

1.GİRİŞ

Patoloji, Yunanca kökenli Pathos ve Logos sözcüklerinden oluşmuş ve normalden farklılaşmış, acayip, hastalık bilimi anlamına gelmektedir (Kumar V 2004). Patoloji bir tanı bilimidir. Patolojide tanıya ulaşılma için makroskobik ve mikroskobik olarak doku ve organların incelenmesi gerekir. Gerek makroskobik gerekse de mikroskobik görüntülerin değerlendirilmesi patoloğun bilimsel yorumuna kalmıştır. Diğer bir ifadeyle, tanıya ulaşırken patoloğun gördüğü görüntü esas alınmaktadır (Avunduk 2003). Değerlendirme bölümü, uzmanlar arasında subjektif karar alınmasına neden olmaktadır. Hatta farklı zamanlarda değerlendirilen görüntülerde bile farklılık göstermektedir. Görüntüler sayısal ortama aktarıldığında ise tanılarda ortak bir objektif değerlendirme yapılabilir.

Patolojide öznitelik çıkarma ve sayma işlemleri için alanların bölümlenmesi temel problemlerdendir. Bilgisayarsız ortamda yapılan bu çalışmalar zor olması ile birlikte öznel sonuçlar vermektedir (Babaoğlu 2005). Bilgisayar Destekli değerlendirmeler nesnel ve hızlı ölçümler için güçlü bir araç olacağını ortaya koymuştur (Fernandez ve ark 1995).

Görüntü analizleri, farklı algoritmalar kullanarak görüntü üzerinde işe yarar bilgi elde etme amaçlı olarak kullanılabilir. Analizler, görüntülerin sayısal değerleri ile ilgilendikleri için görüntünün öncelikli olarak sayısal duruma çevrilmiş olması gerekmektedir.

Bu çalışmada hasta tanısı, tedavisi ve takibi için hayati önem taşıyan patolojik incelemelerde kullanılan bilgisayarlı görüntü analiz programı yapılmıştır.

2.KAYNAK ARAŞTIRMASI

Kaynaklar alfabetik isim sırasına göre ilgili kısımları özet olarak verilmiştir.

Al Sarraj ve ark. (2002), beyin dokusundaki nöron kayıpları ve inkülüzyonlar bilgisayarlı doku analizi ile belirlenmiş. Bu değişikliklerin olduğu alanlar diğer alanlarla kıyaslanarak motor nöron hastalıklarının teşhisi mümkün olmuştur.

Ang ve ark. (2002) EGFR ile immunohistokimyasal boyamalar sonrası yapılan bilgisayarlı doku analizi ile elde edilen sonuçların, klinik gidiş, son noktanın saptanması, lokal ve reyonel nüks ile uzak organ metastazlarının saptanmasında istatistiksel olarak anlamlı değerler verdiği gösterilmiş. Bu anlamlı korelasyonun saptanmasında immunohistokimyasal boyamaların bilgisayarlı doku analizi ile değerlendirilmesinin önemi vurgulanmıştır.

Canzonieri ve ark. (1998), Bilgisayarlı görüntü analizi ile hücre morfolojisi, marker ekspresyonu, DNA indeks ve yapısal değişiklikler belirlenerek teşhise gidilebilmiş. Bu işlemler bilgisayarlı görüntü analizi ile daha objektif olarak yapılabilmıştır.

Dulewicz ve ark. (2001), mesane epitelindeki neoplastik nükleuslar bilgisayarlı görüntüleme analizinde belirlenerek kanser geliştirme riski yüksek vakalar belirlenmiş. Bu işlemin bilgisayarlı görüntü analizi ile kolayca gerçekleştiği vurgulanmıştır.

Erlar ve ark. (1994), Bilgisayarlı görüntü analizi teknolojisinin diagnostik patolojideki önemi belirtilmiş. Bu yöntem sayesinde DNA ploidi analizi, kantitatif immunohistokimya, motilite çalışmalarında ve kromozomal analizde kullanabileceği bildirilmekte. Morfometri ile hücre ve dokuların boyut, şekil ve özelliklerinin kantitatif ölçümü yapılabilmekte imiş. Ayrıca bu işlemler oldukça kısa sürede gerçekleştirilebiliyordu.

Mink ve ark. (1992), meme karsinomlarında immünohistokimyasal olarak belirlenen steroid reseptörlerini bilgisayarlı görüntü analizi ile belirlediklerinde daha objektif sonuçlar elde ettiklerini bildirmişlerdir.

Mozdziak ve ark. (2002), mitotik aktivite gösteren hücreler immünohistokimyasal yöntemlere bilgisayarlı görüntü analizi kullanılarak belirlemiştir.

Okamura ve ark. (1997), Meme karsinomlarında immünohistokimyasal olarak, PCNA, Catepsin D, EGFR boyaması sonrası görüntü analizi ile histolojik grade ve metastatik davranışlar araştırılmış. Veriler arasında anlamlı değişiklikler belirlenmiş. Bu veriler bilgisayarlı görüntü analizi ile hassas bir biçimde saptanabilmiştir.

Setton ve ark. (2002), Deneysel bir çalışmada ligasyon sonrası siyatik sinirde meydana gelen değişiklikler immünohistokimyasal yöntemler sonrası bilgisayarlı görüntü analizi ile zamana bağlı olarak değerlendirilmiştir.

Spyridonos ve ark. (2001), bilgisayarlı görüntüleme analizinin, mesane tümörlerinde kullanıldığında, nükleer atipinin belirlenmesi ve grade'lemede daha objektif değerler verdiği bildirilmiştir.

Zhang ve arkadaşları (2003), bir çalışmada prostat karsinomlarını benign prostat hiperplazisi ve normal prostat dokusundan ayırımında bilgisayarlı görüntü analizi kullanılmış. Stromal elemanlar immünohistokimyasal ve van Gieson gibi konvansiyonel histokimyasal yöntemlerle boyanmış. Boyanma sonrası bilgisayarlı görüntü analizi yapılmış ve aralarında önemli farklılıklar olduğu gözlenmiş. Böylelikle prostatik zonda fark edilmesi oldukça zor olan karsinomların benign lezyonlarla, bilgisayarlı görüntü analizi, immünohistokimya ve histokimya kullanılarak ayrılabilirdiği göstermişlerdir.

3.MATERYAL ve METOD

3.1. Mikroskopik ve Makroskopik Patoloji

3.1.1.Mikroskopik tanımlama

Mikroskopik tanımlama da, hücrelerinin mikroskop altındaki görünimleri, onların birbirleri ile ilişkileri, normalden farklılaşmaları, yerleşim yerleri, dağılımları vs. tarif edilir. Burada aynı zamanda normalden farklılaşan hücreler hemen yanlarındaki normal hücrelerle karşılaştırılır (Castelman 1996).

3.1.2. Makroskopik tanımlama

Makroskopik tanımlama , dışarıdan gözle veya bir büyüteç (lup) ile görülebilen özellikler patolojik disiplin ile bilimsel olarak yapıldığı tanımlama biçimidir (Castelman 1996).

3.1.3. Patoloji

3.1.3.1. Tanım

Patoloji, eski yunanca hastalık anlamındaki 'pathos' teriminden türetilmiştir ve hastalıkların bilimsel yöntemlerle incelenmesi anlamında kullanılır. Daha geniş anlamıyla patoloji, hastalıklara yol açan nedenleri, bunların doku ve organları etkileme biçimlerini, hastalıklı doku ve organların özellikle morfolojik (biçimsel, görüntüsel) özelliklerini inceler (www.patoloji.gen.tr). Bu anlamda patoloji, tıbbın temelini oluşturur.

Patalog, patoloji işlemini yapan kişiye verilen isimdir. Patologdan en çok beklenen, hastalıklı olduğu düşünülen doku ve organları inceleyerek hastaya belli bir hastalık tanısı koyması veya konulmuş olan bir tanının doğruluğunu

değerlendirmesidir. Doku ve organlar vücuttan değişik biçimlerde alınır ve patoloğun incelemesine sunulurlar (Ülker 2003). Lenf düğümü biyopsisi ile lenfoma adlı kötü huylu tümörün tanısının konulması; endoskopik yolla alınmış bir mide biyopsisi örneğinde gastrit mi, peptik ülser mi, kanser mi bulunduğunun saptanması gibi örnekler verilebilir.

3.1.3.2. Patoloji de yapılan işlemler;

Bir hastanenin işleyişi içinde patoloji bölümünün katkısı; hastalardan tarama veya tanı amacıyla hücre/doku örneklerinin alınmasıyla veya organların çıkarılmasıyla başlar. Bu örneklerin önce dış görünüşleri (makroskopi) değerlendirilir ve mikroskop altında incelenmesi gerekli görülen kısımlar seçilerek ayrılır. Patolojik incelemenin en kritik ve en çok deneyim gerektiren aşamasının bu olduğu kabul edilebilir (www.patoloji.gen.tr).

Patolojiyi en iyi yansıttığı düşünülen kısımlar örneklenip, çok ince (4-5 mikron kalınlıkta) kesitlerin alınabilmesine olanak verecek işlemlerden (doku takibi) geçirilir ve hazırlanan kesitler rutin olarak "hematoksilen-eosin" yöntemiyle boyanır. (Hücre çekirdekleri mavi, sitoplazmalar kırmızı boyanır). Daha sonra, bu boyanmış kesitlerin ışık mikroskopunda incelenmesiyle morfolojik bir değerlendirme yapılır. Bu değerlendirmenin birtakım kuralları olmakla birlikte, temelde, morfolojik incelemeler subjektiftir. Bu subjektifliğin asıl nedeni, canlı organizmaların özellikleri için 'normal'in kesin sınırlı olarak tanımlanamamasıdır. (Normal saç rengi nedir? Normal boy kaç santimetredir?) Dolayısıyla; belli bir organ veya hücrenin görünümünün normalden ne kadar sapmış olduğu sorusunun yanıtı, kaçınılmaz olarak kişisel ve subjektiftir.

Patolojik incelemenin sonuçta subjektif olması, onun kuralları ve sistematığı olmasına engel değildir. Tıbbi bir değerlendirmenin işe yararlılığının ve güvenilirliğinin ölçüsü, hastanın tanı ve tedavisine yapılan katkıdır. Bir dokudaki bütün atomların adlarını ve miktarlarını objektif, bilimsel yollarla saptamak mümkündür ancak, bunun bir lenfoma olgusunun tanı ve tedavisine katkısı yoktur.

Subjektif morfolojik değerlendirme, patoloğun tanıya ulaşmada kullandığı yollardan yalnızca birisidir. Patolog, yeri geldiğinde biyokimyasal, farmakolojik, mikrobiyolojik, genetik, moleküler biyolojik verileri kullanabilir; özel yöntem ve düzeneklerin yardımıyla dokular üzerinde nitel (kalitatif) veya nicel (kantitatif) incelemeler yapabilir. Bunlar arasında histokimya, immunohistokimya, in situ hibridizasyon, DNA sitometrisi, dijital görüntü analizi gibi yöntemler sayılabilir. Bu yöntemlerin hemen tümü, ülkemizdeki bir çok merkezde kullanılmaktadır. (www.patoloji.gen.tr).

Patolojide, uyguladığımız yöntemler, kaçınılmaz olarak subjektifliğe neden olmaktadır. Bu subjektiflikten kaynaklanan tanı farklılıkları "gerçek" ve artık "kabul edilemeyecek ölçüde yaygın"dır. Daha çok okuyarak, daha dikkatli olarak, daha fazla özel histokimyasal boya uygulayarak bu subjektifliği olabilecek en az düzeye indirebiliriz belki. Ancak, bu yolla, en basit bir serum bilirubin düzeyi ölçümünün objektiflik düzeyine bile ulaşabilmemiz olanak dışıdır. Ayrıca, morfolojik olarak aralarında hiçbir fark göremediğimiz lezyonların farklı davranış göstermeleri karşısındaki çaresizliğimiz de bizi yeni tanısal ve prognostik yöntemler arayışına itmeli (www.patoloji.gen.tr).

3.1.3.3. Nicel Patoloji

"Görüntü analizi" ile mikroskopik veya makroskopik verilerden elde ettiğimizi dijital ortama aktararak üzerinde işlem yapmamızı ve sonuç üretmemizi sağlayan bir yöntemdir. Bu yöntemin temel amacı patolojik incelemelere "objektiflik" getirmektir. Bu nedenle, bu tür yöntemler için "nicel patoloji", "tanısal nicel patoloji" terimlerinin kullanılması yeğlenmelidir. (Nicel=quantitative). Görüntü analizi, tıp dışında da yaygın olarak kullanılan bir yaklaşımdır ve temelde, matematiğin özel bir uzmanlık alanıdır.

Görüntü analizi amacıyla kullanılan algoritmaların, hesapların tümü matematik teorilerine dayanmaktadır. Bu dala, optik bilimlerin de daha az oranda katkısı vardır. Görüntü analizi uygulamalarının önemli bir kısmını askeri amaçlı "hedef tanıma" projeleri oluşturmaktadır. Tıpta görüntü analizi göreceli olarak daha yakın zamanlarda uygulanmaya başlamıştır. Öte yandan, patolojik tanının daha

objektif hale getirilmesi çabaları bugün anladığımız biçimdeki görüntü analizinden çok daha önce başlamıştır. Bu çabalar; hücrelerin belli kısımlarının uzunluklarının, çaplarının, çevrelerinin, koyuluk derecelerinin bugün ilkel bulabileceğimiz düzenekler ile ölçülmesidir.

Bugün bilgisayar teknolojisindeki gelişmelere koşut olarak ulaştığımız noktada yaptığımız ölçümler, son derece hızlanmış ve bir parça çeşitlenmiş olmak dışında, temelde bu ilk ölçümlerden farksızdır. Tüm bu ölçümler "morfometri" olarak adlandırılabilir (www.patoloji.gen.tr).

- Sitometri: Sitolojik örneklerde yapılan morfometri.
- Histometri: Histolojik kesitlerde yapılan morfometri.
- Karyometri: Hücre çekirdeği ile ilgili ölçümler.
- Mikrofotometri: Belli bir yapının özel koşullardaki ışık geçirgenliğinin derecesinin ölçülmesi. Densitometri ile eş anlamlı olarak kullanılabilir. Bu yöntemle, söz konusu maddenin miktarı değerlendirilebilmektedir.
- Planimetri: Alan veya benzeri iki boyutlu özelliklerin ölçümü.
- Stereoloji: İki boyutlu görüntüler kullanarak, üç boyutlu özelliklerin -hacim gibi- ölçümü.

Günümüzdeki olanaklar ile yapılan tanısal nicel patoloji uygulamalarında ilk adım, rutin olarak değerlendirdiğimiz görüntünün dijital ortama aktarılmasıdır. Bu amaçla, mikroskopta elde edilen görüntü bir videokamera ile alınarak sayısal duruma getirilir. Bu dosya; görüntünün çözünürlüğüne, renklerine, saklama yöntemine, kullanılma amacına göre çok değişik büyüklüklerde olabilir. Örnek olarak, yüksek kaliteli bir görüntü 50-100 sayfalık bir yazının sığabildiği disketlere sığmaz. Dijital bir görüntüyü, saklayabilir, işleyebilir, içeriklerini ayırabilir ve hayal edebileceğiniz her özellik için ölçüm yapılabilir. Dijital bir görüntüde sizin için önemli olan öğelerin daha belirginleştirilmesi mümkün, çoğu kez de gereklidir.

Morfometrik yöntemler ile neler ölçülebilir? Aşağıda, tüm uygulamaları kapsama savi taşımadan, akla ilk gelen örnekler sıralanmıştır. Bunlara, özne konumundaki mikroskopik öğeyi değiştirerek pek çok ekleme yapılabilir. Örnek

olarak; çekirdek yerine, dikine kesilmiş bir sinir lifi, kas lifi veya arterin çapı ölçülebilir.

- Tek boyutlu ölçütler
 1. Çekirdek çapı (en geniş, en dar, ortalama)
 2. Çekirdekçik çapı
 3. İki çekirdek arası uzaklık
 4. Tabaka kalınlığı, bazal membran kalınlığı
 5. Çekirdek çevresi
- Alan ölçütleri
 1. Çekirdek alanı, sitoplazma alanı, gland lümen alanı
 2. Semifer tubulus lümen alanı
- Biçim ölçütleri ve oranlar (hesaplanan ölçütler)
 1. Yuvarlaklık indeksi (form faktörü)
 2. Ağırlık merkezi
 3. Değişkenlik katsayısı (variability)
 4. Çekirdek / sitoplazma oranı
 5. Villus boyu /kript derinliği oranı Kromatin kondensasyon paterni

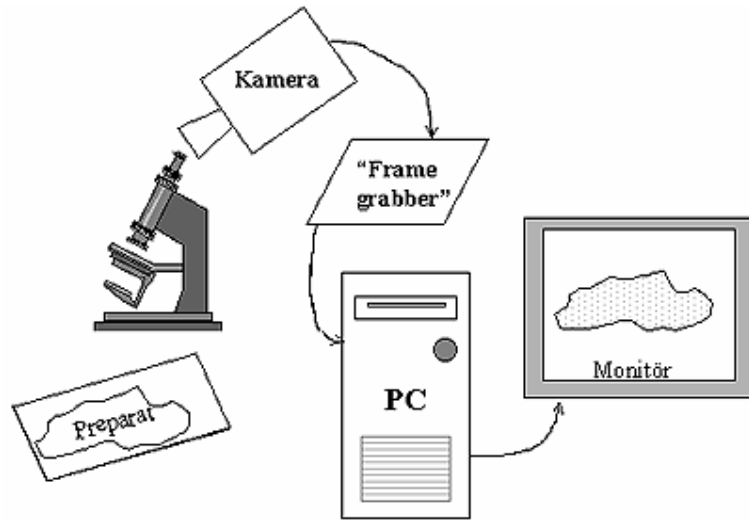
3.1.3.4. Nesnenin belirlenmesi, analog ortamdan bilgisayar ortamına geçiş

Patolojik uygulamalarda sık olarak seçilen ve ilgimizin odak noktasını oluşturan nesne hücre çekirdeğidir. Bu, özellikle tümörlerin tanısında hücre çekirdeğinin morfolojik özellikleri üzerinde çok duruyor olmamızdan kaynaklanmaktadır. Ancak, patolojik bir görüntüde yer alan herhangi bir öge, ilgimizin odak noktasını oluşturabilir. Bilgisayarlı görüntü analizi sistemleri açısından bakıldığında, morfometrik bir çalışmada ölçülecek olacak görüntünün bir astronomik cisme mi, bir insana mı, bir mikroorganizmaya mı ait olduğunun önemi yoktur. Ölçülecek görüntünün bir nesneye ait olması bile gerekli değildir; örneğin, bir endometriyal glandın kıvrımlanma derecesi ölçülebilir.

İlk aşama, görüntünün analog ortamdan bilgisayar ortamına (0'lar ve 1'lerden oluşan bir evrene) aktarılmasıdır. Daha çok mikroskopik görüntüler ile çalışıyor olmamıza karşın, görüntü analizinin mikroskopik nesnelere üzerinde yapılması gibi

bir zorunluluk yoktur. Görüntüyü aktaran düzenekte genellikle bir video-kamera bulunmaktadır. Bu kameranın özellikleri çalışmanın sonraki aşamalarında önem kazanabilir. Çok yüksek çözünürlüklü görüntüler elde edebilen videokameralar ile elde edilen görüntülerin işlenmeleri ve saklanmaları daha zordur. Ayrıca, bu tür kameralarda poz süresi de uzun olduğu için, ortamdaki titreşimler ve ısı değişiklikleri görüntünün kalitesine etki edebilmektedir. En yaygın olarak kullanılan, 512 x 512 piksel çözünürlüktür. (Pixel=picture element=bir resimde kendine özgü rengi-tonu olan en küçük birim). Monokrom bir görüntü, genellikle, her piksele 0 (siyah) ile 255 (beyaz) arasında bir değer verilerek saklanır. İnsan gözü siyah ile beyaz arasında yaklaşık 20 ton ayırabilmektedir. Bu bile, bilgisayarlı bir ortamda insan gözünün yakalaması mümkün olmayan özelliklerin nasıl yakalanabildiğini kavramaya yardımcı olabilir. RGB (red-green-blue) sistemine göre; renkli bir görüntü, üç monokrom görüntünün (kırmızı, yeşil ve mavi) toplamı biçiminde oluşturulmakta ve saklanmaktadır. Başka renk oluşturma sistemleri de bulunmaktadır. Donanımın gelişmesi ile, çok daha yüksek çözünürlüklü sistemler gittikçe yaygınlaşmaktadır.

Densitometrik çalışmalar için kullanılan renkli videokameraların yüksek kaliteli olmaları gerekir. Bazı firmalar, bu amaçla (daha ucuz olan) monokrom kameraları önermektedirler. Çalışma kolaylığı ve kişisel tercihler dışında, kameranın renkli veya monokrom olmasının genel görüntü analizi açısından büyük bir önemi yoktur (www.patoloji.gen.tr).



Şekil 3.1. Analog ortamdan bilgisayar ortamına aktarma işlem şeması

Şekil 3.1.'de, video-kameranın iletmediği görüntü, aynen mikroskopta incelediğimiz gibi, hareketlidir. Işık ayarını değiştirdiğimizde veya şaryoyu oynattığımızda, görüntü de değişir. Bu değişken görüntüyü dondurup adeta onun bir fotoğrafını çeken ve bu fotoğrafı sayısal (digital) ortamda saklanabilir/işlenebilir duruma getiren "frame grabber" adlı bilgisayar parçasıdır. Frame grabber'ın yakaladığı görüntü işlenmeye hazırdır. Bu görüntünün bilgisayar ortamında kalıcı olarak saklanması istenirse, bir dosya halinde saklanması gerekir. En yaygın olarak kullanılan dosyalama formatları şunlardır: BMP, GIF, TIFF, JPEG, TARGA. Bu formatlar, görüntü dosyasının boyutunu (diskte kapladığı yeri) azaltan formüller de içerirler. Bu biçimde saklanan görüntülerin kaliteleri kısmen de olsa değişir. Yukarıda sıralanan dosyalardan "BMP" en çok yer kaplayanı, ancak görüntünün aslına en sadık kalanıdır. Ancak, bu dosya tipi görüntü analizi açısından yeterince kullanışlı değildir. GIF formatı, özellikle internet ortamında iletilen görüntüler için çok yaygın olarak kullanılmaktadır. Dosya boyutunu belirgin biçimde kısaltan bu format, görüntü analizi için çok "kayıplı"dır. JPEG formatında, değişik formüller kullanılarak, dosya boyutunun küçültülmesinin mi, görüntünün kalitesinin mi önemli olduğuna göre ayarlamalar yapılabilmektedir. Çok yüksek kalite gerektirmeyen durumlarda bu format tercih edilebilir. Pek çok yazılım, dosyaları bir formattan diğerine aktarmaya yardımcı olmaktadır. Dolayısıyla, herhangi bir başka ortamda, herhangi bir formatta kaydedilmiş görüntüler üzerinde görüntü analizi çalışması yapılabilmektedir. Görüntüler, "mikroskop-video-kamera-frame grabber" düzeneği dışındaki yollarla da edinilebilirler. Scanner olarak adlandırılan aygıtlar bu amaçla yaygın olarak kullanılmaktadır.

