt. Az LBP algoritmus nem csak hogy pontosabb, de ezt a pontosságot sokkal kisebb szórással teljesíti. A másik algoritmus láthatóan a gyengébb eredményt is nagyobb eltérésekkel éri el.



23. ábra: A hatékonyságvizsgálat során kapott felidézési adatok.

A pontosságon túl fontos a felidézés nevű származtatott eredmény, amely azt mutatja, hogy a referencia bejelölt pixeljei közül mennyit talált meg a vizsgáló algoritmus. 1-es értéknél minden pixelt

megtalál az algoritmus, 0-nál egyetlenegy sem. A felidézés eredményei a 23. ábrán tekinthetők meg. A grafikonon a kék a Feature Extraction algoritmus eredményeit, a piros az LBP alapú algoritmus eredményeit mutatja. Amint az látszik, az eredmények vegyesek. Az LBP algoritmus esetében ez annak köszönhető, hogy a fals pozitív találatok elkerülésére törekszik olyan módon, hogy ha egy terület besorolásában nem biztos, akkor arra a területre inkább egyáltalán semmilyen eredményt nem ad. A Feature Extraction algoritmus jóval „bátrabban” ad eredményeket. Ezt a pontossága megsínyli, de a felidézés metrikájában jól teljesíthet akkor is, ha például a teljes területet egyazon osztályba tartozónak találja (ekkor ugyanis legalább egy osztálynak biztosan minden pixelét megtalálta). Ezt az adatot emiatt szigorúan csak a többivel együtt lehet értelmezni.



24. ábra: A hatékonyságvizsgálat során kapott precízió adatok.

A harmadik vizsgált paraméter a precízió. Pirossal jelenik meg a 24. ábrán az LBP alapú algoritmus és kékkel a Feature Extraction algoritmus precíziós adatsora. 1-es precíziónál az algoritmus csak helyes pixeleket talál meg, minél kisebb ez a szám, annál több helytelent talált meg a vizsgálatok során. Az általunk implementált algoritmus jóval precízebbnek bizonyult a teszt során, ugyanis minden referenciaképen több helyes pixelt talált meg mint a másik algoritmus. A vizsgált 50 teszt közül 15 esetben közel minden megtalált pixel helyes volt.

Összegzés

A rendszer több egyénileg fejlesztett részből áll, amelyek külön nehezen bemutatathatóak, de együtt egy komplex és hasznos rendszert alkotnak a korábban fejlesztett MorphCheck immundiagnosztikai rendszerrel együtt. Az új komponensek elkészítése során sikerült egy, az egészségügyi ellátásban univerzális és hasznos képtárolási rendszert felhasználni az orvosi textúra alapú képi adatbázisok használatához. Ez a rendszer a DICOM szabványon alapszik és egy PACS szerverhez kapcsolódva képes tárolni és visszaadni a szabványos DICOM fájlokat. Kliensoldalról egy lokális PACS-hoz csatlakozik a rendszer és lehetőség van globális is kapcsolódni. Ezen adatbázis feltöltéséhez egy nagyon jól alkalmazható felhasználói felület lett kialakítva. Mind a képi adatbázis rendszert, mind a feltöltési lehetőséget sikerült a meglévő rendszerbe integrálni, ezzel növelve a rendszer hatékonyságát és használhatóságát.

A fejlesztések eredményeként sikerült a rendszerben hatékonyan alkalmazni az általunk fejlesztett LBP alapú osztályozó algoritmust, ami nagyban elősegíti az orvosok diagnosztikai munkáját. Tesztelési célból integrálásra került a 3DHistech Kft Feature Extraction kereső algoritmus is.

Mindkét algoritmus beépítését előre definiált metrikák alapján történő hatékonyságvizsgálat követte. Ezen metrikák között találhatóak a következők: igaz pozitív, igaz negatív, hamis pozitív, hamis negatív találatok és az algoritmusok futási idői. Ezek alapján lettek kiszámolva az összehasonlítás alapját képező mérőszámok.

Az általunk megtervezett és implementált LBP alapú textúraosztályozó algoritmust Matlabban prototipizáltuk és C#-ban is elkészítettük a kísérleti implementációt. A Matlab változat közel 47-szer lassabb a Feature Extraction algoritmusnál, de az előre definiált hatékonysági metrikákból a pontosság 20%-al jobb. A C# nyelvű implementáció azonos eredmény mellett a sebességhátrányból 7x-es sebességelőnyt kovácsolt, a többszálúsítása pedig még további jelentős potenciált rejt a gyorsításra.

A rendszert teszteltük egy valóshoz közeli tesztkörnyezetben a használhatóság bizonyítására, amely teszt során 1039 textúrával feltöltött adatbázist felhasználva sikeresen futott az általunk implementált LBP alapú algoritmus.

