

**İSTANBUL TEKNİK ÜNİVERSİTESİ**  
**Bilgisayar ve Bilişim Fakültesi**

**Dijital Patoloji ve Görüntü İşleme**

**STAJ**  
**Yunus GÜNGÖR**  
**150150701**

**Yaz / 2017**

**İstanbul Teknik Üniversitesi**  
**Bilgisayar ve Bilişim Fakültesi**  
**STAJ RAPORU**

Akademik Yıl:2016-2017

Staj yapılan dönem:☒Yaz ☐Bahar ☐Güz

**Öğrenci ile ilgili bilgiler**

Adı ve Soyadı: Yunus GÜNGÖR

Öğrenci Numarası: 150150701

Bölüm: Bilgisayar Mühendisliği

Program: Bilgisayar Mühendisliği (%30 İngilizce)

E-posta Adresi: yunusgnr@gmail.com

(Cep) Tel No: 05436866535

ÇAP öğrencisi misiniz? ☐Evet (ÇAP yaptığınız Fakülte/Bölüm: \_\_\_\_\_)

☒ Hayır

Mezuniyet durumunda mısınız? ☐ Evet

☒ Hayır

Yaz okulunda ders alıyor musunuz? ☐Evet (Ders sayısı: \_\_)

☒ Hayır

**Öğrencinin çalıştığı kurum ile ilgili bilgiler**

İsmi: VİRASOFT YAZILIM TİCARET LTD. ŞTİ.

Birimi: Görüntü İşleme

Web Adresi: <http://www.virasoft.com.tr>

Kısa Adresi: Reşitpaşa Mahallesi, İTÜ Maslak Kampüsü, 34467 Sarıyer/İstanbul, Türkiye

**Yetkili kiři ile ilgili bilgiler**

Bölümü: Görüntü İşleme  
Unvanı: Arařtırmacı ve Proje Yöneticisi  
Adı ve Soyadı: Salar Razavi  
(Kurumsal) E-posta: salarra67@gmail.com  
(Kurumsal) Tel. No.: -

**Yapılan iş ile ilgili bilgiler**

Staj yeri ☒Türkiye  
☐Yurtdışı  
Staj başlangıç tarihi 31.07.2017  
Staj bitiş tarihi 25.08.2017  
Stajda çalışılan net **gün** sayısı 20  
Staj süresince sigortanız var mıydı? ☒Evet, İTÜ tarafından sigortalandım.  
☐Evet, kurum tarafından sigortalandım.  
☐Hayır, yurtdışı stajı yaptım.  
☐Hayır.

## STAJ RAPORU ONAY FORMU

### İÇİNDEKİLER

|   |   |
|---|---|
| 1. KURULUŞ HAKKINDA BİLGİLER.....         | 1 |
| 2. GİRİŞ.....                             | 2 |
| 3. STAJ PROJESİNİN TANIMI VE ANALİZİ..... | 3 |
| 4. SONUÇ.....                             | 9 |
| 5. REFERANSLAR.....                       | 9 |
| 6. EKLER.....                             | 9 |

150150701 numaralı, Yunus Güngör adlı öğrencinin, yukarıda “İçindekiler” bilgisi verilen staj raporu, görülmüş ve uygun bulunmuştur.

Formu Dolduran Firma Yetkilisinin Adı ve Soyadı: *Sait Rıza*

Yetkilinin Unvanı: *Spesiyel İhtisas Uzmanı / Proje Yöneticisi*

Müdür-İmza-Kaşe: *Gökhan HATİPOĞLU*

(Kurumsal) E-posta: *info@virasoft.com.tr*

(Kurumsal) Tel. No.: *0539 949 65 25*

**VIRASOFT YAZILIM TİC.A.Ş**  
Resitpaşa Mh.Katırlı Cd.Kosgeb  
B Blok No:2/48/1 Sarıyer/İST.  
Sarıyer V.D.9250528222  
Ticaret Sicil No:1672-5