Bizi ilgilendiren nesnenin görüntünün çevresinden ne farkı olduğunu belirlemek görüntü analizi çalışmalarının ilk aşamalarındandır. Ton-renk farklılıkları, biçim farklılıkları ve boyut farklılıkları bu amaçla kullanılabilir. Bu farklılıkların belirginleştirilmesi de mümkündür. Örnek olarak, çekirdekler üzerinde çalışılacaksa ve bunlar (diğer kısımlardan farklı olarak) maviye boyanmışlarsa, bilgisayara tüm mavilerin nesneye ait oldukları söylenebilir. Ancak; çevrede çekirdekteki maviliğe çok yakın yapılar varsa, bilgisayar onları da nesne kabul edebilir. Bu sorunu

gidermenin deęişik yolları bulunmaktadır. Örnek olarak, mavinin insan gözü tarafından seçilemeyecek derecede farklı iki tonu arasındaki fark belirginleştirilerek, bu tonlardan birinin kırmızıya dönüşmesi sağlanabilir. Bu durumda, bilgisayarın neyi nesne olarak seçtiğinin denetlenmesi kolaylaşacaktır.

Son aşamada, seçilen nesnenin beyaz, tüm diğer alanların siyah olması (veya aksi) sağlanır. Bu işlem, monitöre yansıtılmadan da gerçekleştirilebilir. Ancak, bu biçimde 1'ler ve 0'lardan oluşur duruma indirgenmiş görüntülerin elde edilmesi görüntü analizi için genellikle zorunludur. İlgilendığımız şey hücrenin çekirdeği ise, görüntüdeki tüm çekirdekler (ve yalnızca çekirdekler) beyaz; tüm diğer alanlar ise siyah olmalıdır. Bu görüntü "binary" bir görüntüdür. (Şekil 3.2.'de, sağdaki görüntü, soldakinin binary biçimi olarak kabul edilebilir; burada bizi ilgilendiren nesne, T-shirt'dür).



Şekil 3.2. Bir görüntünün üzerinde istenilen bölümü çıkarma.

Bizi neyin ilgilendirdiği bir kez belirlendikten sonra, ölçümlere sıra gelmektedir. Şekil 3.2.'de, tişörtün kapladığı alan hesaplanabilir. Tişörtü çevreleyen sınırın ne derecede girintili çıkıntılı olduğu değerlendirilebilir; en, boy saptanabilir. Bu görüntü şablon olarak kullanılarak, asıl görüntüde buna denk gelen kısmın (tişörtün) renginin koyuluk derecesi belirlenebilir.

3.1.3.5.Verilerin değerlendirilmesi

Morfometri ile elde edilen verilerin değerlendirilebilir nitelikte olmaları, çalışmanın tüm aşamalarının titizlikle hazırlanmasıyla mümkündür. Hücre boyutları

ile ilgili çalışmalarda genellikle 100-200 hücrenin değerlendirilmesi gerekmektedir. Ancak, "pleomorfizm" in bulunduğu durumlarda, bu sayılar yeterli olmayabilmektedir (Ülker 2003). Ne kadar hücre sayılması gerektiğini belirten birtakım formüller de bulunmaktadır. Bu formüllerde, verilerin istatistiksel değerlendirmeye elverişli sınırlarda olması amaçlanmaktadır. Aslında, genel bir kural olarak, ne kadar çok hücre sayılırsa o kadar sağlıklı sonuç elde edilecektir. Ancak, gerekenden çok hücre sayılması da (yavaş sistemlerde) zaman kaybıdır. Değerlendirilen özellik açısından "normal" olduğu kabul edilen hücrelerden elde edilen ölçüm sonuçlarının standart sapmasının olabildiğince düşük olması gerekir. Standart sapmanın ortalamaya oranının yüzde olarak belirtilmesi olan değişim katsayısı da (coefficient of variation=cv) %5'in altında olmalıdır. Değerlerin dağılımı konusunda fikir veren bu ölçüt, tüm görüntü analizi yayınlarında yer almak zorundadır. Kaynaklarda, değerlendirilen serilerde yer alan normal hücre popülasyonuna ait değişim katsayıları %3,5 ile %10 arasında değişmektedir. Öte yandan, anormal bir hücre popülasyonu için "cv"nin yüksek olması pleomorfizmin derecesini yansıtan önemli bir bulgu olarak değerlendirilir.

3.2. Temel Kavramlar

3.2.1. Görüntü elemanı(pixel)

Örneksel görüntünün sayısala çevrilme işlemi sayısallaştırma (digitizing) olarak tanımlanır. Örneksel sinyal ne kadar fazla parçaya bölünerek sayısala çevrilirse elde edilen sayısal görüntü, orjinaline o kadar yakın olacaktır. Sayısala çevrilen görüntü üzerinde görüntüyü oluşturan en küçük görüntü parçalarına görüntü elemanı (pixel) adı verilir. Görüntü, görüntü elemanlarının içerdikleri sayısal olarak kodlanmış renk kod lûmâsının birleşiminden oluşmaktadır (Gökmen 1990).

Her pixel 3 ana renten oluşur. Örneğin 800*600 çözünürlüğünde bir resim 800*600=480.000 pixelden oluşur. Her bir pixel 3 byte yer kaplarsa, o zaman resim hafızada 480.000* 3 bytelık yer kaplar.

3.2.2. Çözünürlük

Bir resmi dijital ortamda temsil eden nokta miktarıdır. Dolayısıyla bir resim ne kadar çok sıklıkla nokta ile temsil ediliyorsa o resim o kadar kaliteli ve çözünürlüğü fazladır. Birimi dpi(dot per inch=inc başına düşen nokta sayısı)dir (Yaşar 2005).

3.2.3. Dijital Görüntü

Bir görüntü fonksiyonu, bir görüntünün matematiksel olarak tanımlanmasından oluşur. Görüntü fonksiyonu küçük sayılardan oluşan parametrelerin vektörel tanımıdır. Görüntü fonksiyonunun özel bir durumu, parametreleri ve fonksiyon değerlerinin tümü tamsayılardan oluşmuş sayısal ya da ayrık görüntü fonksiyonudur. Bir görüntü üzerinde onun önemli karakteristiklerine bağlı olarak farklı birçok görüntü fonksiyonu tanımlanabilir (Gökmen 1990).

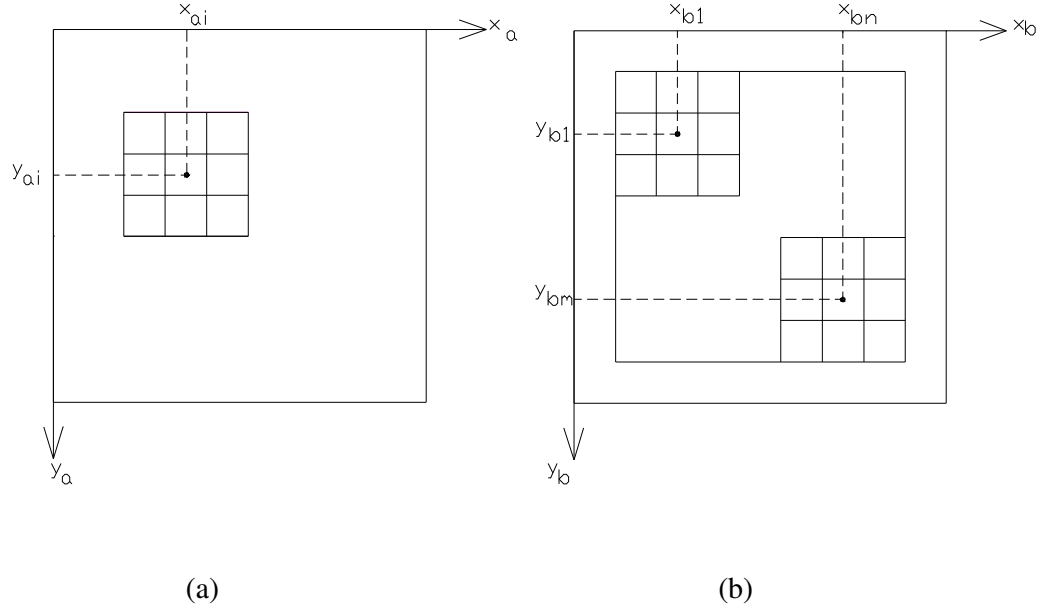
İki boyutlu sürekli uzaydaki analog bir görüntünün $a(x,y)$ 2 boyutlu denk bir uzayda örneklenmesi ile oluşan $a[m,n]$ 'e örnekleme sayısallaştırma olarak tanımlanabilir. Burada fonksiyon N adet satıra ve M adet sütuna bölünmüştür.

$$a(x,y)= \left. \begin{array}{l} a(0,0), a(0,1), \dots, a(0, M-1) \\ a(1,0), \dots \\ \dots \\ \dots \\ a(N-1,0), \dots, a(N-1, M-1) \end{array} \right\} \quad (1)$$

3.2.4. Görüntü korelasyonu

Görüntü korelasyonunun amacı, dijital görüntü üzerindeki homolog noktaların belirlenmesidir. Bu amaçla ilk görüntü üzerinde bir A referans matrisi ve ikinci bir görüntü üzerinde daha büyük boyutta B arama matrisi tanımlanır. Daha sonra referans matris ile arama matrisi içindeki, referans matris boyutundaki her alt

matris arasında bir korelasyon katsayısı hesaplanır (www.ins.itu.edu.tr/skulur/bildiri.ppt).



Şekil 3.3. Görüntü Matrisi

a)Referans matrisi b) Arama matrisi

Şekil 3.3'deki her $f(x,y)$ fonksiyonu bir pixeli oluşturmaktadır. Pixeller birleşerek görüntüyü meydana getirirler.

Bir diğer görüntü fonksiyonu da parlaklığı 3 band genişliğine göre belirlenen renkli görüntü fonksiyonudur. Bu fonksiyon aşağıdaki gibi tanımlanır (Gonzalez R 1993).

$$f(x,y) = [f_{kırmızı}(x,y), f_{yeşil}(x,y), f_{mavi}(x,y)] \quad (2)$$

3.2.5. Renk modelleri

Renklerin gösteriminde standart sağlamak amacıyla çeşitli renk modelleri oluşturulmuştur. Bir renk modeli, üç boyutlu bir koordinat sisteminin ve bu sistem içinde her bir rengin tek bir nokta ile gösterilebileceği bir alt uzayın belirlenmesidir.

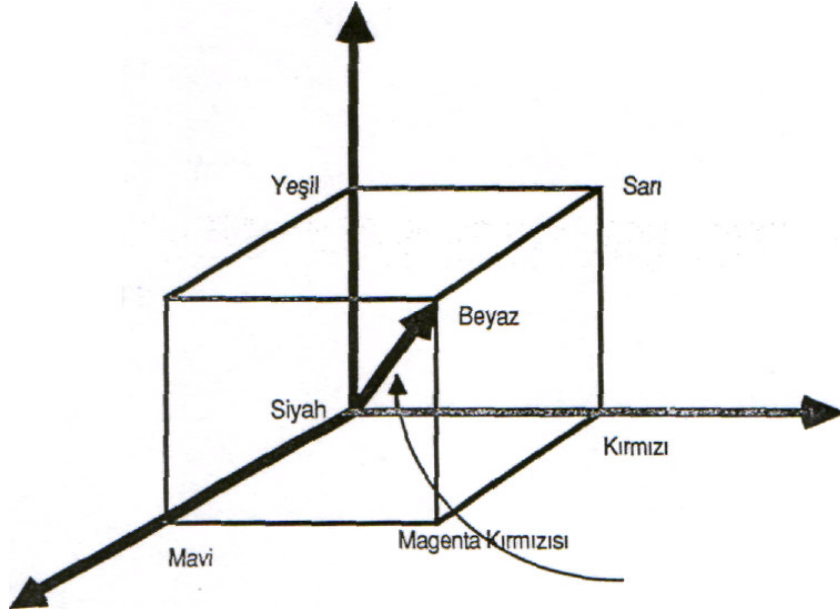
Günümüzde kullanılan renk modelleri çoğunluğu donanıma (renkli monitör ve yazıcı) veya renkli görüntü işleme yöneliktir. Donanıma yönelik en çok kullanılanlar renkli monitörler ve video kameralarda kullanılan RGB modeli, renkli yazıcılar için CMY modeli ve renkli televizyon yayıncılığında standart olan YIQ modelidir (Gümüşer 2001).

3.2.5.1 RGB modeli

RGB modelinde 3 ana renk mevcuttur. Bunlar kırmızı (red), yeşil (green), mavi (blue) renkleridir. Diğer renkler bu renklerin karışımından meydana gelmektedir. Bir renk düzlemindeki renk sayısı, bu düzlemdeki renk değerleri için kaç bit yer ayrıldığına bağlıdır. Örnek vermek gerekirse Mavi renk için 8 bit yer ayrılmışsa, pixel $2^8=256(0-255)$ renk değeri alabilir. 3 renk için aynı şeyi düşünürsek her pixel 24 bit yer kaplamaktadır.

Bu ana renklerin karışımı ile diğer renkler elde edilir. Burada bir pixel $255 \times 255 \times 255$ değer alabilir. Mesela RGB(255,0,0) kodu bize kırmızı rengi verir. Renkli bir kodu gri moda çevirmek için pixele aşağıdaki formül uygulanır (Gonzalez R, Woods 1993). Şekil 3.4'de RGB renk uzayı gösterilmektedir.

$$\text{Gray} := \text{Round}((0.30 * \text{GetRValue}) + (0.59 * \text{GetGValue}) + (0.11 * \text{GetBValue}))$$



Şekil 3.4. RGB modeli renk uzayı

3.2.5.2. CMY modeli

CMY(Cyan magenta yellow) renk modelinde renkler, renklendirici pigmentlerin birincil renkleri olan siyan, magenta ve sarı renklerin çıkarmalı karışımlarından oluşur. Kağıt üzerine renklendirici bırakan pek çok alet(renkli yazıcı veya kopyalayıcı) CMY renk modeli kullanır. Bu aletler CMY cinsinden veri girişine ihtiyaç duyar veya kendisi RGB'den CMY modeline dönüşüm yapar. Bu dönüşüm basit bir denklemle gerçekleştirilir.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3)$$

Burada tüm renklerin [0,1] aralığında olduğu kabul edilmiştir (Pitas 1993).

3.2.5.3. YIQ modeli

Bu model renkli televizyon yayıncılığında kullanılır. Renkli yayınların siyah beyaz televizyonlarda da seyredilebilmesi amacıyla RGB'nin değişik biçimde kodlanmış halidir. Modelin Y bileşeni siyah beyaz televizyon için gereken ışık yoğunluğu verisini içerir. I ve Q bileşenleri, sırasıyla inphase ve quadrature olarak adlandırılan kromatik bileşenlerdir. RGB'den YIQ modeline dönüşüm aşağıdaki şekilde tanımlanmıştır (Gonzalez Woods, 1993).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299.....0.587.....0.114 \\ 0.596... - 0.275.. - 0.321 \\ 0.212... - 0.523.....0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

YIQ renk modelinin görüntü işleme açısından önemi, bu modelde parlaklık ve renk bilgilerinin bağımsız olmalarıdır. Böylece görüntünün renk içeriğini değiştirmeden sadece parlaklık bileşeni üzerinde işlem yapma olanağı doğar (Gümüşer 2001) .

3.2.6. Dijital görüntü işleme

Dijital görüntü işleme; dijital görüntülerin sayısal verileri üzerinde bir takım değişiklikler yapılarak istenen sayısal formları kullanıcıya sunma olarak tanımlanabilir.

Sayısal görüntü işlemenin iki önemli avantajı vardır. Bunlardan biri duyarlılık diğeri de uyumluluktur. Görüntü üzerinde herhangi bir bölge seçilerek sadece o bölüm üzerinde işlem yapılabilir ve donanımsal parçalara ihtiyaç duymaz.

3.2.7. Bitmap tabanlı resimler

Pixellerden oluşan resime Bitmap resim denilir. Resimler yatay ve düşey eksenini göstermek üzere iki şekilde adreslenir. Bitmap resim formatları içinde bmp, gif, jpeg, pcx, tiff... sayabiliriz (Yaşar 2005). Bitmap resimlerin avantajları;

- Daha fazla ayrıntı gösterilebilir.

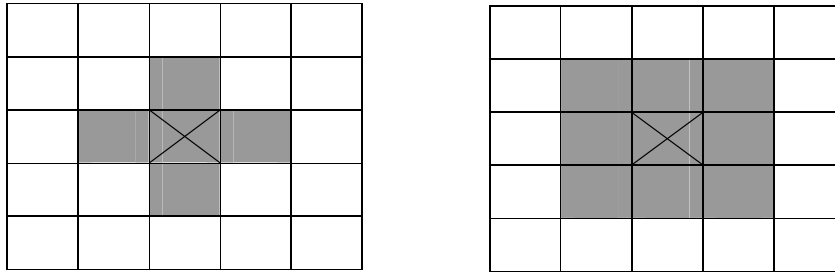
- Formatlar arası dönüşüm kolayca yapılabilir.

Eksileri ise;

- Çözünürlüğe bağlıdır.
- Tekrar boyutlandırıldığında kalitesi azalmaktadır.
- Dikdörtgen bir alanla sınırlıdır. Resmin arka planı vardır.

3.2.8. Komşuluk tipleri

Komşuluk işlemleri modern sayısal görüntü işlemede anahtar rolü oynar. Bundan dolayı bir görüntü işleminde kullanılmış olması, çeşitli komşuluklarla nasıl ilişki kurulduğu ve görüntü örnekleme nasıl olduğu konularını anlamada önem teşkil eder (Jahne 1997) . Donanım ve yazılım sebeplerinden dolayı metot olarak sadece dikdörtgenel örnekleme yapılabilir. Şekil 3.5.'de komşuluk tiplerine örnekler gösterilmiştir.



Şekil 3.5. Örneklem çeşitleri : 4 bağlantılı komşuluk ve 8 bağlantılı komşuluk

3.2.9. Median filtreleme

Median filtresi, görüntüdeki gereksiz ayrıntıları gidermek amacıyla kullanılır. Median filtresi, uygulanacak pikselin komşu pixellerle birlikte sıralandıktan sonra tam ortadaki değer anlamına gelmektedir. Toplam piksel sayısı eğer çift ise, ortadaki iki sayının aritmetik ortalaması alınarak hesaplanır (Babaoğlu 2004).

$$\begin{pmatrix} 9 & 13 & 21 \\ 8 & (18) & 12 \\ 8 & 11 & 15 \end{pmatrix} \quad (5)$$

Bu sıralamadaki median değeri (8, 8, 9, 11, **12**, 13, 15, 18, 21)den 12 olarak bulunur.

3.2.10. Sobel operatörü

Bir görüntüde kenar, gri seviyeleri farklı iki bölge arasındaki geçiş veya sınır bölgesi olarak tanımlanır (Çakar 2003).

Kenar tanıma operatörleri resimdeki kenar bilgisinin, bir pikselin değeri ile komşu piksel değerlerinin ilişkisini inceleme yoluyla bulunacağı fikrine dayanır. Bu çalışmada Sobel kenar tanıma maskeleri kullanılmıştır. Bu maskeler hem yatay hem de dikey yönlerdeki kenarları tanımak için bakarlar ve daha sonra bu bilgileri tek bir ölçüye çevirirler. Maskeler Şekil 3.6.'da görüldüğü gibidir. Birincisi satır, ikincisi ise sütun maskesidir (Umbaugh 1998).

Sütun Maskesi

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Satır Maskesi

$$\begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & -2 \\ -1 & 0 & -1 \end{bmatrix}$$

Şekil 3.6. Sobel matrisi

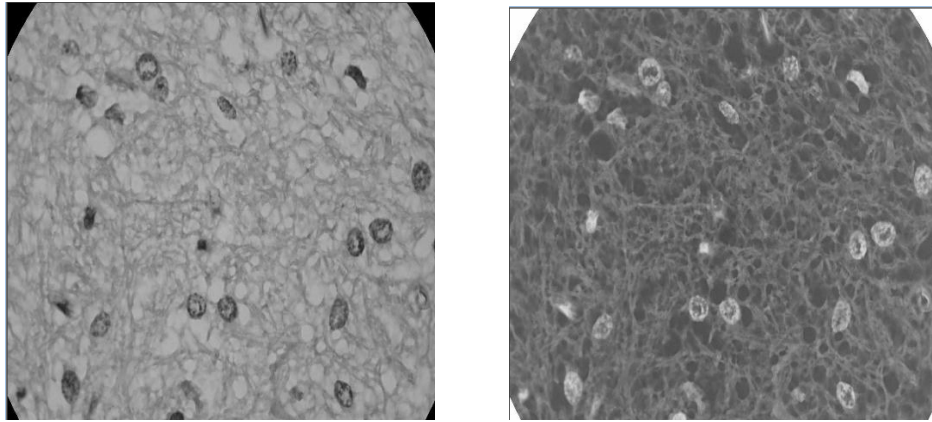
3.2.11. Görüntünün negatifi

İnsan gözü, yapısı gereği açık tonlar arasındaki farklılığı koyu tonlar arasındaki farklılığa göre daha iyi algılayabilmektedir. Resmin gerektiği durumlarda negatifinin alınması algılamanın gücünü artırabilmektedir (www.goruntuisleme.org). Bu nedenle program içerisinde kullanılmıştır.

Bu işlem giriş resimdeki gri-seviye değerlerinin ters çevrilmesiyle yapılmaktadır. Örneğin orijinal resimdeki bir pikselin değeri 0 (siyah) ise negatifinde aynı pikselin değeri 255 (beyaz) olacaktır.

$$\text{Negatif_Renk} = |255 - \text{Renk}| \quad (6)$$

Gri-seviye resim için pikselin maksimum alabileceği değer 255 'dir. Bu resmin negatifini almak için resimdeki piksellerin renk değerlerini maksimum değer olan 255'den çıkarılıp ortaya çıkan sonuç aynı piksele atanırsa ve bu işlem resmin tüm piksellerine uygulanırsa orijinal resmin negatifi alınmış olur. Buradan da anlaşılacağı gibi bir resmi kendi negatifi ile toplarsak ortaya beyaz bir resim çıkar, yani resim içerisinde ki tüm piksellerin değeri 255 olur. Şekil 3.7.'de görüntünün, negatife çevrilmesi işlemine ait örnek yer almaktadır.