Irodalomjegyzék

- [1] Bándi István Imre, Reményi Attila László, 2009. Tudományos diákköri dolgozat, Nagyfelbontású Digitális Szöveti Képek Kötegetelt Feldolgozása
- [2] National Electrical Manufacturers Association: Digital imaging and communications in medicine (DICOM), 2004. <http://dicom.nema.org/>.
- [3] Radiwiki (2012), In:
http://www.radiologia.hu/radiwiki/index.php/Digit%C3%A1lis_k%C3%A9palkot%C3%A1s,_DICOM-szabv%C3%A1ny
- [4] Radiwiki (2012),
http://www.radiologia.hu/radiwiki/index.php/PACS_fel%C3%A9p%C3%ADt%C3%A9se
- [5] Veréb Krisztián, doktori (PHD) értekezés Debreceni Egyetem 2004, Vizuális információ kinyerés és tartalom alapú kinyerési technikák képi adatbázisokban,
http://ganyemedes.lib.unideb.hu:8080/dea/bitstream/2437/78302/1/de_3125.pdf
- [6] Blakiston Division, McGraw-Hill 1968., Manual of Histologic Staining Methods of the Armed Forces Institute of Pathology 3rd edition
- [7] Lillie, R.D., and Fullmer, H.M.: Histopathologic Technique and Practical Histochemistry, 4th ed., New York, McGraw-Hill, 1976, pp. 205-208.
- [8] Luna, L.G.: Manual of Histologic Staining Methods of the Armed Forces Institute of Pathology, 3rd ed., New York, McGraw-Hill, 1968, pp. 34-35.
- [9] ELTE (2012), Mikrotechnikai alapismeretek: A biológiai anyag előkészítése fény-és elektronmikroszkópos morfológiai vizsgálatokhoz,
http://anatomia.elte.hu/Oktatas/Modszerek/Modszerek_jegyzet.rtf
- [10] J. Eakins and M. Graham. Content-based image retrieval. Technical report, University of Northumbria, Newcastle, 1999.
- [11] I. Sethi and I. Coman. Mining association rules between low-level image features and high-level concepts. In SPIE Data Mining and Knowledge Discovery, volume 3, pages 279–290, 2001.
- [12] Michael Lew, et al., ACM Transactions on Multimedia Computing, Communications, and Applications, pp. 1-19, 2006.
- [13] Y. Chen, J. Wang, and R. Krovetz. An unsupervised learning approach to content-based image retrieval. In IEEE International Symposium on Signal Processing and its Applications, pages 197–200, July 2003.

- [14] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000. ISSN 0162-8828.
- [15] Bohern BF, Hanley EN Jr., Extracting Knowledge from Large Medical Databases: an Automated Approach. *Computers and Biomedical Research*, 1995. 28:191–210.
- [16] Márta, Máriáss., Vastagbélrák. [Online] 2009.
<http://www.hazipatika.com/topics/daganatos/denc/Vastagbelrak?did=295>.
- [17] Valcz, Gábor., "Az ép és beteg vastagbél szöveti felépítése." *biotech.nik.bmf.hu*. [Online] 2008. október. [Hivatkozva: 2009. március 10.]
http://biotech.nik.bmf.hu/web/uploads/slides/DKM/2007_11_05_Az_ep_es_a_beteg_vastagbel_szovetani_szerkezete.pdf.
- [18] György, Fehér., A háziállatok funkcionális anatómiája. Budapest : Mezőgazda Kiadó, 2004.
- [19] Pál, Röhlich., Szövettan. Budapest : Egyetemi Tankönyv, 1999.
- [20] American Society of Hematology (2012), <http://www.hematology.org/About-ASH/>
- [21] American Society of Hematology, Hematology Image Bank (2012),
<http://imagebank.hematology.org/>
- [22] Image After online image bank (2012), <http://www.imageafter.com/index.php>
- [23] TinEye reverse image search engine (2012), <http://www.tineye.com/about>
- [24] RevImage, Reverse image search engine (2012), <http://www.revimg.net/what.php>
- [25] Alma Mater Studiorum Università Di Bologna, Windsurf software library (2012), <http://www-db.deis.unibo.it/Windsurf/library.html>
- [26] Alma Mater Studiorum Università Di Bologna, Windsurf (2012), <http://www-db.deis.unibo.it/Windsurf/windsurf.html>
- [27] Alma Mater Studiorum Università Di Bologna, Framework of Windsurf (2012), <http://www-db.deis.unibo.it/Windsurf/framework.html#QP>
- [28] mdcm open source DICOM library (2012), <https://github.com/rcd/mdcm/wiki/Users>
- [29] ClearCanvas Inc (2012), <http://www.clearcanvas.ca/dnn/>
- [30] DICOM is easy(2012), <http://dicomiseasy.blogspot.hu/>

- [31] Marcilan, Maxillofacial Radiology Cone Beam CT Interactive Learning And Navigation(2012),<http://www.marcilan.com/2009/11/installing-dcm4chee-an-open-source-pacs-under-windows-7-with-mysql-database/>
- [32] NEMA (National Electrical Manufacturers Association): Digital imaging and communications in medicine (DICOM) standard, Part 3: Information Object Definitions,2011.
<http://medical.nema.org/Dicom/>.
- [33] NEMA (National Electrical Manufacturers Association): Digital imaging and communications in medicine (DICOM) standard, Part 5: Data Structures and Encoding,2011.
<http://medical.nema.org/Dicom/>.
- [34] R. Kohavi, F. Provost, "Glossary of Terms", Machine Learning vol.30 issue 2, Springer Netherlands, 1998, pp. 271-274, ISSN 08856125
- [35] W.A.Yasnoff, J.K.Mui, J.W.Bacus, "Error measures for scene segmentation", Pattern Recognition 9, pp. 217-231
- [36] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk: SLIC superpixels. EPFL Technical Report no. 149300, 2010. június
<http://www.google.hu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCQQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.165.8269%26rep%3Drep1%26type%3Dpdf&ei=a3iWUPnkMsSE4gTi4YFY&usq=AFQjCNGHx7f5QNdAfVX8wMsQqEOZ0-8CIQ> (2012. 11. 04.)
- [37] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence (ISSN: 0162-8828), vol. 34, num. 11, p. 2274 - 2282
- [38] T. Ojala, M. Pietikäinen, and D. Harwood (1996), "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59.
- [39] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, Kaleem Siddiqi: Fast Superpixels using Geometric Flows. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009. <http://www.cs.toronto.edu/~sven/Papers/turbopixels.pdf> (2012. 11. 04)
- [40] VLFEAT Open Source library (2012), Online: <http://www.vlfeat.org/> (2012. 11. 04.)
- [41] T Ojala and M Pietikäinen: Texture Classification.
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm (2012. 11. 04.)
- [42] Haralick: Textural Features for Image Classification. IEEE Transactions on Systems, Man and Cybernetics, 1973. <http://dceanalysis.bigr.nl/Haralick73-Textural%20features%20for%20image%20classification.pdf> (2012. 11. 04.)