## 1. KURULUŞ HAKKINDA BİLGİLER

*ViraSoft* 2015 tarihinde kurulmuş, dijital patoloji üzerine araştırma ve geliştirme yapmakta olan ve özellikle kanser teşhisi üzerine ürünler üretmeye yoğunlaşmış bir firmadır. Verdiği hizmetler ve geliştirmekte olduğu ürünler arasında laboratuvar bilgi yönetim sistemi, *VIRAPATH* isimli biyomedikal görüntü analiz yazılımı ve *TELEPATH* isimli hasta bilgi yönetim, danışmanlar arası haberleşme ve bilgi paylaşım sistemi bulunmaktadır. Bunların dışında akıllı telefonun kamerası ile mikroskop arasındaki bağlantıyı sağlayarak düşük maliyetli görüntüleme ve bilgi aktarımı sağlayan bir aparat üstünde de çalışılmaktadır.

T.C. Sanayi Bakanlığı ve İTÜ ARI Teknokent'ten destek alan firma İTÜ Ayazağa kampüsü KOSGEB ofisleri B Blok'ta iki ofise sahiptir. Ofislerden biri asıl çalışma alanı olarak kullanılırken, diğer ofis toplantı amaçlı ve yönetimde görev alan iki kişinin çalışma alanı olarak kullanılmaktadır. Asıl çalışma alanı içerisinde görüntü işleme ekibi, mobil yazılım ekibi, masaüstü yazılım ekibi, web yazılım ve tasarım ekibi bulunmaktadır. Firmanın çalışan sayısı az olmasına rağmen oluşturulan organizasyon dengeli ve sistemattir. Görüntü işleme ekibi diğer ekipler için görüntü işleme altyapısı ve algoritmaları oluşturmakla; bu sistemlerin stabil çalışmasını sağlamakla yükümlüdür. Masaüstü, mobil ve web ekibi ise belirtilen platformlar için uygulama geliştirmekte ve tasarlanan algoritmaları uygulamalara entegre etmektedir. Yaklaşık 12 çalışanı olan şirkette yarı zamanlı çalışanlar da bulunmaktadır. Çalışanların bir kısmı eğitime devam etmektedir ve çalışanların yaş ortalaması 24-25 civarındadır.

*ViraSoft* bu yıl 6 stajyere staj imkânı sağlamıştır. Bu stajyerlerden 4 kişi haziran ayında 2 kişi ağustos ayında stajını tamamlamıştır. Ağustos ayında staj yapan 2 öğrenci de görüntü işleme üzerine çalışma yapmıştır. Şirketteki çalışma düzeni görevlendirme üzerine kuruludur. Görevlerin çalışanlara dağıtılmasını proje müdürleri yapmaktadır ancak proje müdürleriyle bir üst-alt ilişkisi bulunmamaktadır zira proje müdürü unvanına sahip kişiler de görev alır ve diğer yazılımcılarla birlikte bu görevleri yerine getirirler. Genel olarak rahat bir çalışma ortamına sahip olan şirkette yapılan projeler aksamadan ilerlemektedir. Stajyerlerden istenen haftalık rapor ile yapılan işler takip edilmektedir. Stajyerler ve birçok yazılımcı için hafta içi 10:00-18:00 saatleri arası mesai saati iken bazı durumlarda cumartesi çalışmakta olan yazılımcılar da vardır. Benim görev yaptığım görüntü işleme departmanında cumartesi görevlendirme yapılmamaktadır.