Şekil 3.7. Görüntünün Negatife çevrilmesi

3.3. Uygulama

Program Intel Centrino 1.500 MHZ işlemcili, 256 RAM'e sahip bilgisayar da, Windows XP Professional işletim sistemi ortamında Delphi 7.0 programı ile geliştirilmiştir.

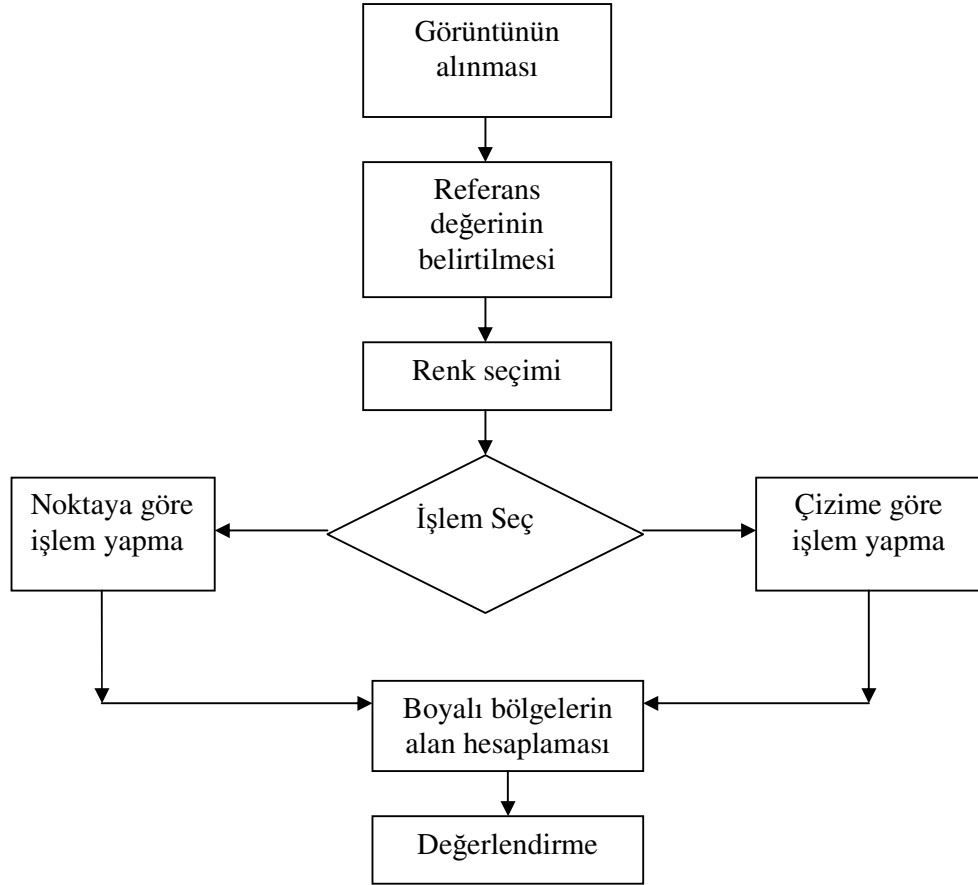
Yapılan çalışma Selçuk Üniversitesi Tıp Fakültesinden alınan patolojik görüntüler kullanılarak gerçekleştirilmiştir. Görüntüler mikroskoptan 24 bit BMP uzantılı ve gri tona çevrilerek kaydedilmiştir.

RGB modelinde her renk tayfın kırmızı, yeşil ve mavi birincil bileşenlerinden meydana gelir. Bilgisayarlarda görüntülerini bu renk modeline uygun olarak kaydederler. Gri ton ölçeği, siyahtan beyaza giden bir doğru üzerinde yer alır (Gökmen 1990).

Bu çalışmada, resimler RGB modeli kullanılarak analiz edilmiştir. RGB modelinde her bir bileşenin değeri ışık yoğunluğundan etkilenmektedir (Oduncu 2005). Bu nedenden dolayı işlemler gri tonlama üzerinde yapılmaktadır.

Program , görüntü üzerinde bir takım hesaplamalar yaparak kullanıcıya sonuca ulaşmasını yardımcı olmayı amaçlamaktadır. Genel işlevi, alınan görüntü üzerinde boyutlarını değiştirebilme, görüntü üzerinde çizim yaparak belirli bir bölgeyi işaretleyip alanını hesaplayabilme, belirli bir noktadaki tolerans değerine bağlı olarak komşu bölgeleri çıkarabilme , gri tonlama veya siyah-beyaz çevirimleri yapma ve son olarak da resmin içerisinde kaç tane farklı bölgelik olduğunu hesaplama gibi görevleri yapmaktadır.

Sistemin akış şeması Şekil 3.8.' de görüldüğü gibidir. İlk aşamada, görüntü alınıp, uzman tarafından referans değeri belirlenir. Referans değeri, uzmanın görüntü üzerinde belirlemiş olduğu ölçü birimini oluşturur. Referans değerleri; mikron, milimetre, santimetre, metre veya kilometre olabilir. İkinci aşamada, uzman tarafından işaretlenecek alanın rengi belirlenir. Sonraki aşamada ise işlem yapılacak algoritma seçilir. İşaretleme yapıldıktan sonra alan hesaplanır ve uzman sonuca göre değerlendirme yapar.



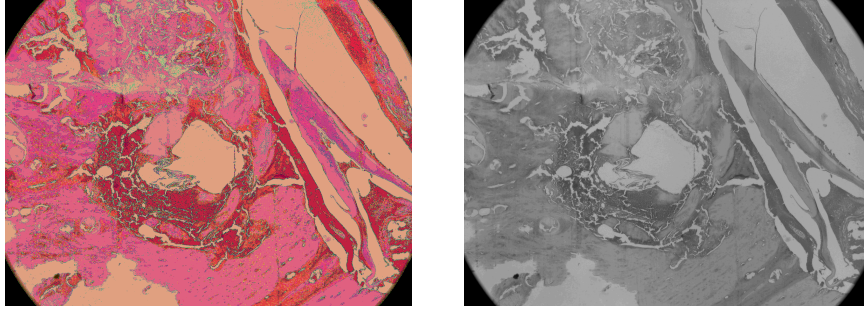
Şekil 3.8. Görüntü Analiz Programının Algoritması

3.3.1. Gri tona çevirme

Genellikle görüntü işlem algoritmalarında 8 bit gri tonlu görüntü üzerinde çalışılmaktadır. Bunun nedeni çalışma yapılacak görüntü aralığının azaltılmasıdır. Programda görüntü gri tonlu ise gri tona çevirmeye ihtiyaç yoktur. Görüntü renkli ise gri tona çevirme algoritması kullanılarak görüntü 8 bite çevrilir. Şekil 3.9.'da gri tona çevrilmiş görüntü yer almaktadır.

$$\text{Griderger} = \text{Round}(\text{R_degeri}(f(x,y)), \text{G_degeri}(f(x,y)), \text{B_degeri}(f(x,y)))/3;$$

$$f(x,y) = \text{RGB}(\text{Griderger});$$



Şekil 3.9. Gri tona çevirme işlemi

3.3.2. Referans Değerinin Belirlenmesi

Görüntü işleme ile yapılan çalışmalarda birbirinden çok farklı boyutlarda ve tiplerde resim kullanılmaktadır. Burada yapılacak ölçümler, kullanıcının belirteceği bir ölçüt dikkate alınarak yapılmaktadır. Aksi takdirde ölçüt değeri 1 olarak alınmaktadır. Ölçü birimi ve seçilen bölgenin hangi değere karşılık geldiği de önemlidir. Bu sorunları çözmek amacıyla Referans bölümü oluşturulmuştur. Referans, resim üzerinde belirli bir bölgeyi işaretleyerek bu aralığın karşılığını kullanıcıya sormaktadır. Aşağıdaki şekilde çizilen çizginin pixel olarak karşılığı verilmektedir ve bu değere karşılık gelecek ölçü birimi ve değer istemektedir. Şekil 3.10.'da görüldüğü üzere 29 pixel=5 Mikron karşılığını almaktadır. Bundan sonraki tüm hesaplamalar bu eşitlik üzerine yapılmaktadır.

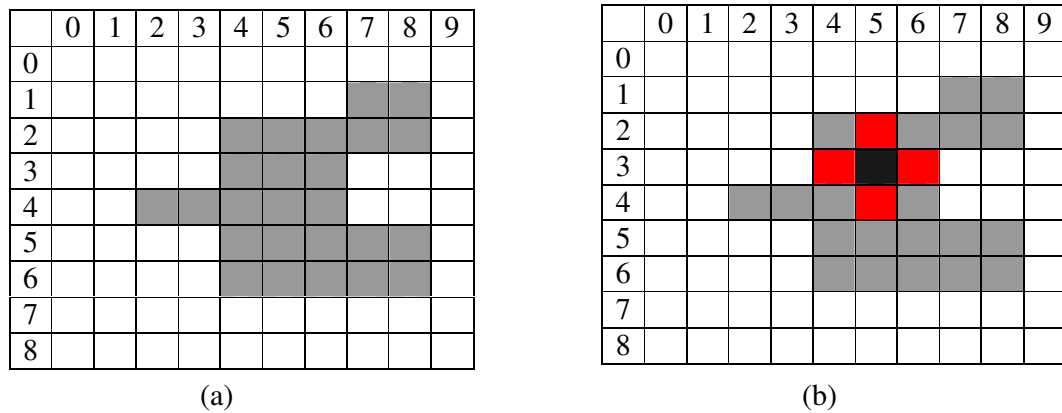


Şekil 3.10. Referans değerini belirleme (Örnek resmi ; Yücel M.T. 2005)

3.3.3.Noktaya Göre İşlem Yapma

Yayılma algoritması, görüntüde seçilecek bölge üzerinde herhangi bir noktayı işaretleme ile başlar. Bu noktanın koordinatları bir diziye aktarılmaktadır. Bu dizi içerisinde kontrol yapılacak olan noktaların koordinatları bulunmaktadır. Seçim sonucunda, işaretlenen noktanın $f(x,y)$ renk değerleri elde edilir.

Şekil 3.11.'deki görüntüde hesaplanması istenen bölge üzerinde herhangi bir nokta işaretlenir. Örnekte $f(5,3)$ noktasının renk değerleri alınır. Bu renk değerleri, işaretlenen noktanın çevresindeki dört komşu noktanın renk değerleri ile karşılaştırılır. Eğer belirli bir toleransa -uzman tarafından belirlenir- göre bir renk uyuşması tespit edilirse o nokta boyanır ve diziye eklenir. Bu karşılaştırma işlemi, dizideki tüm elemanlar kontrol edilene kadar sürer. Sonucu dizi elemanına gelindikten sonrada renklendirilen bölgedeki noktalar sayılır. Şekil 3.13.'de programın bu algoritma sonucu elde ettiği görüntüler yer almaktadır.



Şekil 3.11. Yayılma Algoritma Şeması

a)Örnek görüntü

b)Yayılma algoritmasının işleme görüntüsü

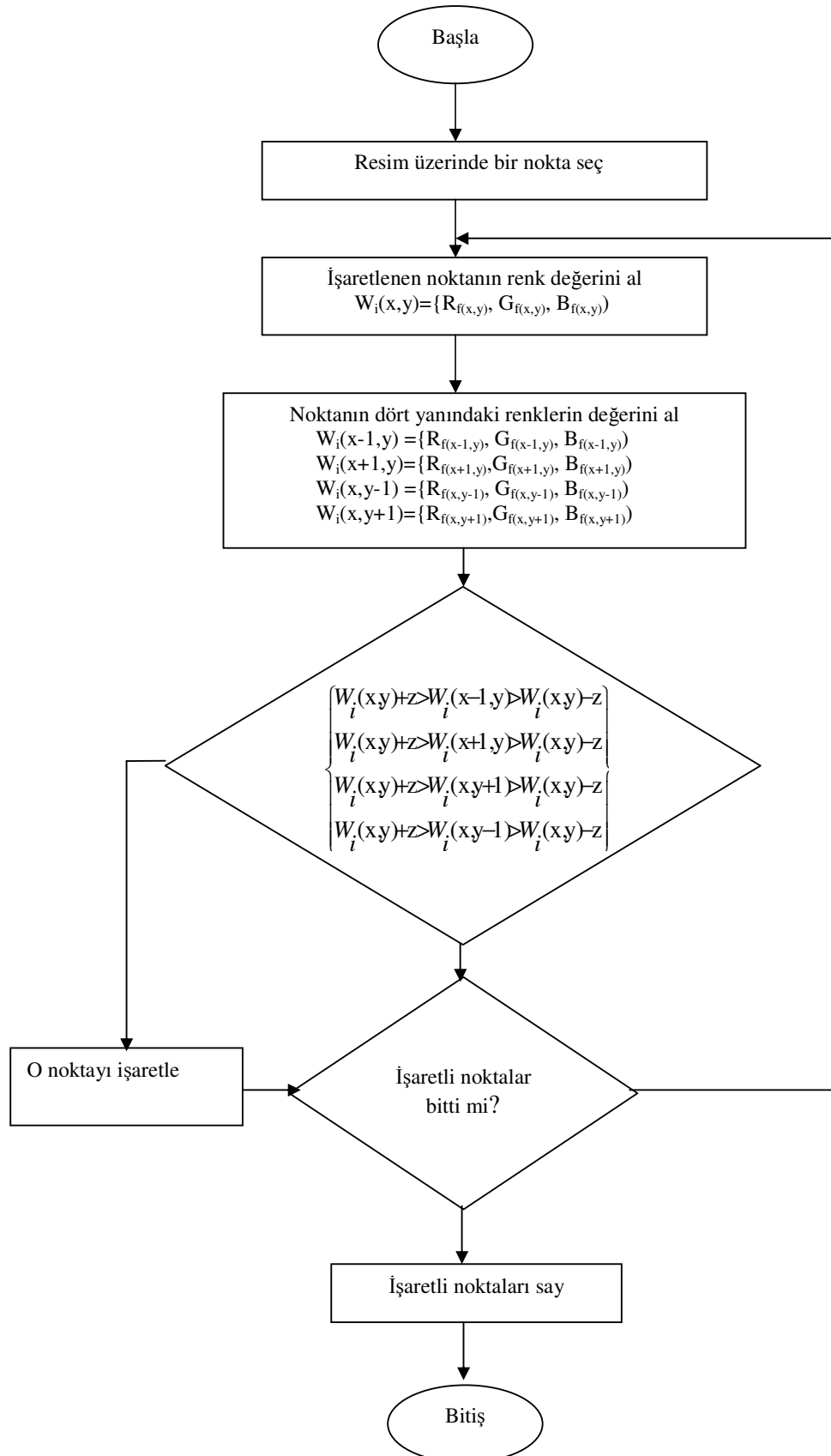
(x,y) koordinatlarındaki bir p pikselinin dört adet dikey ve yatay komşusu vardır. Bu komşuların koordinatları

$$(x+1,y), (x-1,y), (x,y+1), (x,y-1) \quad (7)$$

ile verilir. Bu pixel kümesine p 'nin 4 komşuluğu adı verilir ve $N_4(p)$ ile gösterilir.

Noktaya göre işlem yapma fonksiyonunda, görüntü üzerinde herhangi bir yere işaretlenir. Daha önceden ayarlanan tolerans değerine göre de görüntünün çevresine yayılma algoritması uygulanır.

Bu algoritma Şekil 3.12.'de gösterilmiştir.



Şekil 3.12. Noktaya göre işlem yapma algoritması

Adım 1. İşaretlenen noktanın renk değerini al.

$$f(x,y) = [f_{kırmızı}(x,y), f_{yeşil}(x,y), f_{mavi}(x,y)] \quad (8)$$

$$W_i(x,y) = f(x,y)$$

Adım 2. Noktanın $f(x,y)$ koordinatlarına göre çevresindeki $N_4(f)$ komşuluğundaki yerlerin renk değerleri al.

$$f(x-1,y) = [f_{kırmızı}(x-1,y), f_{yeşil}(x-1,y), f_{mavi}(x-1,y)] \quad (9)$$

$$f(x+1,y) = [f_{kırmızı}(x+1,y), f_{yeşil}(x+1,y), f_{mavi}(x+1,y)]$$

$$f(x,y-1) = [f_{kırmızı}(x,y-1), f_{yeşil}(x,y-1), f_{mavi}(x,y-1)]$$

$$f(x,y+1) = [f_{kırmızı}(x,y+1), f_{yeşil}(x,y+1), f_{mavi}(x,y+1)]$$

Adım 3. Eğer seçilen noktadaki renk değeri ile $N_4(f)$ komşuluğundaki renklerden birisi arasında (tolerans değeri de göz önüne alınarak) ortaklık varsa; o noktayı işaretle ve diziyeye aktar.

$$W_i(x,y) = \left\{ \begin{array}{l} 1, f(x,y) + z > f(x-1,y) > f(x,y) - z \\ 1, f(x,y) + z > f(x+1,y) > f(x,y) - z \\ 1, f(x,y) + z > f(x,y+1) > f(x,y) - z \\ 1, f(x,y) + z > f(x,y-1) > f(x,y) - z \\ 0, dışında \end{array} \right\} \quad (10)$$

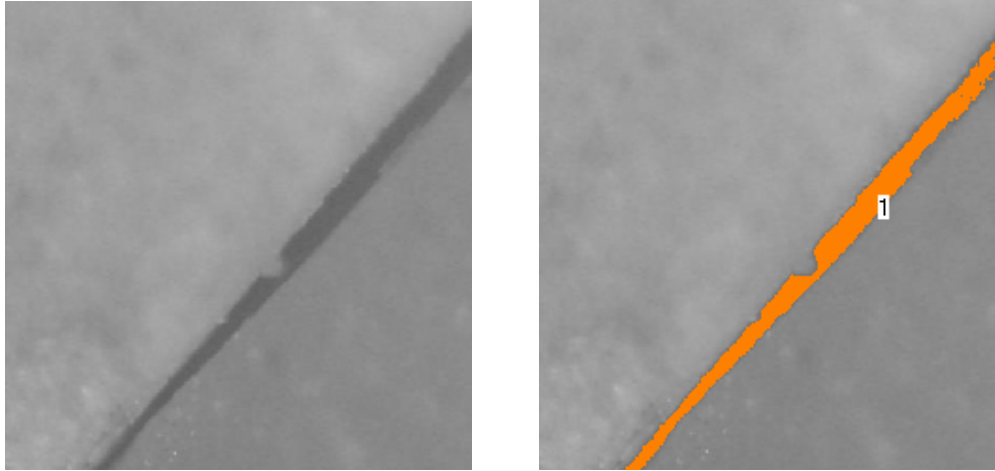
$W_i(x,y)$ = işaretlenen dizilerin görüntü fonksiyonu

z = tolerans değeri

Adım 4. Eğer dizideki kontrol edilen eleman sonuncu değilse Adım 2'ye git.

Adım 5. İşaretli noktaları say.

Adım 6. Dur.



Şekil 3.13. Noktaya göre işlem yapma sonucu elde edilen görüntüler

3.3.4.Çizime göre işlem yapma

Rastgele seçim algoritmasında, öncelikli olarak görüntü üzerinde hesaplanması istenen alan fare ile seçilerek işaretlenir. Daha sonra iç bölgede kalan alanın herhangi bir noktası işaretlenir ve noktanın koordinatları diziye eklenir. Şekil 3.14.'de, $f(7,5)$ koordinatı işaretlenmiştir. Bu noktanın renk değerleri alınarak komşu noktaların renk değerleri ile karşılaştırılır. Eğer kontrol edilen nokta çizginin rengine eşit değil ise nokta boyanır ve kontrol için diziye aktarılır. Aksi durumda noktalar işaretlenmez.