- [43] T. Ojala, M. Pietikäinen, and D. Harwood (1994), "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.
- [44] "An HOG-LBP Human Detector with Partial Occlusion Handling", Xiaoyu Wang, Tony X. Han, Shuicheng Yan, ICCV 2009
- [45] Strickland NH (1998) Impact of the filmless hospital. Proceedings of the Pacific Medical Technology Symposium, Honolulu, Hawaii. pp 236–239.
- [46] Nicola H. Strickland, 2000. Arch Dis Child 2000;83:82-86 doi:10.1136/adc.83.1.82
PACS (picture archiving and communication systems): filmless radiology, Archives of DISEASE IN CHILDHOOD ,(2012) <http://adc.bmj.com/content/83/1/82.full?sid=059e1c18-480c-4375-a7b3-76b552146298#ref-3>
- [47] Langer S, Wang J (1996) A goal based cost-benefit analysis for film versus filmless radiology departments. J Digit Imaging 9:104–112.
- [48] Van Gennip EMSJ, Fischer F, Glaser KH, et al. (1996) Guidelines for cost-effective implementation of picture archiving and communication systems. An approach building on practical experience in three European hospitals. Int J Biomed Comput 43:161–178.
- [49] Strickland NH (1996) Some cost benefit considerations for PACS: a radiological perspective. Br J Radiol 69:1089–1098.
- [50] D. Crockford, RFC 4627, The application/json Media Type for JavaScript Object Notation (JSON) , <http://tools.ietf.org/html/rfc4627>
- [51] ECMA Standard, ECMAScript Language Specification 5.1 Edition 2011
jun.(2012), <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>
- [52] Rafael C. Gonzalez, Richard E. Woods 2002. Digital Image Processing Second Edition, p 351
- [53] D. Lee Fugal: Conceptual wavelets in digital signal processing. Space & Signals Technical Publishing 2009, ISBN 978-0982199459

Függelék

F01. Függelék: Értékrepresentációk

Érték reprezentáció	Leírás	Hossz
AE	Application Entity	16 Byte maximum
AS	Age String	4 Byte
AT	Attribute Tag	4 Byte
CS	Code String	16 Byte maximum
DA	Date	8 Byte
DS	Decimal String	16 Byte maximum
DT	Date/Time	26 Byte
FL	Floating Point Single	4 Byte
FD	Floating Point Double	8 Byte
IS	Integer String	12 Byte maximum
LO	Long String	64 Byte maximum
LT	Long Text	10240 Byte maximum
OB	Other Byte	Nincs limit
OF	Other Float	Nincs limit
OW	Other Word	Nincs limit
PN	Person Name	64 Byte maximum
SH	Short String	16 Byte maximum
SL	Signed Long	4 Byte
SQ	Sequence of Items	Nincs limit
SS	Signed Short	2 Byte
ST	Short Text	1024 Byte maximum
TM	Time	16 Byte maximum
UI	Unique Identifier	64 Byte maximum
UL	Unsigned Long	4 Byte
UN	Unknown	Nincs limit
US	Unsigned Short	2 Byte
UT	Unlimited Text	Nincs limit

F01. táblázat: A DICOM szabvány lehetséges érték reprezentációi [33].