## 2. GİRİŞ

Görüntü işleme üzerine yaptığım ilk zorunlu stajımda bilgisayar bilimlerinde araştırma geliştirme nasıl yapılır, kullanılan yöntemler nelerdir, teorik bilginin pratik uygulamaya dönüştürülme aşamaları nelerdir öğrenmiş bulunmaktayım. Bunların yanında yeni kurulan bir şirketin organize edilmesi ve yönetilmesi nasıl yapılır yakından gözlemlene şansı elde ettim. Daha çok araştırma ve geliştirme yapan bir bölümde çalıştığım için paket yazılım nasıl hazırlanır, testlerin yapılması ve hata ayıklama gibi konularda tecrübe elde etme imkânı bulamamış olmama rağmen, görüntü işleme ve dijital patoloji ile ilgili birçok tecrübe ve bilgi kazanmış durumdayım. Elde ettiğim bu kazanımlar gelecekte mesleğimi daha iyi yerine getirebilmeme yardımcı olacak ve kendimi yönlendirmeme yardımcı olacaktır. Yapılan araştırmalarda sistematik bir yöntemin kullanılmasının önemini kavradığım staj dönemimde ayrıca teorik bilgilerin gelişmiş bir ortam olmadan uygulanmasının pratik zorluklarını yaşamış bulunmaktayım. Güncel araştırmaları kullanarak daha önce denenmemiş veya testleri tamamlanmamış algoritmaları pratik uygulamaya geçirmek gerçekten zorlayıcıydı.

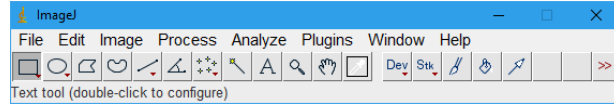
### 3. STAJ PROJELERİNİN TANIMI VE ANALİZİ

Stajyerler staj sırasında verilecek görevlere uyum sağlayabilmek için ilk önce kısa bir eğitim ve araştırma döneminden geçirildi. Daha sonra yaptıkları sunuma ve yoğunlaştıkları konulara göre görüntü işleme ekibi, mobil yazılım ekibi, masaüstü yazılım ekibi, web yazılım ve tasarım ekiplerinden birine atanan stajyerler, diğer yazılımcılar ile birlikte çalışarak stajlarını bu ekiplerden birinin bir üyesi olarak tamamladılar. Verilen görevleri yerine getirerek yapılan projelere katkıda bulunan stajyerler birbirinden farklı projeler geliştirdi. Ağustos ayında stajını tamamlayan diğer stajyer arkadaşıyla birlikte görüntü işleme üzerine yoğunlaşarak hazırladığımız sunumlar aracılığıyla görüntü işleme biriminde stajımızı tamamladık. Staj süresince görüntü işleme proje müdürü tarafından verilen görevleri yerine getirerek; *imageJ* platformu üzerinde *Java* dili ile istenen algoritmaları uyguladım, denedim ve dijital patolojik görüntüler üzerinde çalışabilecek şekilde algoritmaları optimize ettim. Daha sonra algoritmaları bu platforma eklenti haline getirdim ve testlerini tamamladım. Daha sonra oluşturduğum kodlar, test ettiğim algoritmalar ve tamamladığım diğer çalışmaları rapor hazırlayarak stajyerler ile ilgilenen proje müdürüne ilettim.