	0	1	2	3	4	5	6	7	8	9
0										
1				■	■	■	■	■		
2			■	■				■		
3			■				■	■		
4		■						■		
5		■							■	
6			■	■	■	■	■	■	■	■
7			■	■	■	■	■	■	■	
8										

(a)

	0	1	2	3	4	5	6	7	8	9
0										
1				■	■	■	■			
2			■	■				■		
3			■					■	■	
4		■						■	■	
5		■					■	■	■	
6			■	■	■	■	■	■	■	■
7			■	■	■	■	■	■	■	
8										

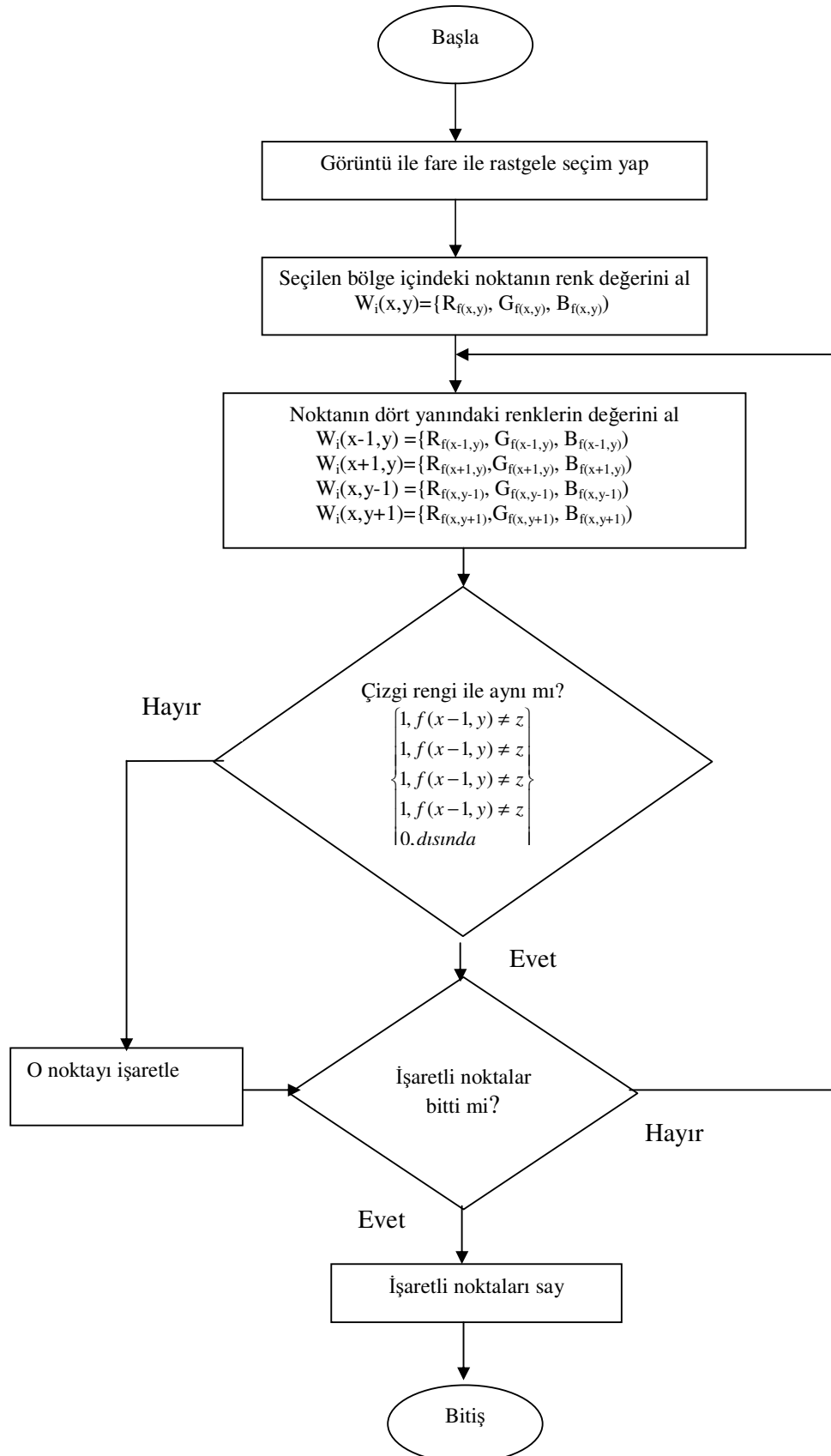
(b)

Şekil 3.14. Rastgele seçim şeması

a)Rastgele işaretlenmiş bölge

b)Rastgele algoritmasının işleme görüntüsü

Şekil 3.14'deki örnekte $f(6,5)$, $f(7,4)$, $f(8,5)$, $f(7,6)$ noktaları kontrol edilir ve $f(6,5)$, $f(7,6)$ noktaları işaretlenerek diziye aktarılır. Diğer noktalar ise çizgi rengine eşit olduğundan işaretlenmez. Bu şekilde kapalı alanda kalan tüm noktalar kontrol edilinceye kadar işlem devam eder. Son olarak işaretli noktalar sayılır. Uzman tarafından bu değerler tanı ve tedavi tespitinde kullanılır.



Şekil 3.15. Rastgele seçim işlemi algoritması

Adım 1. Görüntü üzerinde fare ile rastgele seçim işlemi yap.

Adım 2. İşaretlediğin bölge içerisinde herhangi bir noktayı seç ve o noktanın renk değerlerini al.

$$f(x,y) = [f_{kırmızı}(x,y), f_{yeşil}(x,y), f_{mavi}(x,y)] \quad (11)$$

$$W_i(x,y) = f(x,y)$$

Adım 3. Noktanın $f(x,y)$ koordinatlarına göre çevresindeki $N_4(f)$ komşuluğundaki noktaların renk değerlerini al.

$$f(x-1,y) = [f_{kırmızı}(x-1,y), f_{yeşil}(x-1,y), f_{mavi}(x-1,y)] \quad (12)$$

$$f(x+1,y) = [f_{kırmızı}(x+1,y), f_{yeşil}(x+1,y), f_{mavi}(x+1,y)]$$

$$f(x,y-1) = [f_{kırmızı}(x,y-1), f_{yeşil}(x,y-1), f_{mavi}(x,y-1)]$$

$$f(x,y+1) = [f_{kırmızı}(x,y+1), f_{yeşil}(x,y+1), f_{mavi}(x,y+1)]$$

Adım 4. Eğer noktaların renk değerleri seçili bölgedeki çizginin renginden farklı ise diziye ekle ve o bölgeyi işaretle.

$$W_i(x,y) = \begin{cases} 1, f(x-1,y) \neq z \\ 1, f(x-1,y) \neq z \\ 1, f(x-1,y) \neq z \\ 1, f(x-1,y) \neq z \\ 0, dı\u015fında \end{cases} \quad (13)$$

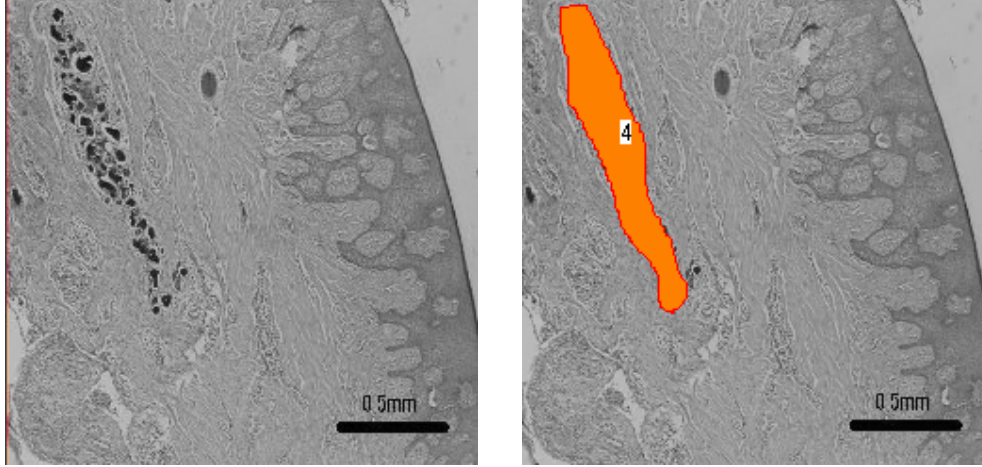
$W_i(x,y)$ = işaretlenen dizilerin görüntü fonksiyonu

z = Seçili bölgedeki çizginin renk değerleri.

Adım 5. Eğer dizide kontrol edilen eleman son eleman değilse Adım 3'e git.

Adım 6. İşaretli noktaları say.

Adım 7. Dur.

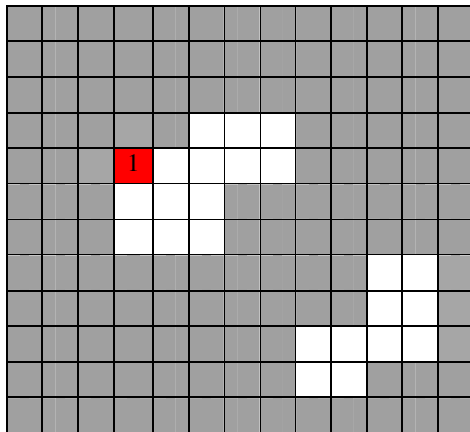


Şekil 3.16. Rastgele Seçim algoritması örnek görüntüler

3.3.5. Bölgeleri saydırma işlemi

Alınan görüntünün üzerindeki bölgelerin saydırılabilmesi için öncelikli olarak görüntünün gri moda olması gerekmektedir. Gri moddaki görüntü üzerinde ayrıntı bölgelerinde sayılması isteniyorsa direk olarak siyah beyaza çevrilmesi gerekmektedir. Eğer ayrıntılar istenmiyorsa median filtresi kullanılarak ayrıntılar giderilir. Son olarak da siyah beyaz görüntü üzerinde sayma işlemi gerçekleştirilir. İşlem sonunda beyaz olan noktaların sayısı bulunur.

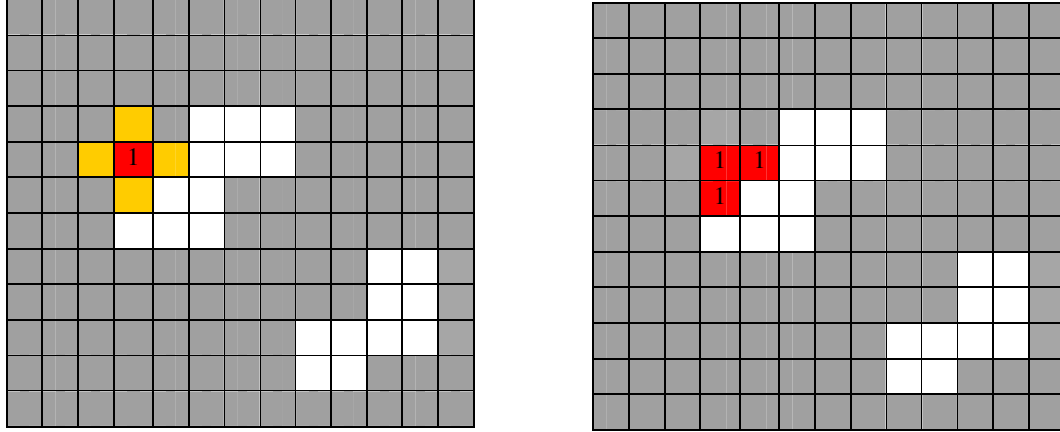
Adım 1. Görüntü üzerinde beyaz bir nokta buluncaya kadar ilerle.



Adım 2. Bu noktanın daha önce kontrol edilip edilmediğine bak.

Adım 3. Kontrol edilmemişse noktanın koordinatlarını diziye aktar ve noktaya sıradaki numarayı ver.

Adım 4. Şekil 3.17.'deki görüldüğü üzere noktanın 4 komşu değerlerine göz at. Bunlar arasından beyaz renge sahip olan varsa bunların da daha önce işaretlenmediğini kontrol et. Daha önce işaretlenmemiş ise diziye aktar ve numara ver.



Şekil 3.17. Bölge saydırma algoritmasının şeması

Adım 5. Dizideki kontrol edilen elamanlar bitinceye kadar tekrarla.

Adım 6. Numarayı 1 artır.

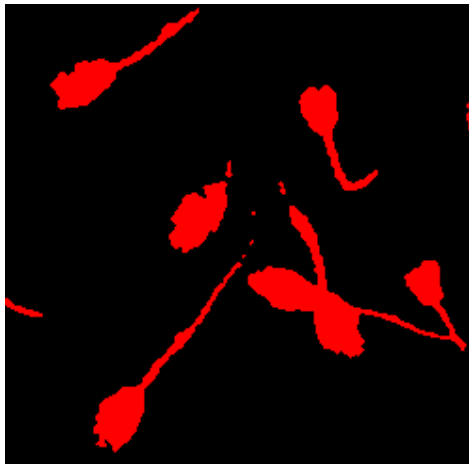
Saydırma işlemi görüntüleri aşağıda verilmiştir. Görüntü siyah beyaza çevrildikten sonra saydırma işlemi yapılmış ve belirli bir değer üzerinde olan bölge sayısı bulunmuştur.



(a)



(b)



(c)

Şekil 3.18. Bölge saydırma sonucu elde edilen görüntüler

- a) Sperm hücrelerinin ilk görüntüsü
- b) Sperm hücrelerinin Siyah-Beyaza çevrilmiş görüntü
- c) Sperm hücrelerinin bölgelerinin kırmızıya boyanmış görüntüsü

Şekil 3.18.'de bölge saydırma işlemi sonucu elde edilen görüntüler yer almaktadır. Bölge saydırma işlemi, birbirine benzeyen kaç parça olduğunu görüntü üzerinde belirlemektedir. Uzmanın belirleyeceği belirli bir değer altındaki bölgeler göz önüne alınmaz.

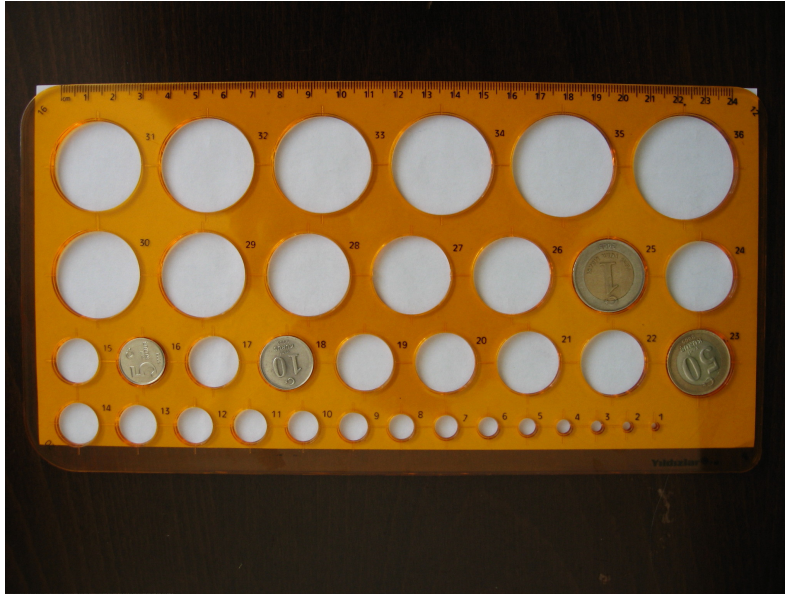
4.GERÇEKLEŞTİRİLEN SİSTEMİN PERFORMANS ANALİZİ

Bu çalışmada tıbbi örneklerden alınan görüntüler üzerinde analiz işlemleri uygulanmaya çalışılmıştır. Çalışmada; alan bulma, açı bulma, kenar bulma, bulanıklaştırma, keskinleştirme, alan saydırma, renkleri birbirine oranlama, negatifini alma, griye çevirme, renk skalasına göre boyama, şekil(dikdörtgen, daire, poligon, rastgele) kullanarak analiz işlemleri gerçekleştirilmektedir.

4.1. Alan Hesap Testi

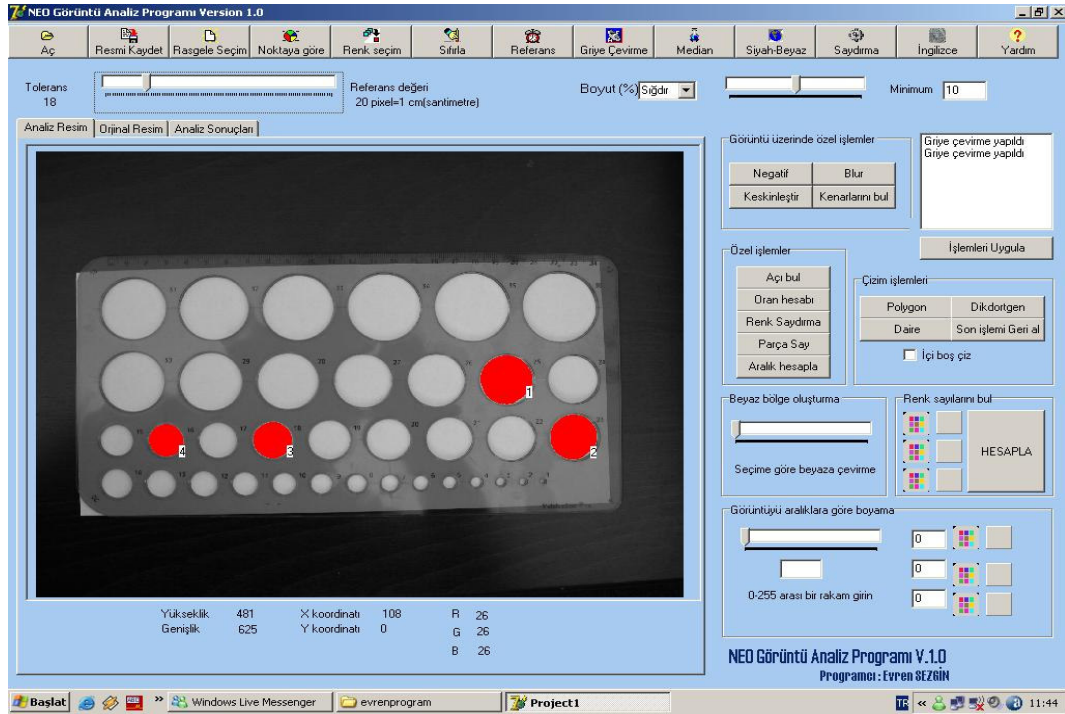
Programı test etmek amacıyla, 1 adet dairesel şablon, 1 adet iletke ve T.C. Merkez Bankasına ait 5 kuruş ,10 kuruş ,50 kuruş ve 1 YTL lik bozuk paralar kullanılmıştır. Görüntüler dijital fotoğraf makinesi ile çekilip, bilgisayara aktarılmıştır.

Şekil 4.1.'de paraların dairesel şablon üzerinde hangi bölgeye eşit olduğu belirtilmiştir.



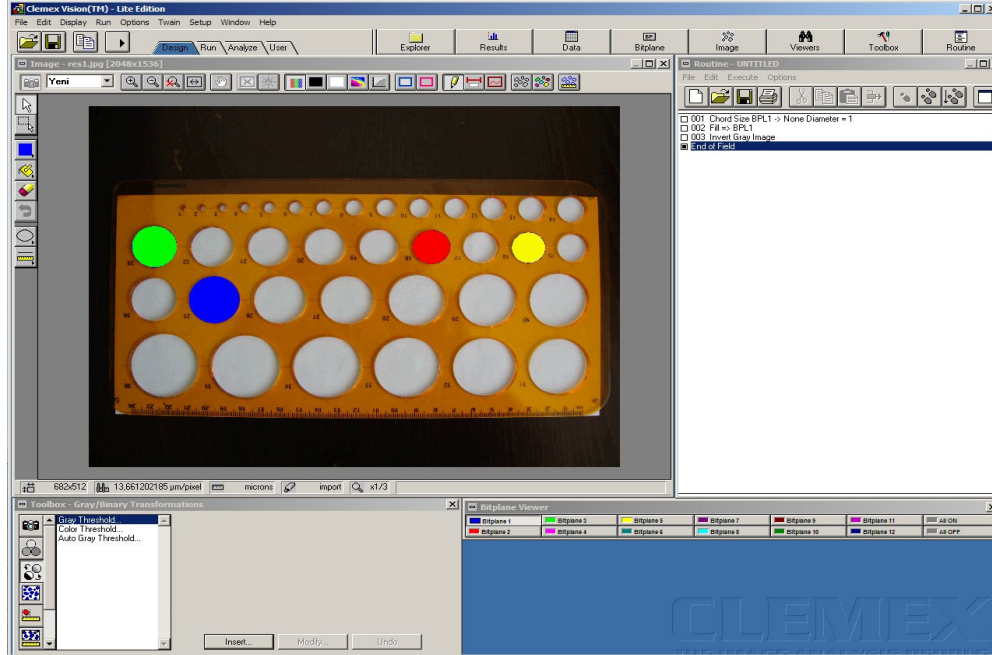
Şekil 4.1. Dairesel şablon üzerinde paraların alanının tespiti.

Bozuk paraların dairesel şablonda 16, 18, 23, 25 nolu bölgelere geldiği tespit edilmiş ve Şekil 4.2.'de alan hesaplaması yapılmıştır.



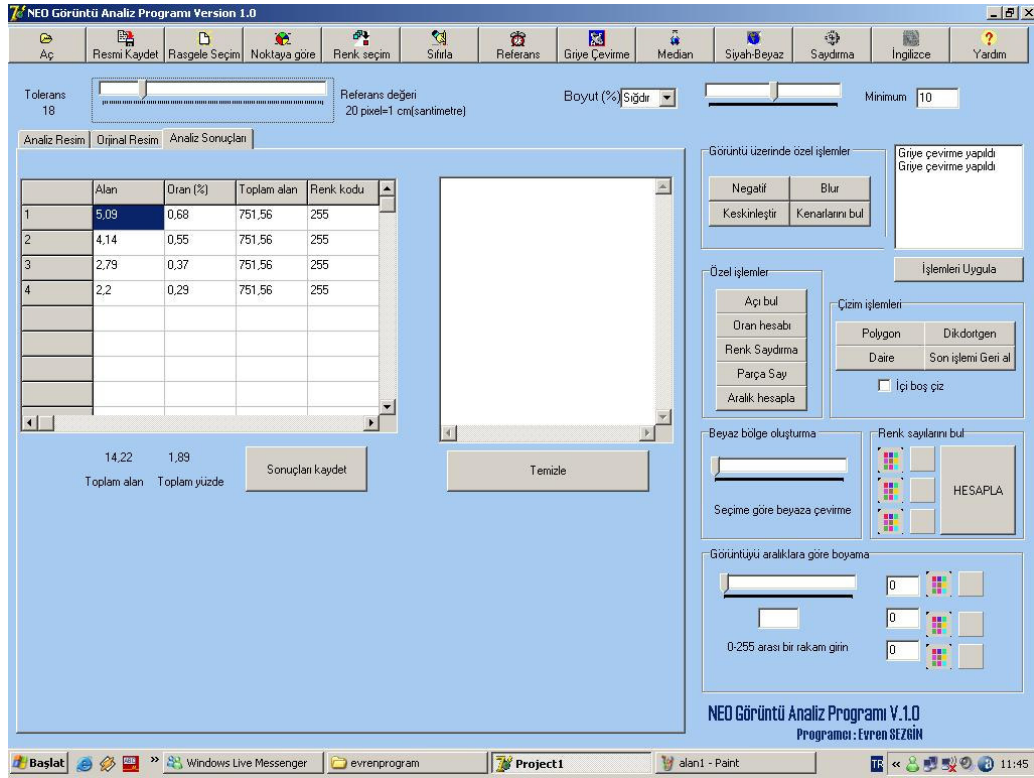
Şekil 4.2. Şablonda Alan Hesaplama Görüntüsü

Alan hesaplamasında ilk olarak referans değeri belirlenmektedir. Bu değer 1 cm'nin kaç pixele karşılık geldiğini belirlemektedir. Bu referans değeri dairesel şablonda bulunan cetvel üzerinden belirlenerek, noktaya göre alan hesaplama yöntemi kullanılmıştır. Alan hesaplamasında ; görüntünün çekim açısı, ortamdaki ışık ve belirlenecek referans değeri önem arz etmektedir. Bu parametrelerde değişikli olduğunda sapmalar meydana gelebilir.



Şekil 4.3. Clemex Vision Lite 3,5 Programı görüntüsü

Şekil 4.3.'de Clemex Vision Lite 3,5 (Canada) yazılımı tarafından elde edilen görüntü yer almaktadır. Bu programın sonuçları Çizelge 4.1.'de gösterilmektedir.