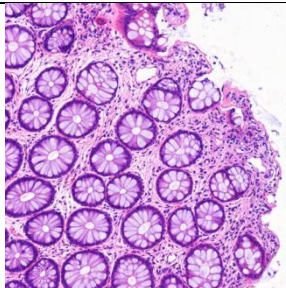
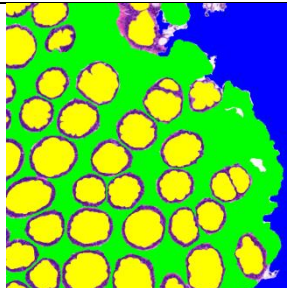
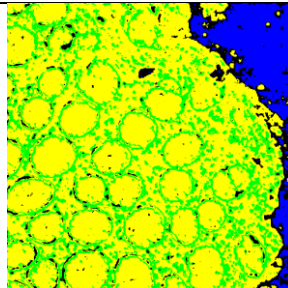
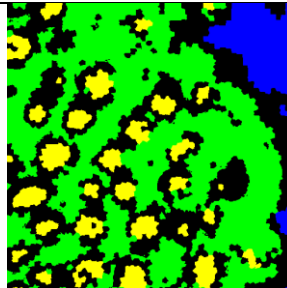
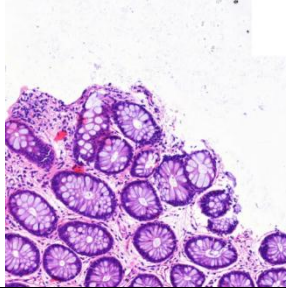
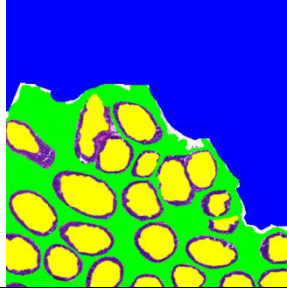
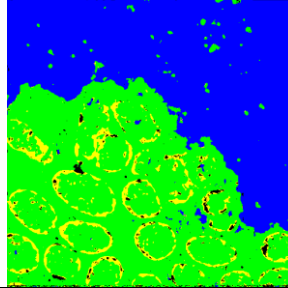
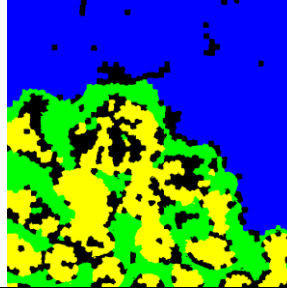
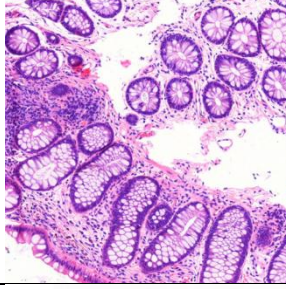
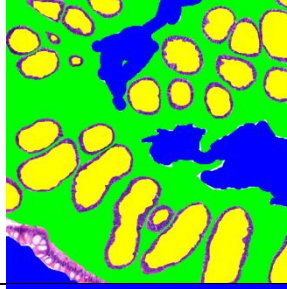
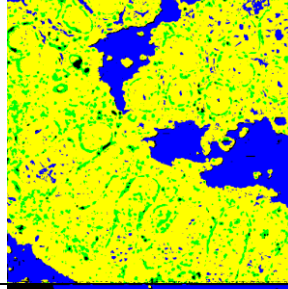
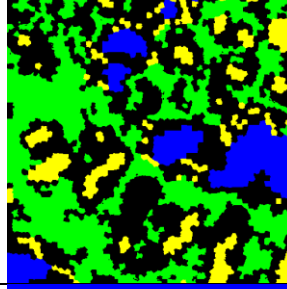
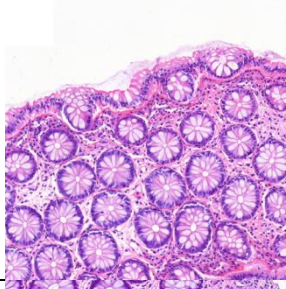
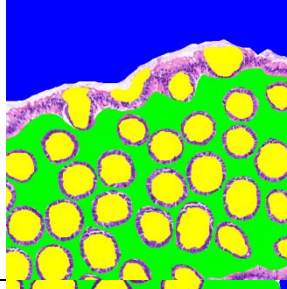
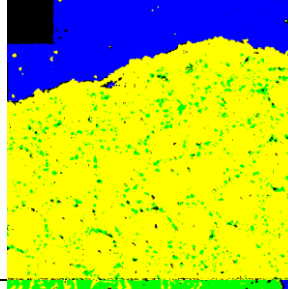
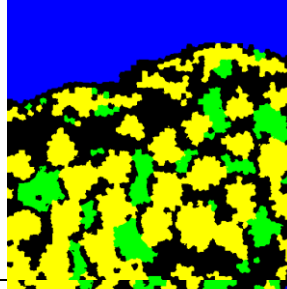
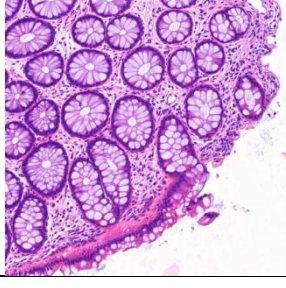
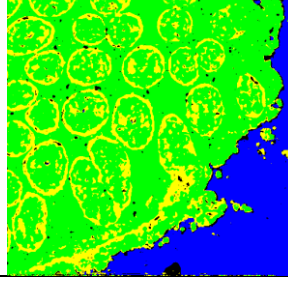
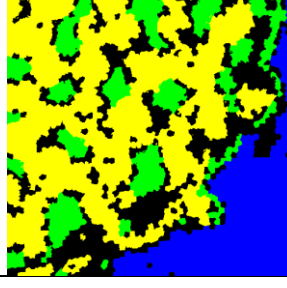
F02. Függelék: A felhasznált DICOM tagok