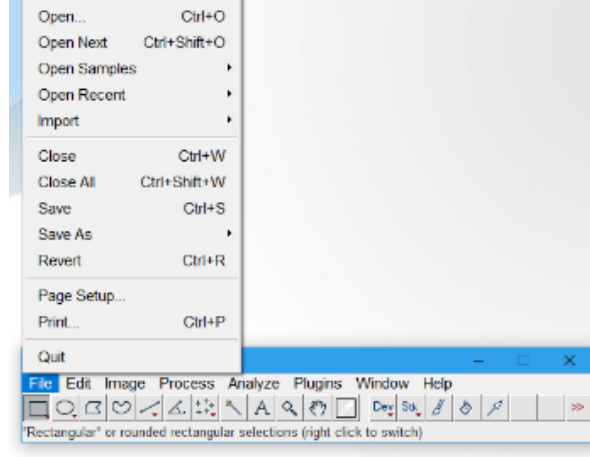
Nesneye dayalı bir programlama dili olan *Java*, yazılım geliştirilmesi kolay bir dil olarak değerlendirilebilir. Özellikle birçok karmaşık matematiksel işlem ve algoritma için hazır kütüphaneleri bulunduran, taşınabilir olan, güvenli olan, paralel iş yeteneği bulunan *Java* dili; hafızadan otomatik olarak yer ayırması ve hafıza yönetimini gerçekleştirmesi sebebiyle patolojik resimler gibi büyük boyutlu dosyalar ile yazılım geliştirme yaparken büyük avantaj sağlamaktadır. Ayrıca nesneye dayalı bir programlama dili olması sebebiyle projeler arası entegrasyon, kodun daha sonra geliştirilmeye devam edilmesi, proje yönetimi gibi konularda üstünlük sağlamaktadır. *Java* dili taşınabilir ve platformdan bağımsız olabilmesini *Java Çalışma Zamanı Ortamına* borçludur. *Java Çalışma Zamanı Ortamı* içerisinde bulunan *Java Sanal Makinesi* ve *Java Uygulama Programı Arabirim*, *Java* dilinin derlenmesi ile elde edilen *bytekod* adlı makine dili ile gerçek makine arasındaki bağlantıyı kurarak komutların sistemden bağımsız gerçekleşmesini sağlar. Her sistem için yalnızca bir kez uyarlanan *Java Sanal Makinesi* ve *Java Uygulama Programı Arabirimi* ile uyarlanan sistem *Java* dili ile uyumlu hale getirilir. Bu yöntemin dezavantajı ise; *Java Çalışma Zamanı Ortamının* sistemde yüklü olması gerekmesidir. Herhangi bir metin düzenleyicisi ile yazılabilen *Java* dilini derlemek için *Java Geliştirici Kiti* kullanılır. *Eclipse*, *Netbeans*, *IntelliJ* gibi *Java Geliştirici Kitini*, metin editörünü ve yazılım yazmaya ve yazılan yazılımı yönetmeye yarayan birçok farklı sistemi bünyesinde bulunduran Entegre Geliştirici Ortamı (IDE) yazılım geliştirmek için kullanılabilir. Staj süresince *Java* kodlarını yazabilmek için *Atom v1.19.3 x64* metin editörünü, derleyebilmek için ise *ImageJ* ile birlikte *Java Geliştirici Kiti 8u144* kullandım.

*ImageJ* platformu dijital medikal resimleri işlemeye yönelik *National Institutes of Health (NIH)* tarafından geliştirilmiş bir yazılımdır. *Java* programlama dili ile açık kaynak kodlu olarak yazılan *ImageJ*, medikal resim işleme için maliyeti düşük bir alternatif olarak kullanılmaktadır; bu yazılımı gene *Java* diliyle yazılan eklentileri ve daha önceden kaydedilmiş makroları desteklemekte, *Java* kodunu bünyesinde derleyebilmekte ve çalıştırabilmektedir (Girish V, Vijayalakshmi A, 2004, s.47). Esnek yapısı sayesinde birçok farklı medikal görüntüleme metodu ve formatı için rahatça kullanılabilen *ImageJ* yazılımı 8-bit, 16-bit, 32-bit resimleri TIFF, GIF, JPEG, BMP, DICOM, FITS ve işlenmemiş formatlarda okuyabilir, paralel olarak çalışabilir, resmi kesebilir, döndürebilir, 32 kat yaklaşılabılır, renk uzayı değiştirme, kontrast, parlaklık, keskinleştirme, yumuşatma ayarlarını; medyan filtresi, köşe bulma gibi popüler görsel işleme algoritmalarını destekler; gerçek dünyada bulunan milimetre gibi ölçüm birimleri ile kalibre yapılmasına imkan verir (Werner B, 2000). *ImageJ* yazılımı <https://imagej.nih.gov/ij/download.html> adresinden elde edilebilir. *ImageJ* üzerinde eklenti