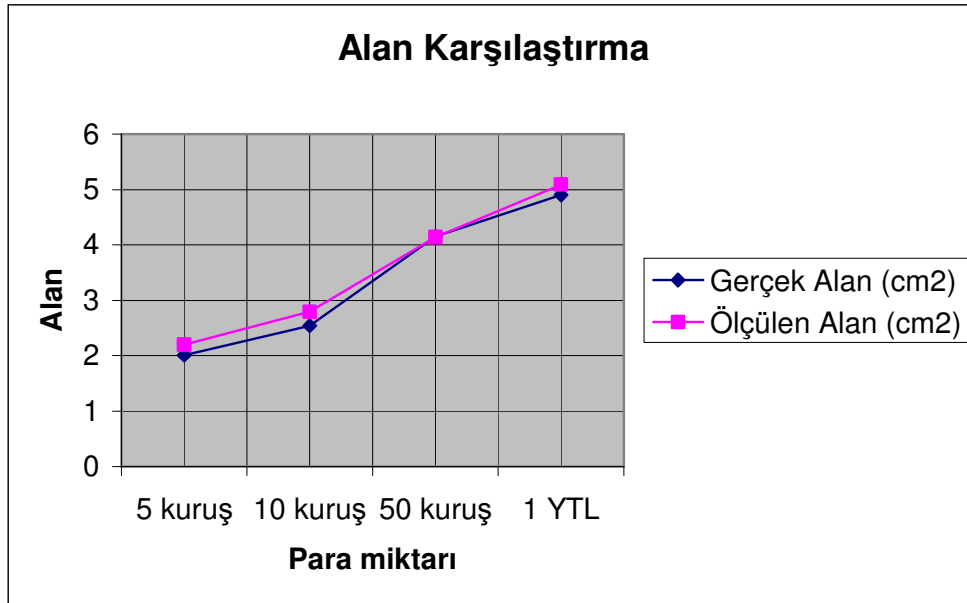


Şekil 4.4. Alan hesaplama sonuç bölümü

Şekil 4.4'de ise geliştirilen programın işlem sonuç görüntüleri yer almaktadır. Çizelge 4.1.'de ise bozuk paraların alanları ile programın ölçtüğü alan karşılaştırılması yapılmıştır. Çizelgeye göre, geliştirilen program %95'lik bir başarı oranı ile alanı tespit etmiştir. Clemex Vision Lite 3,5 programı ise %92,25'lik bir başarı oranı yakalamaktadır.

Çizelge 4.1. Programın alan karşılaştırması sonuçları

Para Birimi	Gerçek Alan (cm ²)	Ölçülen Alan (cm ²)	Başarı oranı (%)	Clemex Lite 3,5 (cm ²)	Vision Ölçüm	Başarı Oranı (%)
5 kuruş	2,01	2,2	92	2,2		92
10 kuruş	2,54	2,79	92	2,91		86
50 kuruş	4,15	4,14	99	4,23		98
1 YTL	4,90	5,09	97	5,24		93
		Ortalama Başarı Oran	95	Ortalama Başarı Oranı		92,25



Şekil 4.5. Karşılaştırılan Alanların Çizgisel Grafiği

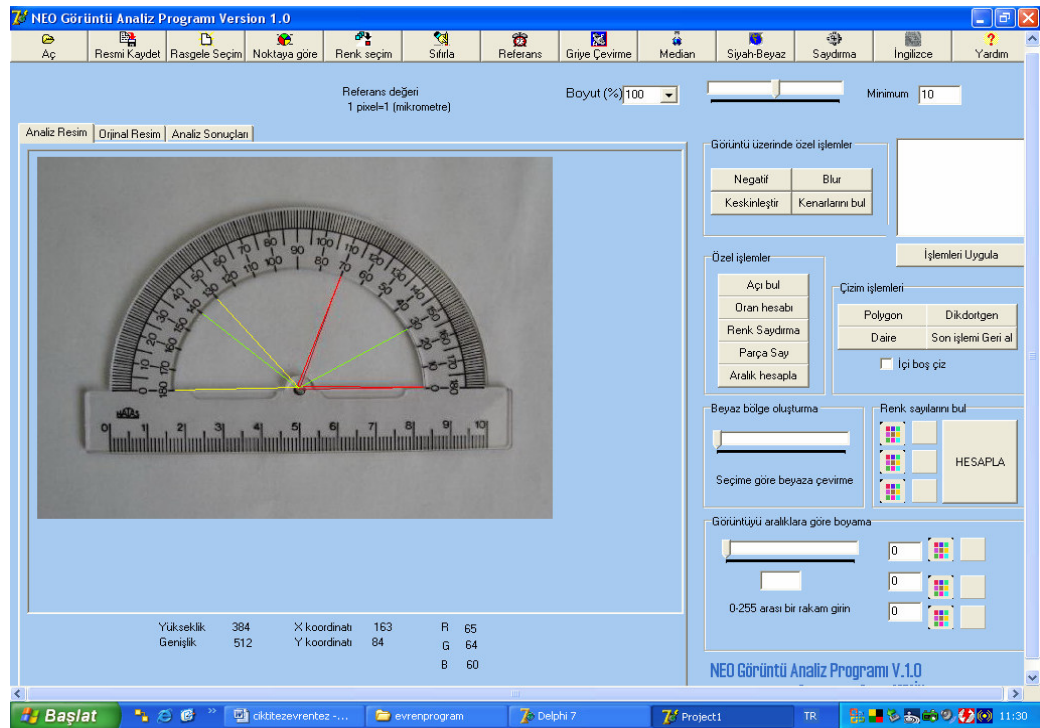
Şekil 4.5.' de ise bu ölçümler grafik halinde gösterilmektedir.

4.2. Açı Ölçüm Testi

Açı ölçüm testi için, beyaz bir kağıt üzerinde iletkinin görüntüsü alınmıştır. Şekil 4.6.'da, görüntü üzerinde açının ölçülmesi istenen noktalar belirlenmektedir. Çizelge 4.2'de ise program tarafından bulunan açı sonuçları gösterilmektedir. Program, yapılan denemeler sonucunda açı bulma işleminde %99.6'lık bir başarı sağlandığı görülmüştür. Clemex Vision Lite 3,5 programı açı ölçüm işlemi gerçekleştirmemektedir.

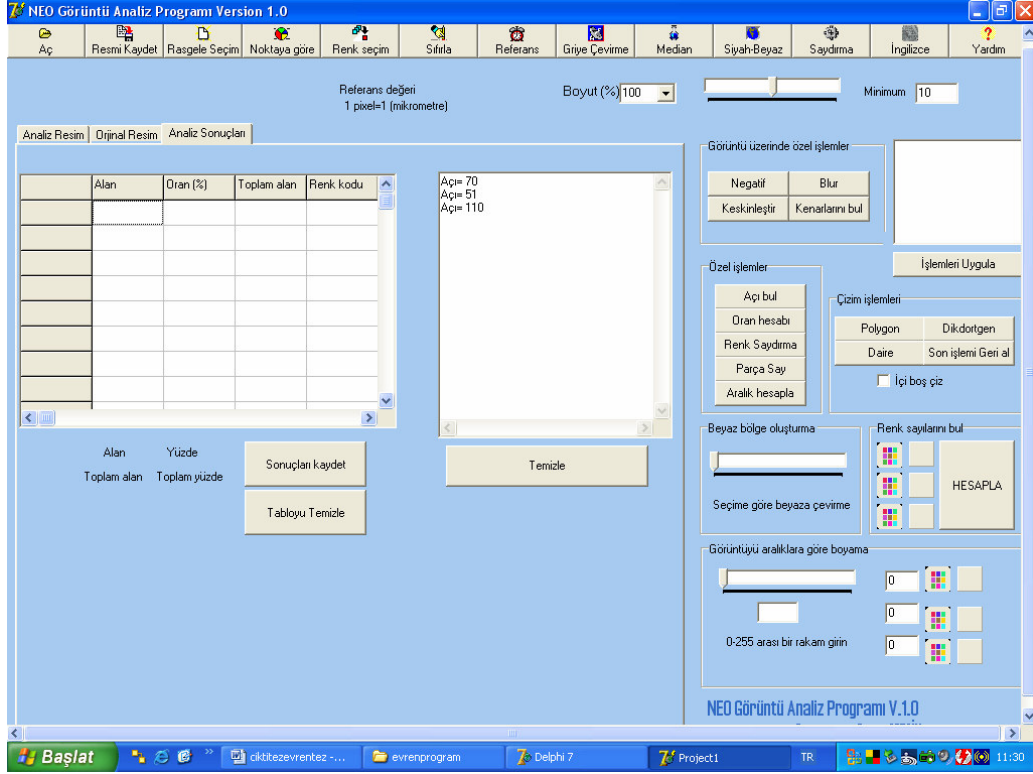
Çizelge 4.2. Açı ölçüm analiz sonuçları

Renkler	Ölçülmek istenen açı	Hesaplanan açı	Başarı Yüzdesi (%)
Yeşil	110	110	100
Sarı	50	51	99
Kırmızı	70	100	
Toplam Başarı (%)			99.6



Şekil 4.6. Açı ölçme görüntüsü

Şekil 4.7’de görüntü analiz sonucuna göre açıların aldığı değerler gösterilmektedir.



Şekil 4.7. Açı ölçme sonucu

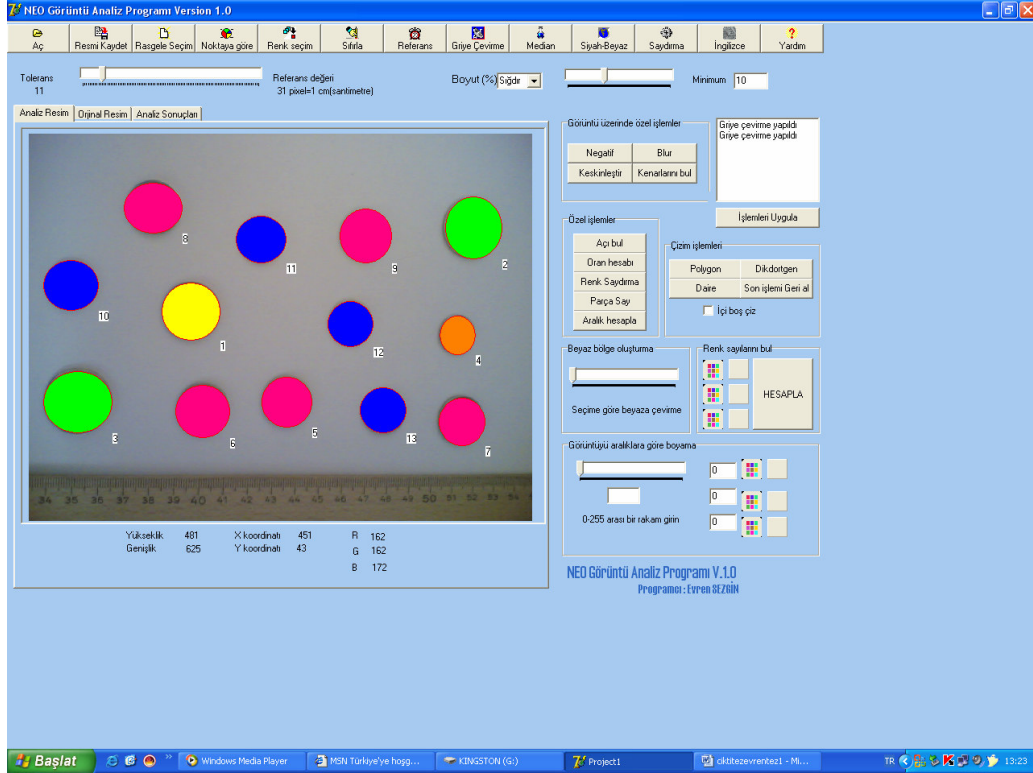
4.3.Parça Saydırma Testi

Parça saydırma işlemi için Şekil 4.8'deki görüntü test edilmektedir. Bu görüntüde iki rengin alanları ve kaç parça oldukları tespit edilmiştir. Bu işlem için, sayılması istenen bölgelerin en az kaç pixel içermesi gerektiği önceden belirlenmektedir. Testte en az 10 pixel içeren bölgelerin sayılması istenmektedir. Çizelge 4.3.'de sonuçlar gösterilmektedir. Sonuçlara göre aynı renkten toplam alanların hesaplanması işleminde %96.5' lük, parça saydırma işleminde de %100' lük bir başarı sağlandığı görülmüştür. Clemex Vision Lite 3,5 programı da parça saydırma işleminde %100'lük bir başarı elde etmektedir.

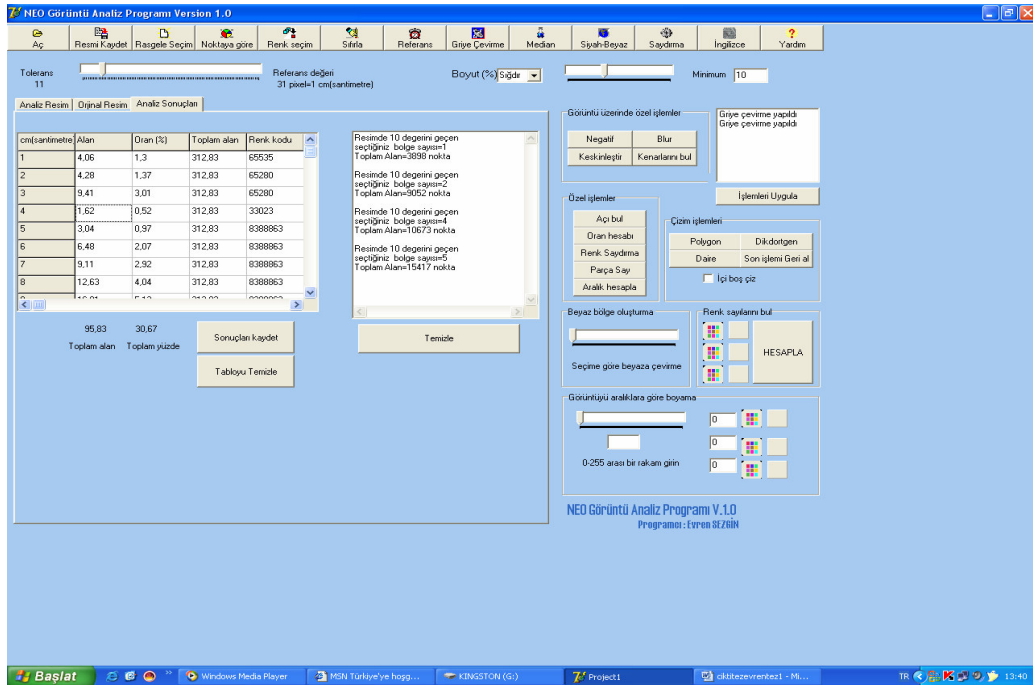
Şekil 4.9.'de seçilen rengin analiz sonucunda kaç parça olduğu ve kapladıkları toplam alanını gösteren ekran görüntüsü yer almaktadır.

Çizelge 4.3. Bölge saydırma sonuçları

Renkler	Toplam alan (cm²)	Hesaplanan alan (cm²)	Alan Başarı Yüzdesi (%)	Toplam Bölge Sayısı	Hesaplanan Bölge sayısı	Bölge Başarı Yüzdesi (%)
Sarı (50 kuruş)	4,15	4,06	97	1	1	100
Yeşil (1 Ytl)	9,80	9,41	96	2	2	100
Pembe (25 kuruş)	16,00	16,01	99	5	5	100
Mavi (10 kuruş)	10,16	10,92	94	4	4	100
	Toplam Başarı (%)		96.5	Toplam Başarı (%)		100



Şekil 4.8. Bölge sayısını bulma test örneği



Şekil 4.9. Bölge sayısını bulma sonucu

5.SONUÇ VE TARTIŞMA

Çalışmada, bilgisayarlı görüntü analizi sisteminin patoloji pratiğinde kullanılması ile çoğu sübjektif değerlendirmeler objektif değerlendirmeler haline gelmiştir. Bazı değerlendirmelerde de kantitatif değerlere ulaşılabilmiştir. Böylelikle kişiden kişiye değişebilen sonuçlar standardizasyon kazanmıştır. Net sayısal değerlerin verilmesi de sonuçların güvenilirliği artırmıştır. Görüntü analizi kullanılmadan önce yapılması mümkün olmayan veya yapılması çok zor görünen işlemler de bu sayede kolayca uygulanabilmiştir. Görüntü analizinin kullanımı zaman tasarrufu da sağlamış, güçlkle uzun sürede yapılabilen işlemler daha kısa sürede ve daha kolay gerçekleştirilmektedir.

Görüntü analizi ;bir görüntüdeki işlemlerin performansını belirlemek için kullanılır. Bu analiz teknikleri görüntü işlemenin geniş bir bölümünü oluşturur. Genellikle görüntü analizi ardışık adımlarla gerçekleştirilir ve bazı özel sorulara cevap bulmak için kullanılır

Görüntü analizi işlemi üç grupta incelenebilir. Bunlar:

- Ön-işlem (gürültüleri filtrelemek ve istenmeyen detayları elimine etmek)
- Veri azaltma
- Detay analizi

Bu çalışmada, tanı amacıyla hastalardan alınan doku örneklerinden hazırlanan preparatlar üzerinde değerlendirme yapmamızı sağlayan, “Görüntü Analizi Programı” geliştirilmiştir. Programda görüntünün netliği, çekildiği açı ve referansın belirlenmesi gibi önemli ayrıntıların sonuca tesir ettiği görülmektedir.

Dijital görüntü işleme parametreleri yeniden düzenlenerek ve sonucu ulaşacak değişik algoritmalar geliştirilebilir. Bu konuda patoloğların ve bilgisayar yazılımcılarının ortak çalışması gerekmektedir.

6.KAYNAKLAR

Ang KK, Berkey BA, Tu X, Zhang HZ, Katz R, Hammond EH, Fu KK, Milas L. 2002. Impact of epidermal growth factor receptor expression on survival and pattern of relapse in patients with advanced head and neck carcinoma. *Cancer Res* 15;62(24):7350-6

Avunduk,M.C. 2003, İmmunohistokimyasal ve Konvansiyonel Doku Boyamaların Bilgisayarlı Görüntü Analizi İle Değerlendirilmesi, Selçuk Üniversitesi Sağlık Bilimleri Enstitüsü Konya.

Babaoğlu, İ. 2004. Dijital Görüntü İşleme Teknikleri Kullanarak Üreme Hücrelerinin İncelenmesi, Selçuk Üniv. Fen Bilimleri Enstitüsü Konya.

Cantu, M., 2005. Mastering Borland Delphi 2005, Sybex, USA..

Canzonieri V, Carbone A. 1998. Clinical and biological applications of image analysis in non-Hodgkin's lymphomas. *Hematol Oncol* 16(1):15-28

Castelman, R. K.1996. Digital Image Processing, Prentice hall, Englewood Cliffs, New Jersey, USA.

Çakar S.,Öz C., Yurtay N. 2003, Video Görüntüsü Üzerinde Cisim Hareketlerini Saptama, İzleme ve Hedef Belirleme, III. Uluslar arası İleri Teknolojiler Sempozyumu, 18-20 Ağustos, Ankara.

Dulewicz A, Pietka BD, Jaszczak P, Nechay A, Sawicki W, Pykalo R, Kozminska E, Borkowski A.,2003. Computer identification of neoplastic urothelial nuclei from the bladder. *Anal Quant Cytol Histol* 23(5):321-9.

Erler BS, Chein K, Marchevsky AM. 1994. An image analysis workstation for the pathology laboratory. *Mod Pathol.* Jun;7(5):616-7.

Fernandez, G.,Kunt, M, Zryd J-P. 1995. A new plant of Cell Image Segmentation Algoritm, 8th International Conference in Image Analysis and Processing system, Italy.

Gümüřer, M. 2001. Dijital Görüntü İşleme, YTÜ Fen Bilimleri Enstitüsü, İstanbul.

Gökmen, H. 1990. Sayısal Görüntü işleme ile Genetik Hastalıkların Teşhisine Yardımcı bir Bilgisayar Sisteminin Tasarım ve Gerçekleştirimi, Ege Üniversitesi, İzmir.

Gonzalez, R.C.,Wintz P. 1983. Dijital Image Processing, Addison-Wesley Pub. Company.

<http://www.patoloji.gen.tr>

<http://www.goruntuisleme.org>

<http://www.efg2.com/Lab/>

<http://www.ins.itu.edu.tr/skulur/bildiri.ppt>

Jain, A.K. 1989. Fundamentals Of Digital Image Processing, Prentice Hall.

Jahne, B. 1997. Image Processing for Scientific Applications, USA.

Kumar V., Abbas A.K., Fausto N., 2004. Pathologic Basis of Disease,seventh ed. Saunders, Philadelphia, PA.

Mink D, Bonkhoff H, Herth G, Schmidt W. 1992. Use of computer-assisted image analysis for the detection of immunohistochemical receptor status in breast cancer. Immunohistochemical densitometry receptor analysis—IRDA. Geburtshilfe Frauenheilkd Oct;52(10):617-23

Mozdziak PE, Walsh TJ, McCoy DW.2002. The effect of early posthatch nutrition on satellite cell mitotic activity. *Poult Sci* 81(11):1703-8

Niblack W, 1986. *An introduction to Dijital Image Processing*, Prentice Hall Inc.

Oduncu H., Aslantaş V., Tunçkanat M.,Kurban R. 2005. Deri Üzerindeki Yaraların Sayısal Görüntü İşleme İle Analizi , IEEE 13. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 16-18 Mayıs.

Okamura K, Kobayashi I, Matsuo K, Kiyoshima T, Yamamoto K, Miyoshi A, Sakai H.1997. Immunohistochemical localization of cathepsin D, proliferating cell nuclear antigen and epidermal growth factor receptor in human breast carcinoma analysed by computer image analyser: correlation with histological grade and metastatic behaviour. *Histopathology* Dec;31(6):540-8

Parker,J.R. 1997. *Algorithms for Image Processing and computer Vision*, John Wiley&Sons.Inc.

Pratt, W.K. 1991. *Digital Image Processing*, A Wiley-Interscience Publication.

Russ, J.C. 1994. *The Image Processing Handbook*, Crc Pres.

Sarraj S, Maekawa S, Kibble M, Everall I, Leigh N. 2002. Ubiquitin-only intraneuronal inclusion in the substantia nigra is a characteristic feature of motor neurone disease with dementia. *Neuropathol Appl Neurobiol* 28(2):120-8.

Setton-Avruj CP, Aquino JB, Goedelman CJ, Soto EF, 2002. Villar MJ. P0 and myelin basic protein-like immunoreactivities following ligation of the sciatic nerve in the rat. *Neurochem Res* 27(11):1293-303

Spyridonos P, Ravazoula P, Cavouras D, Berberidis K, Nikiforidis G., 2001. Computer-based grading of haematoxylin-eosin stained tissue sections of urinary bladder carcinomas. *Med Inform Internet Med* 26(3):179-90.