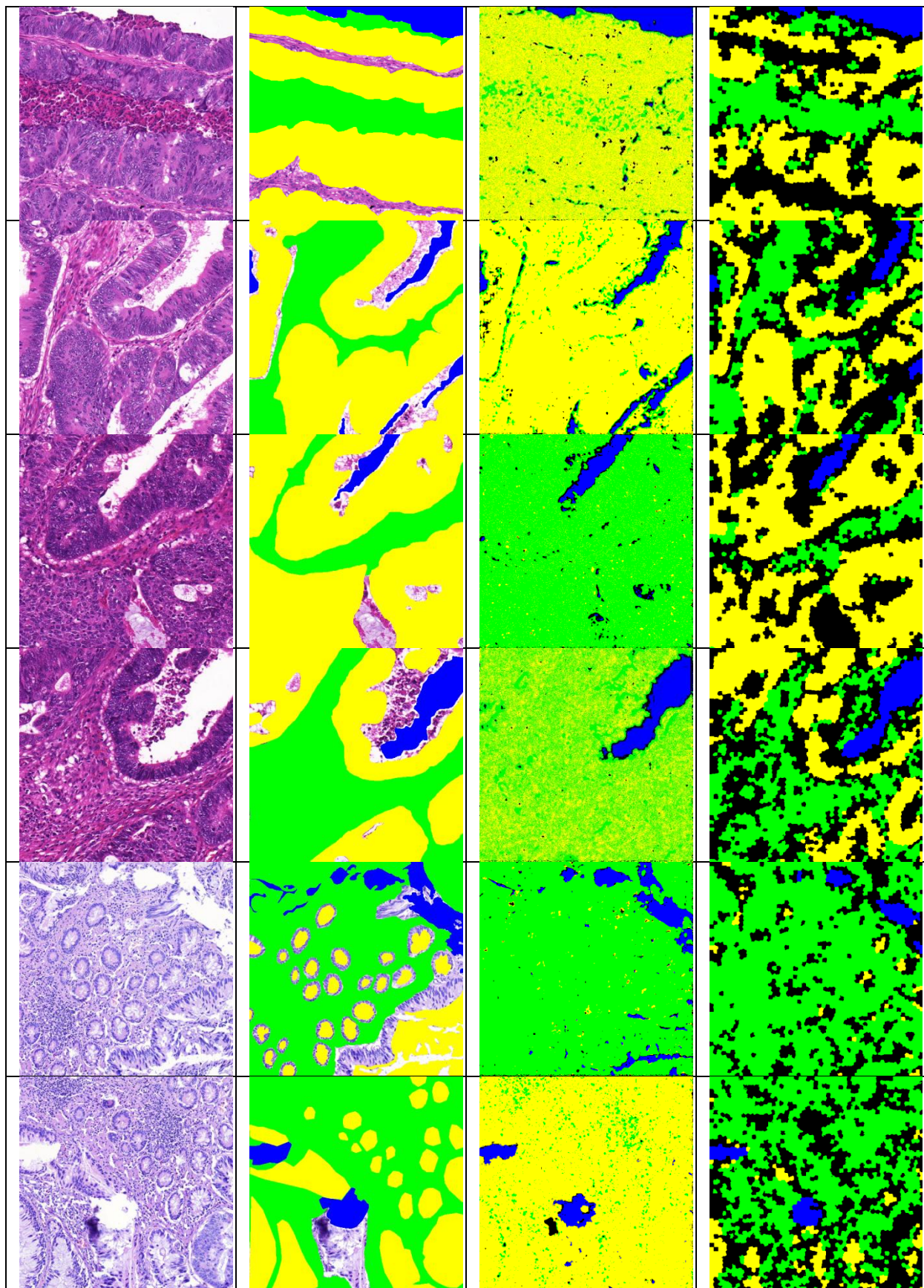
Attribútum név	Tag	Érték reprezentáció
Image Comments	(0020,4000)	LT
Columns	(0028,0011)	US
Rows	(0028,0010)	US
Stage Name	(0008,2120)	SH
Study Date	(0008,0020)	DA
Hardcopy Device Manufacturer's Model Name	(0018,101B)	LO
Study Status ID	(0032,000A)	CS
Source Applicator Name	(300A,0294)	LO
Therapy Description	(0018,0039)	CS
Admitting Diagnoses Description	(0008,1080)	LO
Admitting Diagnoses Code Sequence	(0008,1084)	SQ
Overlay or Image Magnification	(2040,0072)	CS
Spatial Resolution	(0018,1050)	DS
Topic Author	(0088,0910)	LO
Treatment Time	(3008,0251)	TM
Digital Signatures Sequence	(FFFA,FFFA)	SQ
ROI Name	(3006,0026)	LO
ROI Physical Property	(3006,00B2)	CS
ROI Display Color	(3006,002A)	IS
Color Image Printing Flag	(2000,0062)	CS
ROI Area	(60xx,1301)	IS
ROI Description	(3006,0028)	ST
ROI Observation Label	(3006,0085)	SH
ROI Observation Description	(3006,0088)	ST

F02. táblázat: A DICOM szabvány felhasznált tagjai [32].

Az F02. táblázat összefoglalja, hogy a megvalósítás során mely tagok kerültek felhasználásra a DICOM fájlok elkészítése során. A táblázat első oszlopában az attribútum neve látható, a második oszlopban a tag, vagyis az azonosítója az attribútumnak. A legutolsó oszlop pedig az adott taghoz tartozó érték reprezentációt mutatja. Az érték reprezentáció az adott taghoz tartozó maximális adatmennyiség értékét határozza meg. A táblázatban csak a szabványos tagok szerepelnek, ezen felül szükség van néhány olyan tagra is amit a szabvány nem tartalmaz. A nem létező tagokat privát adattagként hozzá lehet csatolni a DICOM fájlhoz, de ezek nem fognak megjelenni a PACS-ban, csak a feldolgozás során a mi általunk elkészített algoritmusok tudják hasznosítani.