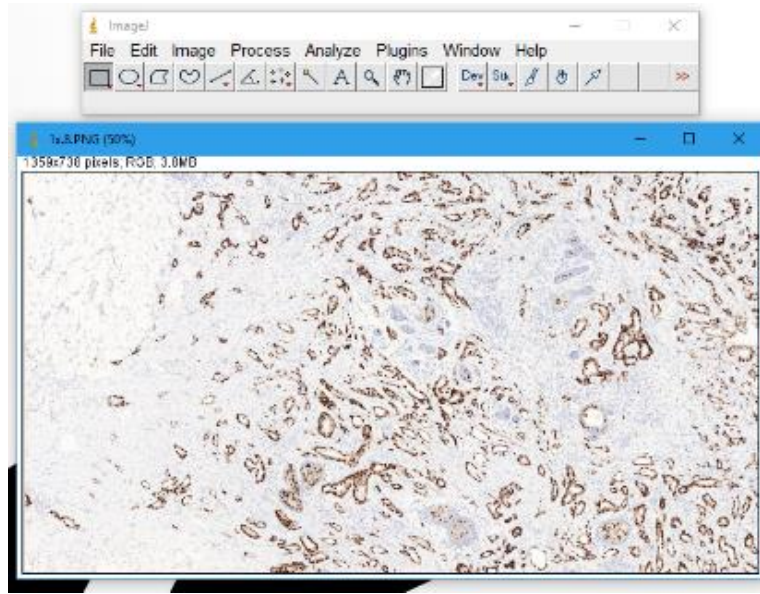




Şekil 1 - ImageJ ara yüzü

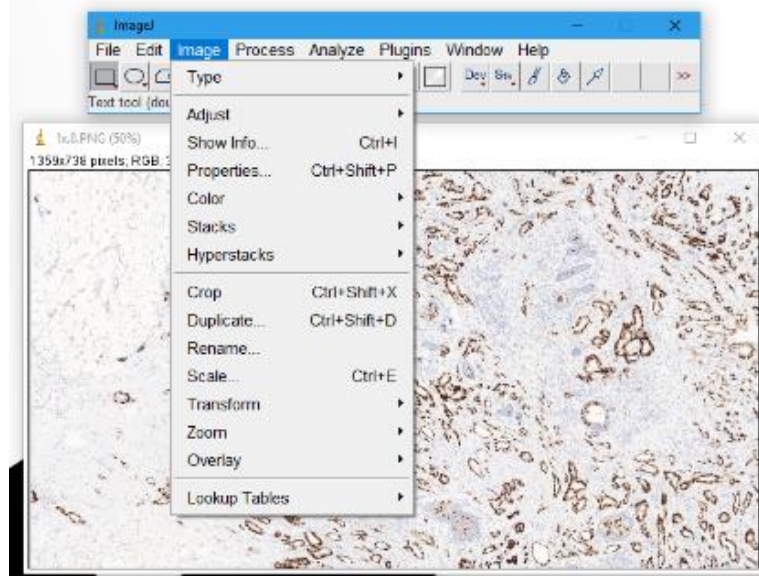


Şekil 2 - ImageJ dosya menüsü

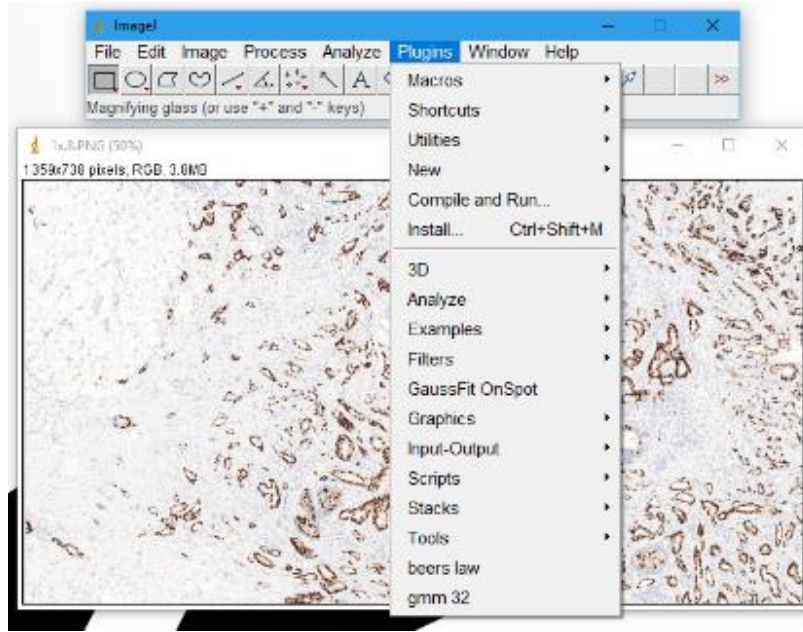


Şekil 3 - ImageJ resim penceresi ve ara yüz

geliştirmesi yapılırken *Writing ImageJ PlugIns – A Tutorial* (Werner B, 2000) adlı kaynaktan yararlanılarak; resmin okunması ve bir matrise dönüştürülmesi için platformun sağladığı araçlardan özellikle faydalanılmıştır. *ImageJ* ara yüzü çeşitli menülerin ve araçların bulunduğu yatay bir pencereden oluşmaktadır. Bu ara yüz şekil 1’de görülebilir. Herhangi bir resmi açmak için şekil 2’deki dosya menüsündeki aç seçeneği kullanılabilir. Resim açıldıktan sonra ara yüzde bulunan araçlar ve düzenle menüsü ile seçim, yakınlaştırma ve kırpma yapılarak işlem uygulanmak istenen alan seçilebilir. Şekil 4’de görülen resim menüsünde ise renk uzayı, renk derinliği, parlaklık, kontrast, boyut gibi seçeneklerin değiştirilmesi için gerekli seçenekler vardır. Şekil 5’de görülen eklentiler menüsünden ise programla birlikte gelen eklentiler kullanılabilir, yeni eklentiler ve makrolar yüklenebilir, yazılan eklentiler derlenebilir ve çalıştırılabilir, yeni eklenti oluşturulabilir.



Şekil 4 – ImageJ resim Menüsü



Şekil 5 - ImageJ eklentiler menüsü