Teuber, J., 1992. Digital Image Processing, Prentice Hall.

Yaman, K. ve Aktürk, N., 2001. Görüntü İşleme ile Kişi Yoğunluklarının Belirlenmesi, UMTS, Selçuk Üniversitesi, Konya, 12-14 Eylül.

Yaman,K., Sarucan A., Atak M, Aktürk N. 2001. Dinamik Çizelgeleme İçin Görüntü İşleme Ve Arıma Modelleri Yardımıyla Veri Hazırlama, J. Fac. Eng. Arch. Gazi Univ., Vol 16, No 1, 19-40.

Yaşar, E. 2005. Algoritma ve Delphi 2005, Ekin Kitabevi yayınları.

Yücel M.T. 2005. Farklı tip tam seramik kronların marjinal uyumlarının in vitro olarak değerlendirilmesi. Doktora Tezi, Selçuk Üniversitesi Sağlık Bilimleri Enstitüsü, Konya.

Umbaugh, S. 1988. Computer vision and image processing fundamentals, Computer Vision and Image Processing, Prentice Hall PTR, Bernard Goodwin.

Ülker S., 2003. Patoloji , İnkılap Kitapevi, İstanbul.

Zhang Y, Nojima S, Nakayama H, Jin Y, Enza H. 2003. Characteristics of normal stromal components and their correlation with cancer occurrence in human prostate. Oncol Rep 10(1):207-11

EKLER

Ek 1- Görüntü Analiz Programı Kodları

```

var
  Form1: TForm1;
  islem:byte;

yukse,genis,toplam,sut,sat,EskiX,EskiY,YeniX,YeniY,clicksay,cizim,referans,x1,y
1,x2,y2: integer;
  oran,olcu1,olcu2,olcu: real;
  BMP,ybitmap,bitmap:tbitmap;
  jpg:tjpegimage;
  cumle:string;
c,acideger,basla,polideger,ilknoktax,ilknoktay,ortanoktax,ortanoktay,sonnoktax,sonn
oktay,uzunluk:integer;
  aci,renkoran,renk1sec,renk2sec,renk3sec:boolean;
  tiklama,noktasayisi:integer;
  renk1,renk2:TColorRef;
  pxlsay,noktasay,aaa:integer;
  clickrenk,zitrenk :tcolor;
  ren,rboy : integer;
  parcasay,parcaadet:integer;
implementation

uses Unit2, Math;

{$R *.dfm }

procedure TForm1.ciz(xp,yp:integer);

begin
  if (Image1.Canvas.Pixels[xp,yp]=clickrenk) then
  begin
    inc(pxlsay);
    Image1.Canvas.Pixels[xp,yp]:=zitrenk;
    if (yp<rboy) then ciz(xp,yp+1);
    if (xp<ren) then ciz(xp+1,yp);
    if (yp>0) then ciz(xp,yp-1);
    if (xp>0) then ciz(xp-1,yp);
  end;
end;

procedure JPG2BMP(JPEGFile:String;BitmapFile:String);
Var

```

```

BMP : TBitmap;
Buffer : TImage;
Temp : Boolean;

begin
Buffer:=TImage.Create(Application);
with Buffer do
try
try
Picture.LoadFromFile(JPEGFile);
except
on EInvalidGraphic do
Exit;
end;
Temp := Picture.Graphic is TJPEGImage;
if Temp then
with TJPEGImage(Picture.Graphic) do
begin
Smoothing:=True;
end;
BMP:=TBitmap.Create;
with BMP do
try
Assign(Buffer.Picture.Graphic);
SaveToFile(BitmapFile);
savetofile('c:\islem.bmp');
finally
Free;
End;
finally
Free;
end;
end;

procedure TForm1.ToolButton3Click(Sender: TObject);
begin

if openpicturedialog1.execute then
begin
image1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
jpg2bmp(OpenPictureDialog1.FileName,'c:\test.bmp');
image2.picture.LoadFromFile('C:\test.bmp');
image1.Picture.LoadFromFile('c:\test.bmp');
//image1.picture.LoadFromFile(openpicturedialog1.filename);
//image1.picture.Bitmap.LoadFromFile

label24.Caption:=inttostr(image1.Height);

```

```
label25.Caption:=inttostr(image1.Width);
//yukse:=image1.picture.height;
//genis:=image1.picture.width;
end;
basla:=1;
end;

procedure TForm1.ToolButton2Click(Sender: TObject);
begin
SavePictureDialog1.Execute;
image1.Picture.SaveToFile(SavePictureDialog1.FileName);

end;

procedure TForm1.ToolButton1Click(Sender: TObject);
begin
islem:=1;
TrackBar1.Visible:=False;
label1.Visible:=false;
end;

procedure TForm1.ToolButton7Click(Sender: TObject);
begin
label1.Visible:=true;
label29.Visible:=true;
TrackBar1.Visible:=true;
islem:=2;
end;

procedure TForm1.ToolButton4Click(Sender: TObject);
begin
colordialog1.execute;

end;

procedure TForm1.ToolButton5Click(Sender: TObject);
begin
ListBox1.Items.add('Griye çevirme yapıldı');
griyecevir(image1.Picture.Bitmap);
end;

procedure tform1.griyecevir(yeni:tbitmap);//griye çevirme fonksiyonu
var
a,b:integer;
gray:byte;
thergbvalue:tcolorref;
begin
a:=0;b:=0;
```

```

while (a<=yeni.Width ) do begin
  while (b<=yeni.Height) do begin
TheRgbValue := ColorTorgb(colortorgb(yeni.Canvas.Pixels[a,b]));
Gray := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
yeni.Canvas.Pixels[a,b]:= (RGB(Gray, Gray, Gray));
  b:=b+1;
  end;
  a:=a+1;
  b:=0;
  end;
yeni.SaveToFile('c:\islem.bmp');
end;

```

```

procedure TForm1.ToolButton8Click(Sender: TObject);
var
  i,j:integer;
begin
image1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
jpg2bmp(OpenPictureDialog1.FileName,'c:\test.bmp');
image1.Picture.LoadFromFile('c:\test.bmp');
islem:=0;noktasayisi:=0;
toplam:=0;
for i:=0 to 10 do
for j:=0 to 10 do
stringgrid1.Cells[i,j]:";

sat:=0;sut:=0;
stringgrid1.Cells[1,0]:='Alan';
stringgrid1.Cells[2,0]:='Oran (%)';
stringgrid1.Cells[3,0]:='Toplam alan';
stringgrid1.Cells[4,0]:='Renk kodu';
oran:=0; Memo1.Clear;
label24.Caption:=inttostr(image1.Height);
label25.Caption:=inttostr(image1.Width);
end;

```

```

procedure TForm1.ToolButton9Click(Sender: TObject);
begin
referans:=1;
end;

```

```

procedure TForm1.FormCreate(Sender: TObject);

```

```

begin
aaa:=0;PageControl1.Pages[0].Show;
parcasay:=0;parcaadet:=0;
clicksay:=0;basla:=0;noktasayisi:=0;
cizim:=0;sut:=0;sat:=0;
acideger:=0;aci:=false;
stringgrid1.Cells[1,0]='Alan';
stringgrid1.Cells[2,0]='Oran (%)';
stringgrid1.Cells[3,0]='Toplam alan';
stringgrid1.Cells[4,0]='Renk kodu';

toplam:=0;oran:=0;
referans:=0;olcu:=1; pxlsay:=0;
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var
ind,ix,iy:integer;
TheRgbValue : TColorRef;
degerr,degerb,degerg:byte;
aci1,aci2:extended;
deger1,deger2,i,o_toplam_renk:integer;
aci1bolge,aci2bolge:string;
begin
c:=0;//sekilcizme dikdortgen veya daire dursun

if (cizim=3) then begin//poligon çizme
if (polideger=1) then begin
ortanoktax:=x;
ortanoktay:=y;
image1.Canvas.LineTo(ortanoktax,ortanoktay);
image1.Canvas.MoveTo(ortanoktax,ortanoktay);
end;
end;

//acı hesaplama
if (referans=1) then
begin
x2:=x;y2:=y;
end;
olcu:=olcu/strtofloat(form2.Edit1.text);//secilen degerin form2 deki edite gore
degeri
label6.Caption:=floattostr(olcu);

```

```

if (islem=1) then begin //rastgele secim
  Image1.Canvas.LineTo(EskiX,EskiY);
  clicksay:=clicksay+1;//secme degerinden kurtarmak için
end;

if (polideger=3) then begin
  polideger:=0;cizim:=1;islem:=1;
end;

if (polideger=2) then begin//poligonda sectirme içeriği
  polideger:=polideger+1; //burada bekletme yapıyorum
end;

//noktaya göre işlem yap
if (cizim=1) or (cizim=2) then begin
  clicksay:=0;
// screen.cursor:=crhourglass;
  bitmap:=tbitmap.create;
  bitmap.assign(image1.Picture.Bitmap); //image1.picture.bitmap
// (assign=belirlemek,saptamak)

  ind:=-1;
  if ind<0 then
    ind:=0;
  if (islem=1) then

doldur(bitmap,x,y,colordialog1.color,bitmap.canvas.pixels[x,y],trackbar1.Position/100,ind,0.1,0.5)
  else if (islem=2)then

doldur1(bitmap,x,y,colordialog1.color,bitmap.canvas.pixels[x,y],trackbar1.Position/100,ind,0.1,0.5);
//edit1.Text:=inttostr(bit.Width*bit.Height);
  islem:=0;
  image1.picture.bitmap:=bitmap;
  iy:=0;ix:=0;//renkli alanı buldurma

//bu bölümde daha önce aynı renkten olanların toplamını bulduruyorum
o_toplam_renk:=0;
for i:=1 to sat do begin
  if (floattostr(colordialog1.Color)=stringgrid1.Cells[4,i]) then
    o_toplam_renk:=o_toplam_renk+strtoint(stringgrid1.Cells[1,i]);
end;

noktasayisi:=0;

```

```

while (iy<image1.Height) do begin
  while (ix<image1.Width) do begin
    TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[ix,iy]));
    degerr:=getrvalue(ColorDialog1.Color);//bu bölümm yeni renge gore sayma
işlemi yapar
    degerg:=getgvalue(ColorDialog1.Color);
    degerb:=getbvalue(ColorDialog1.Color);
    if ((GetRValue(TheRGBValue)=degerr) and (GetGValue(TheRGBValue)=degerg)
and (GetBValue(theRGBvalue )=degerb)) then
      begin
        noktasayisi:=noktasayisi+1;
        end; //boyabasla if
        ix:=ix+1;
        end;//while end genişlik
        iy:=iy+1;
        ix:=0;
        end;//while end yükseklik
        //-----sırano-----alan----oran----toplam alan-----

        sat:=sat+1;
        stringgrid1.Cells[0,sat]:=inttostr(sat);
        stringgrid1.Cells[1,sat]:=floattostr(round(abs((noktasayisi-
o_toplam_renk)/round(olcu))));
        stringgrid1.Cells[2,sat]:=floattostr(roundto(abs((noktasayisi-
o_toplam_renk)/(bitmap.Width*bitmap.Height)), -2)*100);

        stringgrid1.Cells[3,sat]:=floattostr(round((bitmap.Width*bitmap.Height)/round(olcu)
));
        stringgrid1.Cells[4,sat]:=floattostr(colordialog1.color);
        image1.Canvas.TextOut(x,y,inttostr(sat));

        oran:=oran+abs((noktasayisi-toplam)/(bitmap.Width*bitmap.Height));

        label11.Caption:=floattostr(roundto(noktasayisi/olcu,-2));
        label12.Caption:=floattostr(roundto(oran,-2)*100);

//burada sonuçları yazdırma ve seklin numarasını verme
//işlemini gerçekleştiriyorum.
        toplam:=noktasayisi;
        bitmap.free;
        //screen.cursor:=crdefault;
        cizim:=0;
        end; //if cizim =1 in end i

        end;

```

```
procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
var
```

```
  TheRgbValue : TColorRef;
```

```
begin
```

```
  //bu satırlar dikdortgen ve elips çizme amacı ile yazıldı.
```

```
  if (c=1) then begin
```

```
    image1.Picture.Bitmap:=bmp;
```

```
    if (checkbox1.Checked=true) then
```

```
      image1.Canvas.Brush.Style:=bsclear
```

```
    else
```

```
      image1.Canvas.Brush.Color:=ColorDialog1.Color;
```

```
    image1.Canvas.Pen.Color:=clred;
```

```
    if (cizim=1)then
```

```
      image1.Canvas.Rectangle(ilknoktax,ilknoktay,x,y);
```

```
    if (cizim=2) then
```

```
      image1.Canvas.Ellipse(ilknoktax,ilknoktay,x,y);
```

```
  end;
```

```
if (referans=0) then
```

```
  begin
```

```
    if (clicksay=1)then Image1.Canvas.LineTo(X,Y);
```

```
  end
```

```
else
```

```
  begin
```

```
    x2:=x;y2:=y;
```

```
  end;
```

```
if (basla=1) then begin
```

```
  TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[x,y]));
```

```
  label15.Caption:=inttostr(getrvalue(thergbvalue));
```

```
  label17.Caption:=inttostr(getgvalue(thergbvalue));
```

```
  label19.Caption:=inttostr(getbvalue(thergbvalue));
```

```
  label7.Caption:=inttostr(Y);
```

```
  label9.Caption:=inttostr(X);
```

```
end;
```

```
end;
```

```
procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
```

```
var
```

```
  thergbvalue:TColorRef;
```



```
ix,iy,renk1sayisi,renk2sayisi,renk1r,renk1g,renk1b,renk2r,renk2g,renk2b:integer;
ab,bc,ac,x3,y3 : integer;
aci1 : real;
```

```
begin
// renk saydırmak için renk alma
clickrenk := Image1.Canvas.Pixels[x,y];

//parcasaydırma
if (parcasay=1) then begin
image1.Canvas.Brush.color:=ColorDialog1.Color;
image1.Canvas.Rectangle(x,y,x+4,y+4);
parcaadet:=parcaadet+1;
end;

//aci
if (aaa=1) then begin
image1.canvas.Pen.Color:=clred;
noktasay:=succ(noktasay);
if (noktasay>3) then noktasay:=1;
if (noktasay=1) then
begin
//Birinci nokta yani A noktası
//Ekranı temizle
x1:=x;
y1:=y;
image1.Canvas.MoveTo(x1,y1);
//label2.Caption:='A'+#13+inttostr(x1)+' '+inttostr(y1);
end;
if (noktasay=2) then
begin
//İkinci nokta yani B noktası
x2:=x;
y2:=y;
image1.Canvas.LineTo(x2,y2);
//label3.Caption:='B'+#13+inttostr(x2)+' '+inttostr(y2);
end;
if (noktasay=3) then
begin
//Üçüncü nokta yani C noktası
x3:=x;
y3:=y;
image1.Canvas.LineTo(x3,y3);
// label4.Caption:='C'+#13+inttostr(x3)+' '+inttostr(y3);
//AB doğrusu ile BC doğrusu arasındaki açıyı bulmak için
//kullandığımız formül
ab:=sqr(x1-x2)+sqr(y1-y2);
bc:=sqr(x2-x3)+sqr(y2-y3);
```

```

ac:=sqr(x1-x3)+sqr(y1-y3);
aci11:=radtodeg(arccos((ab+bc-ac)/(2*sqr(ab)*sqr(bc))));
memo1.lines.add('Açı= '+ floattostr(round(aci11)));
aaa:=0;
end;
end;

//renklerin birbirine oranı
if (renkoran=true) then begin
if (tiklama=1) then begin //oranda birinci rengin deęrleri
renk1:= ColorTorgb(colortorgb(image1.Canvas.Pixels[x,y]));
end;
if (tiklama=2) then begin
tiklama:=tiklama+1;
renk2:= ColorTorgb(colortorgb(image1.Canvas.Pixels[x,y]));
iy:=0;ix:=0;//renkli alanı buldurma
renk1sayisi:=0;renk2sayisi:=0;
while (iy<image1.Height) do begin
while (ix<image1.Width) do begin
TheRgbValue := colortorgb(image1.Canvas.Pixels[ix,iy]);
renk1r:=getrvalue(renk1);//bu bölüm yeni renge göre sayma işlemi yapar
renk1g:=getgvalue(renk1);
renk1b:=getbvalue(renk1);
if ((GetRValue(TheRGBValue)=renk1r) and (GetGValue(Thergbvalue)=renk1g)
and (GetBValue(theRGBvalue )=renk1b)) then
begin
renk1sayisi:=renk1sayisi+1;
end;
//ikinci renk
renk2r:=getrvalue(renk2);//bu bölüm yeni renge göre sayma işlemi yapar
renk2g:=getgvalue(renk2);
renk2b:=getbvalue(renk2);
if ((GetRValue(TheRGBValue)=renk2r) and (GetGValue(Thergbvalue)=renk2g)
and (GetBValue(theRGBvalue )=renk2b)) then
begin
renk2sayisi:=renk2sayisi+1;
end; //boyabasla if
ix:=ix+1;
end;//while end genişlik
iy:=iy+1;
ix:=0;
end;//while end yükseklik
memo1.lines.add('Birinci Rengin sayısı='+floattostr(renk1sayisi));
Memo1.lines.add('İkinci Rengin sayısı='+floattostr(renk2sayisi));
Memo1.lines.add('Birinci rengin seçilen toplam renge
oranı='+floattostr(roundto(renk1sayisi/(renk1sayisi+renk2sayisi),-3)));
Memo1.lines.add('İkinci rengin seçilen toplam renge
oranı='+floattostr(roundto(renk2sayisi/(renk1sayisi+renk2sayisi),-3)));

```

```

end;//tıklama2 endi
tiklama:=tiklama+1;
end;//oran hesabı bitti.

//dikdortgen ciziminde kullanılır
if (cizim=1)or (cizim=2)then begin
c:=1;
ilknoktax:=x;
ilknoktay:=Y;
end;
//poligonsa
if (cizim=3) then begin
  if (polideger=0) then begin
    ilknoktax:=x;
    ilknoktay:=y;
    image1.Canvas.Pen.Color:=clred;
    image1.Canvas.MoveTo(ilknoktax,ilknoktay);
    polideger:=polideger+1;
  end;
end;

if (islem=1) then begin//rasgele secim seçilmiş ise
  clicksay:=clicksay+1;
  EskiX:=X;
  EskiY:=Y;
  Image1.Canvas.MoveTo(EskiX,EskiY);
  image1.canvas.pen.Color:=clred;
end;

if (clicksay=3) then
  begin
    cizim:=1;clicksay:=1;
  end;

if (islem=2) then cizim:=1;//noktaya göre

if (referans=1) then//refea
  begin
    X1:=X;
    Y1:=Y;
    Image1.Canvas.MoveTo(X1,Y1);
  end;

end;

procedure TForm1.TrackBar1Change(Sender: TObject);

```

```

begin
label1.Caption:=inttostr(TrackBar1.position);
end;

procedure TForm1.Image1Click(Sender: TObject);
begin
if (referans=1) then begin//referans degerini belirleme
image1.Canvas.MoveTo(x1,y1);{ serbest cizim }
image1.Canvas.Pen.Color:=clred;
image1.Canvas.LineTo(X2,Y2);
olcu1:=abs(x2-x1);
olcu2:=abs(y2-y1);
form2.label5.caption:='(+floattostr(x1)+'+floattostr(y1)+'');
form2.label6.caption:='(+floattostr(x2)+'+floattostr(y2)+'');
olcu:=round(abs(sqrt((olcu1*olcu1)+(olcu2*olcu2))));
form2.show;
form2.label2.caption:=floattostr(olcu);
end;
referans:=0;
end;

procedure TForm1.TrackBar2Change(Sender: TObject);

var
c,d,birincirenk,ikincirenk,ix,iy,noktasayisi,ucuncurenk:integer;
TheRgbValue : TColorRef;
begin
image1.Picture.LoadFromFile('c:\islem.bmp');

bitmap:=tbitmap.create;
bitmap.assign(image1.Picture.Bitmap);

edit4.Text:=floattostr(trackbar2.position);
birincirenk:=strtoint(edit1.Text);
ikincirenk:= strtoint(edit2.Text);
ucuncurenk:= strtoint(edit3.Text);
c:=0;d:=0;
while (c<Image1.Width ) do begin
  while (d<Image1.Height) do begin
TheRgbValue := ColorToRGB(colortorgb(image1.Canvas.Pixels[c,d]));
if
((GetRValue(TheRGBValue)<trackbar2.position)or(GetGValue(TheRGBValue)<trackbar2.position)or(GetBValue(TheRGBValue)<trackbar2.position))
then begin
  if
((GetRValue(TheRGBValue)<birincirenk)or(GetGValue(TheRGBValue)<birincirenk)or(GetBValue(TheRGBValue)<birincirenk))then
    image1.Canvas.Pixels[c,d]:=colordialog2.Color

```

```

        else if
((GetRValue(TheRGBValue)<ikincirenk)or(GetGValue(TheRGBValue)<ikincirenk)
or(GetBValue(TheRGBValue)<ikincirenk)) then
            image1.Canvas.Pixels[c,d]:=colordialog3.Color
        else if
((GetRValue(TheRGBValue)<ucuncurenk)or(GetGValue(TheRGBValue)<ucuncurenk)
or(GetBValue(TheRGBValue)<ucuncurenk)) then
            image1.Canvas.Pixels[c,d]:=colordialog4.Color
        else
            image1.Canvas.Pixels[c,d]:=clblue;
        end;
        d:=d+1;
    end;
    c:=c+1;
    d:=0;
    end;
noktasayisi:=0;toplam:=0;ix:=0;iy:=0;sat:=0;
while (iy<image1.Height) do begin
    while (ix<image1.Width) do begin

        TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[ix,iy]));
        {   degerr:=getrvalue(ColorDialog1.Color);//bu bölümm yeni renge gore sayma
işlemi yapar
        degerg:=getgvalue(ColorDialog1.Color);
        degerb:=getbvalue(ColorDialog1.Color);
        }
        if ((GetRValue(TheRGBValue)=0) and (GetGValue(TheRGBValue)=0) and
(GetBValue(theRGBValue )=255)) then
            begin
                noktasayisi:=noktasayisi+1;
                end; //boyabasla if
                ix:=ix+1;
                end;//while end genişlik
                iy:=iy+1;
                ix:=0;
                end;//while end yükseklik
                //-----sırano-----alan----oran----toplam alan-----
                sat:=sat+1;
                stringgrid1.Cells[0,sat]:=inttostr(sat);
                stringgrid1.Cells[1,sat]:=floattostr(round(abs(noktasayisi-toplam)/round(olcu)));
                stringgrid1.Cells[2,sat]:=floattostr(abs((noktasayisi-
toplam)/(bitmap.Width*bitmap.Height)));

                stringgrid1.Cells[3,sat]:=floattostr(round((bitmap.Width*bitmap.Height)/round(olcu
)));
                oran:=oran+abs((noktasayisi-toplam)/(bitmap.Width*bitmap.Height));
                //burada sonuçları yazdırma ve seklin numarasını verme
                //işlemini gerçekleştiriyorum.