F03. Függelék: Hatékonyságvizsgálat szöveti eredményei

Eredeti kép	Referencia eredmény készlet	Feature Extraction algoritmus teszt eredmény készlete	LBP alapú algoritmus teszt eredmény készlete
			
			
			
			
			



F03. táblázat: A hatékonyság vizsgálat eredményei közül néhány.

F04. Függelék: Hatékonyságvizsgálat adatsorai

Mirigy											
Mérések	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Idő (ms) LBP algoritmus	48032	50503	45815	53523	48907	45962	49185	51919	47646	46471	46991
Idő (ms) 3Dhistech	1615	2744	1672	1877	1198	2067	1811	2183	2148	1237	2182
FP Feature Extraction algoritmus	92143	13401	126074	117353	29336	59611	66202	2558	65956	2108	155022
FN Feature Extraction algoritmus	5620	44358	7109	890	59608	24496	8565	187394	51590	39774	3050
TP Feature Extraction algoritmus	72248	3312	60425	65124	19684	139580	153810	8822	53176	1169	81922
TN Feature Extraction algoritmus	92133	201073	68536	78777	153516	38457	33567	63370	91422	219093	22150
Pontosság	0,62706	0,77967	0,49195	0,54894	0,66071	0,67916	0,714787	0,27539	0,5516	0,84023	0,397
Felidézés	0,92783	0,06948	0,89473	0,98652	0,24825	0,8507	0,947252	0,04496	0,50757	0,02855	0,96411
Precízió	0,43949	0,19817	0,324	0,35689	0,40155	0,70073	0,699098	0,77522	0,44636	0,35673	0,34574
FP LBP alapú algoritmus	349	21190	7693	25449	37833	1395	199	2186	2118	1777	4814
FN LBP alapú algoritmus	50622	12047	48895	9078	8360	65376	74362	61418	40105	37960	71330
TP LBP alapú algoritmus	27246	35623	18639	56936	70932	98700	88013	134798	64661	2983	13642
TN LBP alapú algoritmus	183927	193284	186917	170681	145019	96673	99570	63742	155260	219424	172358
Pontosság	0,80556	0,87321	0,78413	0,86829	0,82379	0,74529	0,715572	0,75737	0,83893	0,84842	0,70953
Felidézés	0,3499	0,74728	0,27599	0,86248	0,89457	0,60155	0,542035	0,68699	0,61719	0,07286	0,16055
Precízió	0,98735	0,62702	0,70785	0,6911	0,65216	0,98606	0,997744	0,98404	0,96828	0,62668	0,73916

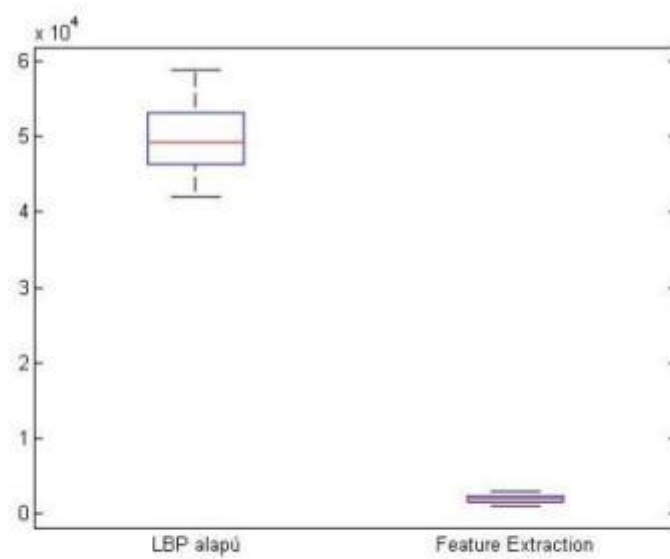
Kötőszövet											
Mérések	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Idő (ms) LBP algoritmus	48032	50503	45815	53523	48907	45962	49185	51919	47646	46471	46991
Idő (ms) 3Dhistech	1615	2744	1672	1877	1198	2067	1811	2183	2148	1237	2182
FP Feature Extraction algoritmus	28080	69419	11520	4760	90417	27505	15468	198861	69468	93718	2905
FN Feature Extraction algoritmus	67979	3635	99470	62273	7182	38473	55326	3120	57023	2681	136626
TP Feature Extraction algoritmus	28485	54916	17097	7613	66799	16866	5035	37247	54472	145159	12245
TN Feature Extraction algoritmus	137600	134174	134057	187498	97746	179300	186315	22916	81181	20586	110368
Pontosság	0,63356	0,72132	0,57661	0,74429	0,62769	0,74831	0,729942	0,2295	0,51748	0,63227	0,46773
Felidézés	0,29529	0,93792	0,14667	0,10893	0,90292	0,30478	0,083415	0,92271	0,48856	0,98187	0,08225
Precízió	0,50358	0,44168	0,59744	0,61529	0,42489	0,38011	0,245574	0,15775	0,4395	0,60767	0,80825
FP LBP alapú algoritmus	33745	8840	17919	3200	4484	14780	24132	3796	16675	66509	36260

FN LBP alapú algoritmus	11519	17044	46620	46853	38166	14583	24040	22389	48441	19878	49829
TP LBP alapú algoritmus	84945	41507	69947	23033	35815	40756	36321	17978	63054	127962	99042
TN LBP alapú algoritmus	131935	194753	127658	189058	183679	192025	177651	217981	133974	47795	77013
Pontosság	0,82733	0,90126	0,7538	0,80906	0,8373	0,88799	0,816238	0,90011	0,7516	0,67046	0,6716
Felidézés	0,88059	0,7089	0,60006	0,32958	0,48411	0,73648	0,60173	0,44536	0,56553	0,86554	0,66529
Precízió	0,71569	0,82442	0,79606	0,87802	0,88873	0,73387	0,600814	0,82566	0,79085	0,658	0,73201