```

```

toplam:=noktasayisi;
  bitmap.free;
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
ColorDialog2.Execute;
panel1.Color:=colordialog2.Color;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
ColorDialog3.Execute;
panel2.Color:=colordialog3.Color;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
ColorDialog4.Execute;
panel3.Color:=colordialog4.Color;
end;

procedure TForm1.ToolButton10Click(Sender: TObject);
var
gray:array[1..9]of integer;
i,j,orta,tampon:integer;
a,b:integer;TheRgbValue : TColorRef;
begin
a:=1;b:=1;
image1.FreeOnRelease;

  while (a<=image1.Width-2 ) do begin
    while (b<=image1.Height-2) do begin

TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a-1,b-1]));
Gray[1] := Round(((GetRValue(TheRGBValue)) +(GetGValue(TheRGBValue))
+(GetBValue(theRGBvalue )))/3);

TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a-1,b]));
Gray[2] := Round(((GetRValue(TheRGBValue)) +(GetGValue(TheRGBValue))
+(GetBValue(theRGBvalue )))/3);

TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a-1,b+1]));
Gray[3] := Round(((GetRValue(TheRGBValue)) +(GetGValue(TheRGBValue))
+(GetBValue(theRGBvalue )))/3);

TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a,b-1]));
Gray[4] := Round(((GetRValue(TheRGBValue)) +(GetGValue(TheRGBValue))
+(GetBValue(theRGBvalue )))/3);

```

```
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a,b]));
Gray[5] := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
```

```
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a,b+1]));
Gray[6] := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
```

```
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a+1,b-1]));
Gray[7] := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
```

```
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a+1,b]));
Gray[8] := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
```

```
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a+1,b+1]));
Gray[9] := Round(((GetRValue(TheRGBValue)) +(GetGValue(Thergbvalue))
+(GetBValue(theRGBvalue )))/3);
```

```
for i:=1 to 8 do begin
  for j:=i+1 to 9 do begin
```

```
    if gray[i]>gray[j] then begin
      tampon:=gray[i];
      gray[i]:=gray[j];
      gray[j]:=tampon;
    end;
```

```
  end;
end;
```

```
  orta:=gray[5];
  image1.Canvas.Pixels[a,b]:= (RGB(orta,orta,orta));
  b:=b+1;
end;
  a:=a+1;
  b:=1; end;
  image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;
```

```
procedure TForm1.ToolButton11Click(Sender: TObject);
begin
  ListBox1.Items.add('Siyah-Beyaz yapıldı');
  siyahbeyaz(image1.Picture.Bitmap);
end;
```

```
procedure TForm1.siyahbeyaz(yeni:tbitmap);
```

```

var
gray,a,b:integer;TheRgbValue : TColorRef;
begin
a:=1;b:=1;
image1.FreeOnRelease;
  while (a<image1.Picture.Bitmap.Width ) do begin
    while (b<image1.Picture.bitmap.Height) do begin
TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a,b]));
Gray := Round(((GetRValue(TheRGBValue)) +(GetGValue(TheRGBvalue))
+(GetBValue(theRGBvalue )))/3);
if (gray>trackbar3.Position) then
image1.Canvas.Pixels[a,b]:= (RGB(0, 0, 0))
else
image1.Canvas.Pixels[a,b]:= (RGB(255, 255, 255));
  b:=b+1;
  end;
  a:=a+1;
  b:=1;
  end;
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;

```

```

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
var birim:string;
begin
if (form2.RadioButton1.Checked=true) then
birim:=form2.radioButton1.Caption
else if (form2.RadioButton2.Checked=true) then
birim:=form2.radioButton2.Caption
else if (form2.RadioButton3.Checked=true) then
birim:=form2.radioButton3.Caption
else if (form2.RadioButton4.Checked=true) then
birim:=form2.radioButton4.Caption
else if (form2.RadioButton5.Checked=true) then
birim:=form2.radioButton5.Caption
else if (form2.RadioButton6.Checked=true) then
birim:=form2.radioButton6.Caption;

```

```

olcu:=olcu/strtofloat(form2.Edit1.text);//secilen degerin form2 deki edite gore degeri
label6.caption:=form2.Label2.Caption+'pixel='+form2.Edit1.Text+' '+birim;
end;

```

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
var
dosya:TextFile;i,j:integer;
begin
SaveDialog1.Execute;

```



```

assignfile(dosya,SaveDialog1.FileName);
Rewrite(dosya);

for j:=0 to sat do begin
for i:=0 to 3 do begin
write(dosya,StringGrid1.Cells[i,j]+' ');
end;
writeln(dosya)
end;
CloseFile(dosya);
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin

Bitmap := TBitmap.Create;
TRY
  Bitmap.LoadFromFile('C:/test.bmp');
  //OpenPictureDialog1.FileName);
  yBitmap := TBitmap.Create;
  TRY
    if (ComboBox1.Text='Sığdır') then
      begin
        yBitmap.Width :=625;
        yBitmap.Height :=481;
      end else begin
        yBitmap.Width :=round(bitmap.Width*(strtoint(ComboBox1.text)/100));
        yBitmap.Height :=round(bitmap.Height*(strtoint(ComboBox1.text)/100));
      end;
      yBitmap.PixelFormat := pf24bit;
      yBitmap.Canvas.StretchDraw(yBitmap.Canvas.ClipRect,Bitmap);
      Image1.Picture.Graphic := yBitmap
    FINALLY
      yBitmap.Free
    END
  FINALLY
    Bitmap.Free
  END;

label24.Caption:=inttostr(image1.height);
label25.Caption:=inttostr(image1.Width);
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;

procedure tform1.negatif(yeni:tbitmap);
var
kirmizi,yesil,mavi,a,b:integer;
thergbvalue:tcolorref;

```

```

begin
a:=0;b:=0;
//yeni.FreeOnRelease;
  while (a<=yeni.Width ) do begin
    while (b<=yeni.Height) do begin
TheRgbValue:= ColorTorgb(colortorgb(yeni.Canvas.Pixels[a,b]));
Kirmizi := Round(255-(GetrValue(TheRGBValue)));
Yesil := Round(255-(GetgValue(TheRGBValue)));
mavi := Round(255-(GetbValue(TheRGBValue)));
yeni.Canvas.Pixels[a,b]:= (RGB(kirmizi,yesil,mavi));
    b:=b+1;
    end;
    a:=a+1;
    b:=0;
    end;
yeni.SaveToFile('c:\islem.bmp');

end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
begin
ListBox1.Items.add('Kenar bulma yapıldı');
kenarbul(image1.Picture.Bitmap);
end;

```

```

procedure TForm1.kenarbul(yeni:tbitmap);
var
orgbmp,islembmp,orgbmpy,org    : TBitmap;
ust,orta,alt,orj : PRGB24Array;

```

```

boy,en,x,y,toplamb,toplamr,toplamg,toplamby,toplamry,toplamgy,topr,topg,topb:inte
ger;

```

```

begin
  org := TBitmap.create;
  org.PixelFormat := pf24bit;
  org.Height := image1.Height;
  org.Width := image1.Width;
  org.Canvas.Draw(0,0,Image1.Picture.Graphic);

```

```

  orgbmp := TBitmap.create;
  orgbmp.PixelFormat := pf24bit;
  orgbmp.Height := image1.Height;
  orgbmp.Width := image1.Width;
  orgbmp.Canvas.Draw(0,0,Image1.Picture.Graphic);

```

```

orgbmp := TBitmap.create;
orgbmp.PixelFormat := pf24bit;
orgbmp.Height := image1.Height;
orgbmp.Width := image1.Width;
orgbmp.Canvas.Draw(0,0,Image1.Picture.Graphic);

```

```

islembmp := TBitmap.create;
islembmp.PixelFormat := pf24bit;
islembmp.Height := image1.Height;
islembmp.Width := image1.Width;
islembmp.Canvas.Draw(0,0,Image1.Picture.Graphic);

```

```

boy:=islembmp.height;
en:=islembmp.width;
with islembmp do
begin
  for y := 1 to boy - 2 do
  begin
    ust := Scanline[y-1];
    orta := Scanline[y];
    alt := Scanline[y+1];
    orj:=org.ScanLine[y];
    for x := 1 to en - 2 do
    begin

```

```
//s(x)
```

```

    toplamb:=round(((ust[x-1].B*1)+(ust[x].B*2)+(ust[x+1].B*0)+
      (orta[x-1].B*0)+(orta[x].B*0)+(orta[x+1].B*0)+
      (alt[x-1].B*-1)+(alt[x].B*-2)+(alt[x+1].B*-1)));
    if toplamb>255 then toplamb:=255;
    if toplamb<0 then toplamb:=0;

```

```

    toplamr:=round(((ust[x-1].r*1)+(ust[x].r*2)+(ust[x+1].r*0)+
      (orta[x-1].r*0)+(orta[x].r*0)+(orta[x+1].r*0)+
      (alt[x-1].r*-1)+(alt[x].r*-2)+(alt[x+1].r*-1)));
    if toplamr>255 then toplamr:=255;
    if toplamr<0 then toplamr:=0;

```

```

    toplamg:=round(((ust[x-1].g*1)+(ust[x].g*2)+(ust[x+1].g*0)+
      (orta[x-1].g*0)+(orta[x].g*0)+(orta[x+1].g*0)+
      (alt[x-1].g*-1)+(alt[x].g*-2)+(alt[x+1].g*-1)));
    if toplamg>255 then toplamg:=255;
    if toplamg<0 then toplamg:=0;

```

```
//s(y)
```

```

    toplamby:=round(((ust[x-1].B*1)+(ust[x].B*0)+(ust[x+1].B*-1)+
      (orta[x-1].B*2)+(orta[x].B*0)+(orta[x+1].B*-2)+
      (alt[x-1].B*1)+(alt[x].B*0)+(alt[x+1].B*-1)));
    if toplamby>255 then toplamby:=255;

```

```

if toplamby<0 then toplamby:=0;

toplamry:=round(((ust[x-1].r*1)+(ust[x].r*0)+(ust[x+1].r*-1)+
(orta[x-1].r*2)+(orta[x].r*0)+(orta[x+1].r*-2)+
(alt[x-1].r*1)+(alt[x].r*0)+(alt[x+1].r*-1)));
if toplamry>255 then toplamry:=255;
if toplamry<0 then toplamry:=0;

toplamgy:=round(((ust[x-1].g*1)+(ust[x].g*0)+(ust[x+1].g*-1)+
(orta[x-1].g*2)+(orta[x].g*0)+(orta[x+1].g*-2)+
(alt[x-1].g*1)+(alt[x].g*0)+(alt[x+1].g*-1)));
if toplamgy>255 then toplamgy:=255;
if toplamgy<0 then toplamgy:=0;

topr:=round(sqrt((toplamr*toplamr)+(toplamry*toplamry)));
topg:=round(sqrt((toplamg*toplamg)+(toplamgy*toplamgy)));
topb:=round(sqrt((toplamb*toplamb)+(toplamby*toplamby)));

with org do
begin
  orj[x].B:=topb;
  orj[x].r:=topr;
  orj[x].g:=topg;
end;
image1.Picture.Bitmap:=org;

end;
end;
end;
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
ListBox1.Items.add('Keskinleştirme yapıldı');
keskinlestir(image1.Picture.Bitmap);
end;

procedure TForm1.keskinlestir(yeni:tbitmap);
var
orgbmp,islembmp : TBitmap;
ust,orta,alt,orjline : PRGB24Array;
boy,en,x,y,toplamb,toplamr,toplamg:integer;
begin
orgbmp := TBitmap.create;
orgbmp.PixelFormat := pf24bit;
orgbmp.Height := yeni.Height;

```

```

orgbmp.Width := yeni.Width;
orgbmp.Canvas.Draw(0,0,image1.Picture.Graphic);
islembmp := TBitmap.create;
islembmp.PixelFormat := pf24bit;
islembmp.Height := yeni.Height;
islembmp.Width := yeni.Width;
islembmp.Canvas.Draw(0,0,image1.Picture.Graphic);

boy:=islembmp.height;
en:=islembmp.width;
with islembmp do
begin
  for y := 1 to boy - 2 do
  begin
    ust := Scanline[y-1];
    orta := Scanline[y];
    alt := Scanline[y+1];
    orjline:=orgbmp.scanline[y];
    for x := 1 to en - 2 do
    begin
      toplamb:=round((((ust[x-1].B*-1)+(ust[x].B*-1)+(ust[x+1].B*-1)+
        (orta[x-1].B*-1)+(orta[x].B*16)+(orta[x+1].B*-1)+
        (alt[x-1].B*-1)+(alt[x].B*-1)+(alt[x+1].B*-1))/8);
      if toplamb>255 then toplamb:=255;
      if toplamb<0 then toplamb:=0;

      toplamr:=round((((ust[x-1].r*-1)+(ust[x].r*-1)+(ust[x+1].r*-1)+
        (orta[x-1].r*-1)+(orta[x].r*16)+(orta[x+1].r*-1)+
        (alt[x-1].r*-1)+(alt[x].r*-1)+(alt[x+1].r*-1))/8);
      if toplamr>255 then toplamr:=255;
      if toplamr<0 then toplamr:=0;

      toplamg:=round((((ust[x-1].g*-1)+(ust[x].g*-1)+(ust[x+1].g*-1)+
        (orta[x-1].g*-1)+(orta[x].g*16)+(orta[x+1].g*-1)+
        (alt[x-1].g*-1)+(alt[x].g*-1)+(alt[x+1].g*-1))/8);
      if toplamg>255 then toplamg:=255;
      if toplamg<0 then toplamg:=0;

      with orgbmp do
      begin
        orjline[x].B:=toplamb;
        orjline[x].r:=toplamr;
        orjline[x].g:=toplamg;
      end;
      image1.Picture.Bitmap:=orgbmp;

    end;
  end;
end;

```

```

end;
end;
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
ListBox1.Items.add('Blur yapıldı');
blurislem(image1.Picture.Bitmap);
end;

procedure TForm1.blurislem(yeni:tbitmap);
var
orgbmp,islembmp : TBitmap;
ust,orta,alt,orjline : PRGB24Array;
boy,en,x,y,toplamb,toplamr,toplamg:integer;
begin
orgbmp := TBitmap.create;
orgbmp.PixelFormat := pf24bit;
orgbmp.Height := image1.Height;
orgbmp.Width := image1.Width;
orgbmp.Canvas.Draw(0,0,Image1.Picture.Graphic);
islembmp := TBitmap.create;
islembmp.PixelFormat := pf24bit;
islembmp.Height := image1.Height;
islembmp.Width := image1.Width;
islembmp.Canvas.Draw(0,0,Image1.Picture.Graphic);

boy:=islembmp.height;
en:=islembmp.width;
with islembmp do
begin
for y := 1 to boy - 2 do
begin
ust := Scanline[y-1];
orta := Scanline[y];
alt := Scanline[y+1];
orjline:=orgbmp.scanline[y];
for x := 1 to en - 2 do
begin
toplamb:=round(((ust[x-1].B*3)+(ust[x].B*3)+(ust[x+1].B*3)+
(orta[x-1].B*3)+(orta[x].B*8)+(orta[x+1].B*3)+
(alt[x-1].B*3)+(alt[x].B*3)+(alt[x+1].B*3))/32);
if toplamb>255 then toplamb:=255;
if toplamb<0 then toplamb:=0;

toplamr:=round(((ust[x-1].r*3)+(ust[x].r*3)+(ust[x+1].r*3)+
(orta[x-1].r*3)+(orta[x].r*8)+(orta[x+1].r*3)+

```

```

    (alt[x-1].r*3)+(alt[x].r*3)+(alt[x+1].r*3))/32);
    if toplamr>255 then toplamr:=255;
    if toplamr<0 then toplamr:=0;

    toplamg:=round(((ust[x-1].g*3)+(ust[x].g*3)+(ust[x+1].g*3)+
    (orta[x-1].g*3)+(orta[x].g*8)+(orta[x+1].g*3)+
    (alt[x-1].g*3)+(alt[x].g*3)+(alt[x+1].g*3))/32);
    if toplamg>255 then toplamg:=255;
    if toplamg<0 then toplamg:=0;

    with orgbmp do
    begin
        orjline[x].B:=toplamb;
        orjline[x].r:=toplamr;
        orjline[x].g:=toplamg;
    end;
    image1.Picture.Bitmap:=orgbmp;

    end;
    end;
    end;
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
    end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    bmp := TBitmap.create;
    bmp.PixelFormat := pf24bit;
    bmp.Height := image1.Height;
    bmp.Width := image1.Width;
    bmp.Canvas.Draw(0,0,Image1.Picture.Graphic);
    cizim:=1;
    c:=0;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    bmp := TBitmap.create;
    bmp.PixelFormat := pf24bit;
    bmp.Height := image1.Height;
    bmp.Width := image1.Width;
    bmp.Canvas.Draw(0,0,Image1.Picture.Graphic);
    cizim:=2;
    c:=0;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin

```

```
image1.Picture.Bitmap:=bmp;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
aaa:=1;
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
  bmp := TBitmap.create;
  bmp.PixelFormat := pf24bit;
  bmp.Height := image1.Height;
  bmp.Width := image1.Width;
  bmp.Canvas.Draw(0,0,Image1.Picture.Graphic);
  cizim:=3;
  c:=0;

end;

procedure TForm1.Image1Db1Click(Sender: TObject);
begin
if (cizim=3)then begin
image1.Canvas.LineTo(ilknoktax,ilknoktay);
polideger:=polideger+1;
end;

end;

procedure TForm1.Button9Click(Sender: TObject);
var
  i:integer;
begin
for i := 0 to ListBox1.Items.Count-1 do begin
  if (listbox1.Items[i]='Griye çevirme yapıldı')
  then griyecevir(image1.Picture.Bitmap);
  if (listbox1.Items[i]='Siyah-Beyaz yapıldı')
  then siyahbeyaz(image1.Picture.Bitmap);
  if (listbox1.Items[i]='Kenar bulma yapıldı')
  then kenarbul(image1.Picture.Bitmap);
  if (listbox1.Items[i]='Keskinleştirme yapıldı')
  then keskinlestir(image1.Picture.Bitmap);
  if (listbox1.Items[i]='Blur yapıldı')
  then blurislem(image1.Picture.Bitmap);
  if (listbox1.Items[i]='Negatifi alındı')
  then negatif(image1.Picture.Bitmap);

end;

end;
```


end;

```
procedure TForm1.Button10Click(Sender: TObject);
begin
renkoran:=true;
tiklama:=1;
end;
```

```
procedure TForm1.ListBox1Click(Sender: TObject);
var j:integer;
begin
listbox1.Items.Delete(listbox1.ItemIndex);
image1.Picture.LoadFromFile('c:\test.bmp');
for j := 0 to ListBox1.Items.Count-1 do begin
  if (listbox1.Items[j]='Griye çevirme yapıldı')
  then griyeyevir(image1.Picture.Bitmap);
  if (listbox1.Items[j]='Siyah-Beyaz yapıldı')
  then siyahbeyaz(image1.Picture.Bitmap);
  if (listbox1.Items[j]='Kenar bulma yapıldı')
  then kenarbul(image1.Picture.Bitmap);
  if (listbox1.Items[j]='Keskinleştirme yapıldı')
  then keskinlestir(image1.Picture.Bitmap);
  if (listbox1.Items[j]='Blur yapıldı')
  then blurislem(image1.Picture.Bitmap);
  if (listbox1.Items[j]='Negatifi alındı')
  then negatif(image1.Picture.Bitmap);
```

end;

end;

```
procedure TForm1.BitBtn4Click(Sender: TObject);
begin
ColorDialog5.Execute;
panel4.Color:=colordialog5.Color;
renk1sec:=true;
end;
```

```
procedure TForm1.BitBtn5Click(Sender: TObject);
begin
ColorDialog6.Execute;
panel5.Color:=colordialog6.Color;
renk2sec:=true;
end;
```

```

procedure TForm1.BitBtn6Click(Sender: TObject);
begin
ColorDialog7.Execute;
panel6.Color:=colordialog7.Color;
renk3sec:=true;
end;

procedure TForm1.BitBtn7Click(Sender: TObject);
var
deger1r,deger1g,deger1b,deger2r,deger2g,deger2b,deger3r,deger3g,deger3b:integer;
renk1sayi,renk2sayi,renk3sayi,ix,iy:integer;
thergbvalue:TColorRef;

begin
iy:=0;ix:=0;//renkli alanı buldurma
renk1sayi:=0;renk2sayi:=0;renk3sayi:=0;
deger1r:=0;deger1g:=0;deger1b:=0;
deger2r:=0;deger2g:=0;deger2b:=0;
deger3r:=0;deger3g:=0;deger3b:=0;

while (iy<image1.Height) do begin
while (ix<image1.Width) do begin
TheRGBValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[ix,iy]));

if (renk1sec=true) then begin//birinci renk
deger1r:=getrvalue(ColorDialog5.Color);
deger1g:=getgvalue(ColorDialog5.Color);
deger1b:=getbvalue(ColorDialog5.Color);
if ((GetRValue(TheRGBValue)=deger1r) and
(GetGValue(TheRGBValue)=deger1g) and (GetBValue(theRGBvalue )=deger1b))
then
begin
renk1sayi:=renk1sayi+1;
end;
end;
if (renk2sec=true) then begin //ikinci renk
deger2r:=getrvalue(ColorDialog6.Color);
deger2g:=getgvalue(ColorDialog6.Color);
deger2b:=getbvalue(ColorDialog6.Color);
if (((GetRValue(TheRGBValue)=deger2r) and
(GetGValue(TheRGBValue)=deger2g) and (GetBValue(theRGBvalue )=deger2b))
then
begin
renk2sayi:=renk2sayi+1;
end;
end;
if (renk3sec=true) then begin //ucuncu renk
deger3r:=getrvalue(ColorDialog7.Color);

```