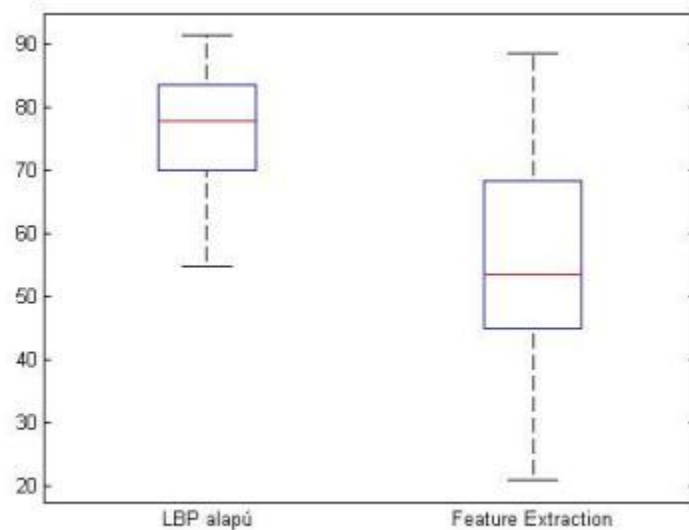
Háttér											
Mérések	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
Idő (ms) LBP algoritmus	48032	50503	45815	53523	48907	45962	49185	51919	47646	46471	46991
Idő (ms) 3Dhistech	1615	2744	1672	1877	1198	2067	1811	2183	2148	1237	2182
FP Feature Extraction algoritmus	27	2608	8532	114	964	184	3430	1891	212	3874	420
FN Feature Extraction algoritmus	17950	5302	5806	12907	7669	2213	1889	306	2810	6717	4065
TP Feature Extraction algoritmus	22798	114627	34324	55408	47703	11743	8717	5200	13959	11953	6898
TN Feature Extraction algoritmus	221369	139607	213482	193715	205808	248004	248108	254747	245163	239600	250761
Pontosság	0,93142	0,96983	0,9453	0,95033	0,96707	0,99086	0,97971	0,99162	0,98847	0,9596	0,98289
Felidézés	0,55949	0,95579	0,85532	0,81107	0,8615	0,84143	0,821893	0,94442	0,83243	0,64022	0,62921
Precízió	0,99882	0,97775	0,80091	0,99795	0,98019	0,98457	0,717626	0,73332	0,98504	0,75523	0,94261
FP LBP alapú algoritmus	2	1699	376	718	439	92	1022	614	265	73	56
FN LBP alapú algoritmus	22451	6639	15428	1156	8414	2433	4733	1240	2608	12818	4854
TP LBP alapú algoritmus	18297	113290	24702	67159	46958	11523	5873	4266	14161	5852	6109
TN LBP alapú algoritmus	221394	140516	221638	193111	206333	248096	250516	256024	245110	243401	251125
Pontosság	0,91435	0,96819	0,93971	0,99285	0,96623	0,99037	0,978046	0,99293	0,98904	0,95082	0,98127
Felidézés	0,44903	0,94464	0,61555	0,98308	0,84805	0,82567	0,553743	0,77479	0,84447	0,31344	0,55724
Precízió	0,99989	0,98522	0,98501	0,98942	0,99074	0,99208	0,851777	0,87418	0,98163	0,98768	0,99092

F04. táblázat: A hatékonyság vizsgálat adatai.

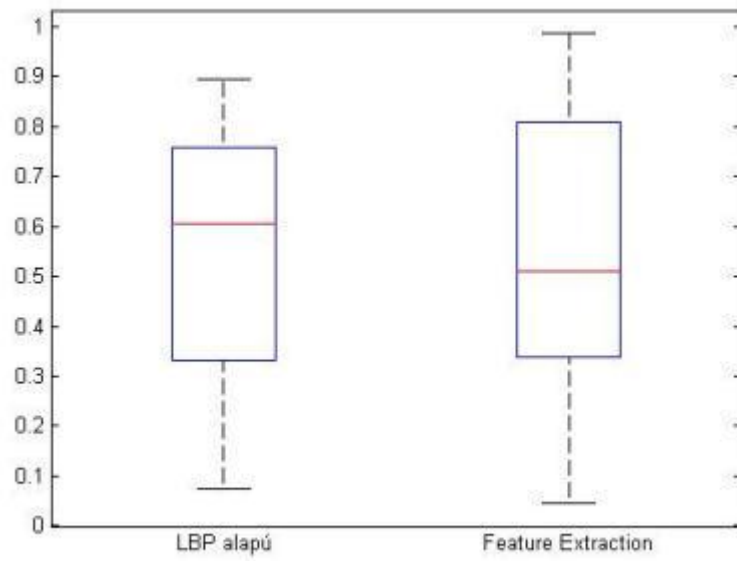
F05. Függelék: Hatékonyságvizsgálat eredményei