```

deger3g:=getgvalue(ColorDialog7.Color);
deger3b:=getbvalue(ColorDialog7.Color);
if ((GetRValue(TheRGBValue)=deger3r) and
(GetGValue(Thergbvalue)=deger3g) and (GetBValue(theRGBvalue )=deger3b))
then
begin
renk3sayi:=renk3sayi+1;
end;
end;
ix:=ix+1;
end;//while end genişlik
iy:=iy+1;
ix:=0;
end;//while end yükseklik
//-----sırano-----alan----oran----toplam alan-----
memo1.lines.add('Renk 1 in degeri='+floattostr(renk1sayi));
memo1.Lines.Add('Renk 2 nin degeri='+floattostr(renk2sayi));
memo1.Lines.Add('Renk 3 nin degeri='+floattostr(renk3sayi));
end;

```

```

procedure TForm1.Button12Click(Sender: TObject);
var
renk,renk1,renk2,renk3:integer;
spermsay:array of array of integer;//[1..250,1..250]of integer;
toplaml:array[1..1000]of integer;
kontrolx,kontroly:array of integer; //[1..40000]of integer;
a,b,sayac,numara,kontrolsay,maxno,i,j,x,y,tane,top,deger:integer;
kedildi:boolean;
TheRgbValue,thergbvalue1 : TColorRef;
kontuzunluk,nekadarrenk:integer;
begin
try
setlength(spermsay,image1.width,image1.Height);
setlength(kontrolx,round((image1.Width*image1.Height)));
setlength(kontroly,round((image1.Width*image1.Height)));
kontuzunluk:=round((image1.Width*image1.Height));

for i:=0 to kontuzunluk-1 do begin
kontrolx[i]:=0;
kontroly[i]:=0;
end;

for i:=1 to 1000 do
toplaml[i]:=0;

a:=1;b:=1;deger:=0;nekadarrenk:=0;
for x:=0 to image1.width-1 do begin
for y:=0 to image1.height-1 do begin

```

```

    spermsay[x,y]:=0;
end;
end;
//beyaz olan yerlere 1 degeri verildi.
//sayac dizinin eleman sayısı
sayac:=0;kontrolsay:=0;numara:=1;
TheRgbValue1 := ColorTorgb(clickrenk);
Reng1:= (GetRValue(TheRGBValue1));
Reng2:= (GetGValue(TheRGBValue1));
Reng3:= (GetBValue(TheRGBValue1));

while (a<image1.Width-1 ) do begin
  while (b<image1.Height-1) do begin
    TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[a,b]));
    Renk:= (GetRValue(TheRGBValue));
    Renk1:= (GetGValue(TheRGBValue));
    Renk2:= (GetBValue(TheRGBValue));

    if ((renk=reng1) and (renk1=reng2) and (renk2=reng3)) then begin
      //bu no kontrol listesine eklenmemişse ekle ve cevresindekileri kontrol et
      kedildi:=false;
      for i:=1 to sayac do begin
        if (kontrolx[i]=a) and (kontroly[i]=b) then kedildi:=true;
        if (kedildi) then break;
      end;
      if (kedildi=false) then begin

        kontrolsay:=kontrolsay+1;
        sayac:=sayac+1;
        kontrolx[sayac]:=a;
        kontroly[sayac]:=b;
        spermsay[kontrolx[sayac],kontroly[sayac]]:=numara;

      end;

      //sayac=kac tane beyaz nokta bulunduđu dizinin eleman sayısını
      //kontrolsay= sunan hangi noktayı kontrol ediyoruz

      while (kontrolsay<=sayac) do begin
        //2 no
        TheRgbValue :=
colortorgb(colortorgb(image1.Canvas.Pixels[kontrolx[kontrolsay]-
1,kontroly[kontrolsay]]));
        Renk:= (GetRValue(TheRGBValue));
        Renk1:= (GetGValue(TheRGBValue));
        Renk2:= (GetBValue(TheRGBValue));

        if ((renk=reng1) and (renk1=reng2) and (renk2=reng3)) then begin

```

```

        kedildi:=false;
        for i:=1 to sayac do begin
            if (kontrolx[i]=kontrolx[kontrolsay]-1)and
(kontroly[i]=kontroly[kontrolsay])
                then kedildi:=true;
                if (kedildi) then break;
            end;
            if (kedildi=false) then begin
                sayac:=sayac+1;
                kontrolx[sayac]:=kontrolx[kontrolsay]-1;
                kontroly[sayac]:=kontroly[kontrolsay];
                spermsay[kontrolx[sayac],kontroly[sayac]]:=numara;

            end;
        end;
        //4 no
        TheRgbValue :=
ColorTorgb(colortorgb(image1.Canvas.Pixels[kontrolx[kontrolsay],kontroly[kontrols
ay]-1]));
        Renk:= (GetRValue(TheRGBValue));
        Renk1:= (GetGValue(TheRGBValue));
        Renk2:= (GetBValue(TheRGBValue));

        if ((renk=reng1) and (renk1=reng2) and (renk2=reng3)) then begin
            kedildi:=false;
            for i:=1 to sayac do begin
                if (kontrolx[i]=kontrolx[kontrolsay])and (kontroly[i]=kontroly[kontrolsay]-
1)
                    then kedildi:=true;
                    if (kedildi) then break;
                end;
                if (kedildi=false) then begin
                    sayac:=sayac+1;
                    kontrolx[sayac]:=kontrolx[kontrolsay];
                    kontroly[sayac]:=kontroly[kontrolsay]-1;
                    spermsay[kontrolx[sayac],kontroly[sayac]]:=numara;

                end;
            end;
            //6 no
            TheRgbValue :=
ColorTorgb(colortorgb(image1.Canvas.Pixels[kontrolx[kontrolsay],kontroly[kontrols
ay]+1]));
            Renk:= (GetRValue(TheRGBValue));
            Renk1:= (GetGValue(TheRGBValue));
            Renk2:= (GetBValue(TheRGBValue));

            if ((renk=reng1) and (renk1=reng2) and (renk2=reng3)) then begin

```

```

        kedildi:=false;
        for i:=1 to sayac do begin
            if (kontrolx[i]=kontrolx[kontrolsay])and
(kontroly[i]=ontroly[kontrolsay]+1)
                then kedildi:=true;
                if (kedildi) then break;
            end;
            if (kedildi=false) then begin
                sayac:=sayac+1;
                kontrolx[sayac]:=kontrolx[kontrolsay];
                kontroly[sayac]:=ontroly[kontrolsay]+1;
                spermsay[kontrolx[sayac],ontroly[sayac]]:=numara;
            end;
        end;
        //8 no
        TheRgbValue :=
ColorTorgb(colortorgb(image1.Canvas.Pixels[kontrolx[kontrolsay]+1,ontroly[kontr
olsay]]));
        Renk:= (GetRValue(TheRgbValue));
        Renk1:= (GetGValue(TheRgbValue));
        Renk2:= (GetBValue(TheRgbValue));

        if ((renk=reng1) and (renk1=reng2) and (renk2=reng3)) then begin
            kedildi:=false;
            for i:=1 to sayac do begin
                if (kontrolx[i]=kontrolx[kontrolsay]+1)and
(kontroly[i]=ontroly[kontrolsay])
                    then kedildi:=true;
                    if (kedildi) then break;
                end;
                if (kedildi=false) then begin
                    sayac:=sayac+1;
                    kontrolx[sayac]:=kontrolx[kontrolsay]+1;
                    kontroly[sayac]:=ontroly[kontrolsay];
                    spermsay[kontrolx[sayac],ontroly[sayac]]:=numara;
                end;
            end;
        // listBox1.items.add(inttostr(sayac)+'sayac-'+inttostr(kontrolsay)+'kontrolsay');
        kontrolsay:=kontrolsay+1;
        end; //while dort nokta kontrol
        //image1.Canvas.TextOut(kontrolx[sayac],ontroly[sayac],inttostr(numara));
        numara:=numara+1;
        kontrolsay:=kontrolsay-1;
        end;//işaretlenmiş bir nokta ise bu noktayı atla
        end;//beyaz nokta olup olmadığı
        b:=b+1;
    end;
    a:=a+1;

```

```

    b:=1;
end;
    for i:=0 to image1.width-1 do
    for j:=0 to image1.height-1 do begin
    for tane:=1 to 1000 do begin
    if (spermsay[i,j]=tane)then begin
    toplam1[tane]:=toplam1[tane]+1;
    nekadarrenk:=nekadarrenk+1;
    end;
    end;
//    image1.Canvas.TextOut(i,j,inttostr(tane));
//    StringGrid2.Cells[i,j]:=inttostr(spermsay[i,j]);
    end;
    top:=0;
    for tane:=1 to 1000 do begin
    if (toplam1[tane]>strtoint(edit5.Text)) then
    top:=top+1;
    end;
memo1.Lines.add('Resimde '+edit5.Text+' degerini geçen');
memo1.Lines.add('seçtiğiniz bolge sayısı='+inttostr(top));
memo1.Lines.add('Toplam Alan='+inttostr(nekadarrenk)+' nokta');

except
showmessage('Resim üzerinde bir renk seçiniz.Daha sonra buraya tıklayınız.');
```

```

end;

procedure TForm1.Button11Click(Sender: TObject);
begin
ListBox1.Items.add('Negatifi alındı');
negatif(image1.Picture.Bitmap);
end;

procedure TForm1.Button13Click(Sender: TObject);
begin
memo1.Lines.Clear;
end;

procedure TForm1.ComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
if (key=#13) then begin

    Bitmap := TBitmap.Create;
```

```

TRY
  Bitmap.LoadFromFile('C:/test.bmp');
  //OpenPictureDialog1.FileName);
  yBitmap := TBitmap.Create;
  TRY
    if (ComboBox1.Text='Sığdır') then
      begin
        yBitmap.Width :=625;
        yBitmap.Height :=481;
      end else begin
        yBitmap.Width :=round(bitmap.Width*(strtoint(ComboBox1.text)/100));
        yBitmap.Height :=round(bitmap.Height*(strtoint(ComboBox1.text)/100));
      end;
      yBitmap.PixelFormat := pf24bit;
      yBitmap.Canvas.StretchDraw(yBitmap.Canvas.ClipRect,Bitmap);
      Image1.Picture.Graphic := yBitmap
    FINALLY
      yBitmap.Free
    END
  FINALLY
    Bitmap.Free
  END;

label24.Caption:=inttostr(image1.height);
label25.Caption:=inttostr(image1.Width);
image1.Picture.Bitmap.SaveToFile('c:\islem.bmp');
end;
end;
procedure TForm1.TrackBar4Change(Sender: TObject);
var
  c,d,ix,iy,noktasayisi:integer;
  TheRgbValue : TColorRef;
begin
  image1.Picture.LoadFromFile('c:\islem.bmp');

  bitmap:=tbitmap.create;
  bitmap.assign(image1.Picture.Bitmap);

  edit4.Text:=floattostr(trackbar4.position);
  c:=0;d:=0;
  while (c<Image1.Width ) do begin
    while (d<Image1.Height) do begin
      TheRgbValue := ColorToRGB(colortorgb(image1.Canvas.Pixels[c,d]));
      if((GetRValue(TheRGBValue)<trackbar4.position)or(GetGValue(TheRGBValue)<trackbar4.position)or(GetBValue(TheRGBValue)<trackbar4.position))
      then begin
        image1.Canvas.Pixels[c,d]:=clwhite
      end;
    end;
  end;
end;

```



```

        end;
        d:=d+1;
    end;
    c:=c+1;
    d:=0;
    end;
noktasayisi:=0;toplam:=0;ix:=0;iy:=0;sat:=0;
while (iy<image1.Height) do begin
    while (ix<image1.Width) do begin
        TheRgbValue := ColorTorgb(colortorgb(image1.Canvas.Pixels[ix,iy]));
    {   degerr:=getrvalue(ColorDialog1.Color);//bu bölüm yeni renge göre sayma
        işlemi yapar
        degerg:=getgvalue(ColorDialog1.Color);
        degerb:=getbvalue(ColorDialog1.Color);
    }
        if ((GetRValue(TheRGBValue)=255) and (GetGValue(TheRGBValue)=255) and
(GetBValue(theRGBvalue )=255)) then
            begin
                noktasayisi:=noktasayisi+1;
                end; //boyabasla if
                ix:=ix+1;
                end;//while end genişlik
                iy:=iy+1;
                ix:=0;
                end;//while end yükseklik
                //-----sırano-----alan----oran----toplam alan-----
                sat:=sat+1;
                stringgrid1.Cells[0,sat]:=inttostr(sat);
                stringgrid1.Cells[1,sat]:=floattostr(round(abs(noktasayisi-toplam)/round(olcu)));
                stringgrid1.Cells[2,sat]:=floattostr(abs((noktasayisi-
toplam)/(bitmap.Width*bitmap.Height)));
                stringgrid1.Cells[3,sat]:=floattostr(round((bitmap.Width*bitmap.Height)/round(olcu
)));
                oran:=oran+abs((noktasayisi-toplam)/(bitmap.Width*bitmap.Height));
                //burada sonuçları yazdırma ve şeklin numarasını verme
                //işlemini gerçekleştiriyorum.
                toplam:=noktasayisi;
                bitmap.free;
            end;
        procedure TForm1.Button14Click(Sender: TObject);
        begin
            parcasay:=parcasay+1;
            if (parcasay=2)then begin
                memo1.lines.Add('İşaretlenen parça sayısı='+inttostr(parcaadet));
                parcasay:=0;parcaadet:=0;
            end;
        end;
    end.

```

**T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**PATOLOJİK GÖRÜNTÜLERİN BİLGİSAYARLI ANALİZ PROGRAMI İLE
DEĞERLENDİRİLMESİ**

Evren SEZGİN

Yüksek Lisans Tezi

Elektronik-Bilgisayar Sistemleri Eğitimi Ana Bilim Dalı

Konya, 2007

T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**PATOLOJİK GÖRÜNTÜLERİN BİLGİSAYARLI ANALİZ PROGRAMI İLE
DEĞERLENDİRİLMESİ**

Evren SEZGİN

YÜKSEK LİSANS TEZİ

ELEKTRONİK-BİLGİSAYAR SİSTEMLERİ EĞİTİMİ ANA BİLİM DALI

Bu tez 03/04/2007 tarihinde aşağıdaki jüri tarafından oybirliği/oy çokluğu ile kabul edilmiştir.

.....
Prof. Dr. Novruz ALLAHVERDİ
(Üye)

.....
Prof.Dr.Mustafa Cihat AVUNDUK
(Üye)

.....
Doç. Dr. Hakan IŞIK
(Danışman)

ÖZET

Yüksek Lisans Tezi

PATOLOJİK GÖRÜNTÜLERİN BİLGİSAYARLI ANALİZ PROGRAMI İLE DEĞERLENDİRİLMESİ

Evren SEZGİN

Selçuk Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik-Bilgisayar Sistemleri Eğitimi Ana Bilim Dalı

Danışman: Doç. Dr. Hakan IŞIK

2007, 81 Sayfa

Jüri: Prof.Dr. Novruz ALLAHVERDİ

Prof.Dr. Mustafa Cihat AVUNDUK

Doç.Dr. Hakan IŞIK

Bu çalışmada, tanı amacıyla hastalardan alınan doku örneklerinden hazırlanan preparatlar üzerinde değerlendirme yapmamızı sağlayan, “Görüntü Analizi Programı” geliştirilmiştir. Programın amacı, kişiden kişiye değişen değerlendirmeleri azaltmak ve ölçümleri belli bir standarda sokmaktır.

Görüntüler Selçuk Üniversitesi Meram Tıp Fakültesi Patoloji Anabilimdalı’ndaki örneklerden alınmıştır. Görüntünün seçimi ile analiz başlar. Görüntü üzerinde; alan bulma, açı bulma, kenar bulma, bulanıklaştırma, keskinleştirme, alan saydırma, renkleri birbirine oranlama, negatifini alma, griye çevirme, renk skalasına göre boyama, şekil(dikdörtgen, daire, poligon, rastgele) kullanarak analiz etme işlemleri yapılabilir. Uzman patologun görüntü analiz sonuçlarına göre değerlendirme yapması ile süreç son bulmaktadır.

Bilgisayarlı görüntü analizi sistemi sayesinde, patolojik görüntüler üzerinde sayısal verileri kullanarak hızlı, güvenli ve gerçeğe daha yakın sonuçlar üretileceđi öngörülmektedir.

Anahtar Kelime: Patoloji, Görüntü Analizi, Görüntü İşleme.

ABSTRACT

Master's Thesis

**EVALUATING PATHOLOGICAL VIEWS WITH COMPUTER ANALYSE
PROGRAMME**

Evren SEZGİN

Selçuk University

Graduate School of Natural and Applied Sciences

Department of Electronic-Computer Systems Education

Supervisor: Assoc. Prof. Dr. Hakan IŞIK

2007, 81 Pages

Jury: Prof.Dr. Novruz ALLAHVERDİ

Prof.Dr. Mustafa Cihat AVUNDUK

Assoc.Prof.Dr. Hakan IŞIK

In this research, an “Image Analysis Program”, which helps to evaluate sample tissues taken from patients for diagnosis, has been developed. The aim of the program is to lower evaluations changing from one person to another and put measurements into a standart.

Images has been taken from the department of patology, Meram Medicine Faculty of Selçuk University. Analysis begins with the selection of the image. Analysis process includes finding area, angle and margin on the image, blur and sharpening the image, comparing colors, turning to image negative and grey, painting according to color scale, shaping(rectangle, circle, polygon, custom shape). The process ends with the evaluation of a pathologist according to the result of analysis.

It is believed that fast, reliable and more definite results will be acquired by using statistical data on pathological images with the usage of computer based image analysis system.

Key Words : Pathology, Image Analysis, Image Processing.

ÖNSÖZ

Dünya’ da teknolojinin hızlı gelişmesi sonucu, tıp alanında da değişimler gözlenmektedir. Bu çalışmada, patoloji alanında görüntü işleme teknikleri kullanılarak hastalara tanı koymada yardımcı olacak bir program geliştirilmiştir.

Bu tez çalışmamda bana manevi desteğini her zaman yanımda hissettiğim aileme ve çalışmamın her aşamasında ilgi ve desteğini yanımda hissettiğim danışman hocam Doç. Dr. Hakan IŞIK ve Prof.Dr. Mustafa Cihat AVUNDUK hocalarıma teşekkür ederim.

2007, Konya

Evren SEZGİN

İÇİNDEKİLER DİZİNİ

1.GİRİŞ	1
2.KAYNAK ARAŞTIRMASI.....	2
3.MATERYAL ve METOD.....	4
3.1. Mikroskopik ve Makroskopik Patoloji.....	4
3.1.1.Mikroskopik tanımlama.....	4
3.1.2. Makroskopik tanımlama	4
3.1.3. Patoloji.....	4
3.1.3.1. Tanım	4
3.1.3.2. Patoloji de yapılan işlemler;.....	5
3.1.3.3. Nicel Patoloji.....	6
3.1.3.4. Nesnenin belirlenmesi, analog ortamdan bilgisayar ortamına geçiş ...	8
3.1.3.5.Verilerin değerlendirilmesi	11
3.2. Temel Kavramlar	12
3.2.1.Görüntü elemanı(pixel).....	12
3.2.2. Çözünürlük.....	13
3.2.3. Dijital Görüntü	13
3.2.4. Görüntü korelasyonu	13
3.2.5. Renk modelleri	14
3.2.5.1 RGB modeli.....	15
3.2.5.2.CMY modeli.....	16
3.2.5.3. YIQ modeli.....	17
3.2.6. Dijital görüntü işleme	17
3.2.7. Bitmap tabanlı resimler.....	17
3.2.8. Komşuluk tipleri.....	18
3.2.9. Median filtreleme	18
3.2.10. Sobel operatörü	19
3.2.11. Görüntünün negatifi.....	19
3.3. Uygulama.....	20
3.3.1.Gri tona çevirme.....	22
3.3.2. Referans Değerinin Belirlenmesi	23
3.3.3.Noktaya Göre İşlem Yapma.....	24

3.3.4.Çizime göre işlem yapma	28
3.3.5. Bölgeleri saydırma işlemi	32
4.GERÇEKLEŞTİRİLEN SİSTEMİN PERFORMANS ANALİZİ.....	35
4.1. Alan Hesap Testi.....	35
4.2. Açık Ölçüm Testi.....	40
4.3.Parça Saydırma Testi.....	42
5.SONUÇ VE TARTIŞMA.....	44
6.KAYNAKLAR.....	45
EKLER.....	49
Ek 1- Görüntü Analiz Programı Kodları.....	49

ŞEKİL DİZİNİ

Şekil 3.1. Analog ortamdan bilgisayar ortamına aktarma işlem şeması.....	10
Şekil 3.2. Bir görüntünün üzerinde istenilen bölümü çıkarma.	11
Şekil 3.3. Görüntü Matrisi.....	14
Şekil 3.4. RGB modeli renk uzayı.....	16
Şekil 3.5. Örnekleme çeşitleri : 4 bağlantılı komşuluk ve 8 bağlantılı komşuluk	18
Şekil 3.6. Sobel matrisi.....	19
Şekil 3.7. Görüntünün Negatife çevrilmesi	20
Şekil 3.8. Görüntü Analiz Programının Algoritması.....	22
Şekil 3.9. Gri tona çevirme işlemi.....	23
Şekil 3.10. Referans değerini belirleme (Örnek resmi ; Yücel M.T. 2005).....	23
Şekil 3.11. Yayılma Algoritma Şeması	24
Şekil 3.12. Noktaya göre işlem yapma algoritması.....	27
Şekil 3.13. Noktaya göre işlem yapma sonucu elde edilen görüntüler	28
Şekil 3.14. Rastgele seçim şeması.....	28
Şekil 3.15. Rastgele seçim işlemi algoritması	31
Şekil 3.16. Rastgele Seçim algoritması örnek görüntüler.....	32
Şekil 3.17. Bölge saydırma algoritmasının şeması	33
Şekil 3.18. Bölge saydırma sonucu elde edilen görüntüler	34
Şekil 4.1. Dairesel şablon üzerinde paraların alanının tespiti.	35
Şekil 4.2. Şablonda Alan Hesaplama Görüntüsü.....	36
Şekil 4.3. Clemex Vision Lite 3,5 Programı görüntüsü.....	37
Şekil 4.4. Alan hesaplama sonuç bölümü	38
Şekil 4.5. Karşılaştırılan Alanların Çizgisel Grafiği.....	39
Şekil 4.6. Açık ölçme görüntüsü	41
Şekil 4.7. Açık ölçme sonucu.....	41
43	
Şekil 4.8. Bölge sayısını bulma test örneği	43
Şekil 4.9. Bölge sayısını bulma sonucu	43

ÇİZELGELER DİZİNİ

Çizelge 4.1. Programın alan karşılaştırması sonuçları.....	39
Çizelge 4.2. Açı ölçüm analiz sonuçları.....	40
Çizelge 4.3. Bölge saydırma sonuçları.....	42