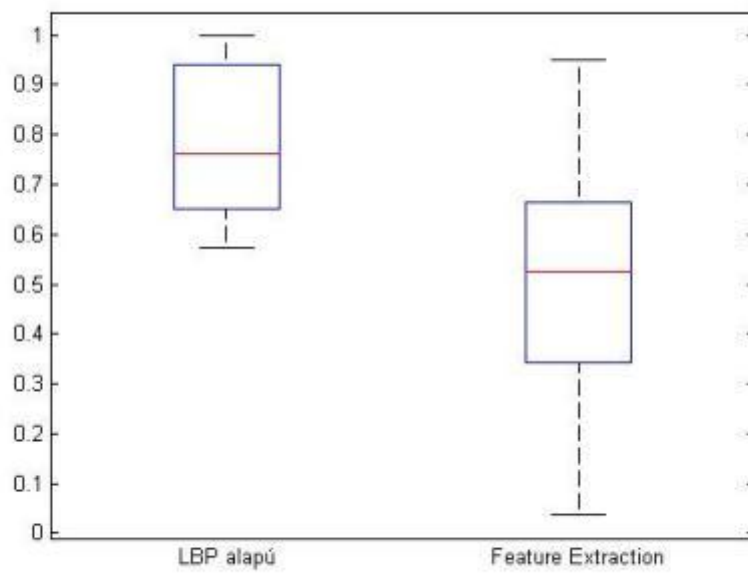
F01. ábra: Mirigy futásidő összehasonlítás



F02. ábra: Mirigy pontosság összehasonlítás



F03. ábra: Mirigy felidézés összehasonlítás



F04. ábra: Mirigy precízió összehasonlítás

F06. Függelék: Használhatósági teszt során felhasznált minták

Minta név	Mirigy			Stróma			Hám			Háttér	
	Ép	Adenóma	Crc	Ép	Adenóm	Crc	Ép	Adenóm	Crc	Nem szennyezett	Szennyezett
10394-04adenomavill+dyspl	-	3	-	-	11	-	-	6	-	6	6
1050-04IIadenomavill+dyspl	-	7	-	-	9	-	-	6	-	5	5
11010-04ep	11	-	-	9	-	-	5	-	-	5	5
11010-04_ep	9	-	-	7	-	-	6	-	-	5	5
11179-05Adenomavill+dysp	-	25	-	11	-	-	-	6	-	3	3
11199-04_II_ep	41	-	-	16	-	-	9	-	-	5	5
1160-05CRCA-B	-	-	7	-	-	13	-	-	19	4	4
11700-04CRCA-B	-	-	5	-	-	8	-	-	-	-	-
11830-03Adenomavill+dysp	-	53	-	-	6	-	-	7	-	2	2
12532-04CRCA-B	-	-	8	-	-	12	-	-	5	3	3
13178-03adenomavill+dyspl	-	4	-	-	3	-	-	10	-	-	-
14270-04adenomavill+dyspl	-	8	-	-	11	-	-	7	-	-	-
14692-04Adenomavillosum	-	5	-	-	5	-	-	8	-	-	-
170-04adenomavillosum	-	2	-	-	2	-	-	1	-	-	-
170-04IIadenomavill+dyspl	-	1	-	-	1	-	-	1	-	-	-
1800-07IIep	6	-	-	6	-	-	5	-	-	6	6
1800-07_II_ep	6	-	-	5	-	-	5	-	-	5	5
1884-07ep	8	-	-	8	-	-	5	-	-	5	6
189-07ep	11	-	-	7	-	-	5	-	-	6	5
189-07_ep	6	-	-	8	-	-	5	-	-	5	5
212-04adenomavillosum	-	4	-	-	8	-	-	8	-	1	1
212-04IIAdenomavillosum	-	3	-	-	8	-	-	8	-	1	1
3671-07IIep	2	-	-	2	-	-	2	-	-	1	1
3674-04-CRC-CD	-	-	-	-	-	12	-	-	-	-	-
3674-04CRCC-D	-	-	-	-	-	9	-	-	-	-	-
3976-05IIAdenomavillosum	-	5	-	-	5	-	-	5	-	4	4
4110-07ep	6	-	-	7	-	-	9	-	-	7	8
4110-07_ep	5	-	-	6	-	-	8	-	-	7	7
4348-04adenomavill+dyspl	-	2	-	-	7	-	-	7	-	1	1
460-04adenomavillosum	-	1	-	-	2	-	-	3	-	1	1
46005Adenomavill	-	-	-	-	6	-	-	1	-	1	1
526-01_I_adenoma_vill-dyspl	-	2	-	-	8	-	-	7	-	1	1
526-01_I_adenoma_vill-dyspl_2	-	2	-	-	5	-	-	5	-	1	1
530-04-IICRCB	-	-	5	-	-	10	-	-	5	-	-

5388-04-CRC-CD	-	-	1	-	-	13	-	-	17	-	-
5725-04-IICRC-CD	-	-	6	-	-	6	-	-	4	-	-
5725-05IICRCA-B	-	-	2	-	-	6	-	-	2	-	-
5814-04CRC-CD	-	-	1	-	-	3	-	-	-	-	-
6134-04ep	5	-	-	5	-	-	5	-	-	5	5
7980-02adenomavillosum	-	6	-	-	5	-	-	5	-	4	3
8656-04ep	5	-	-	5	-	-	6	-	-	5	4
8656-04_ep	3	-	-	3	-	-	5	-	-	3	3
Összesen: 1039 db	124	133	35	105	102	92	80	101	52	108	107

F05. táblázat: A használhatósági teszt során felhasznált minták összefoglalása