Ayrıca staj süresince bazı algoritmaları önce *MATLAB* ortamında deneyerek, daha sonra uygulamaya geçirdim. *MATLAB MathWorks* tarafından geliştirilen, kendi programlama diline sahip olan özellikle karmaşık matematiksel deneyleri gerçekleştirmek için geliştirilen ancak birçok sektörde kullanılan bir programdır. “*toolbox*” adı verilen ek yazılımlar ile işlevselliğinin arttırılabildiği yazılım matris işlemleri, grafik çizimi, kullanıcı ara yüzü oluşturma, diğer programlama dilleri için ara yüz oluşturma gibi işlemleri yerine getirebilir. Ayrıca bilgisayar bilimi ve matematik ile ilgili birçok algoritmayı içeren ve stabil olarak uygulayabilen farklı *toolboxlar* bulunmaktadır. Görüntü işleme, makine öğrenmesi gibi konuları hızlıca öğrenebilmek için ideal ortamdır. Stajım süresince *MATLAB R2016a (9.0.341360)* yazılımını algoritmaları test etmek için kullandım.



Geliştirme ortamları dışında *Adobe Photoshop CC 2017.0.1* adlı resim düzenleme yazılımını işlenen resimleri işlenmeyen ve diğer yöntemlerle işlenen resimlerle karşılaştırmak için kullandım. Hazırlanan sunum için ise *Microsoft Office PowerPoint 2016* adlı sunum yazılımını kullandım. İstenilen haftalık raporları hazırlamak için ise *Microsoft Office Word 2016* adlı kelime işlemci yazılımını kullandım.

### 3.1 Patoloji ve Görüntü İşleme Üzerine Eğitim

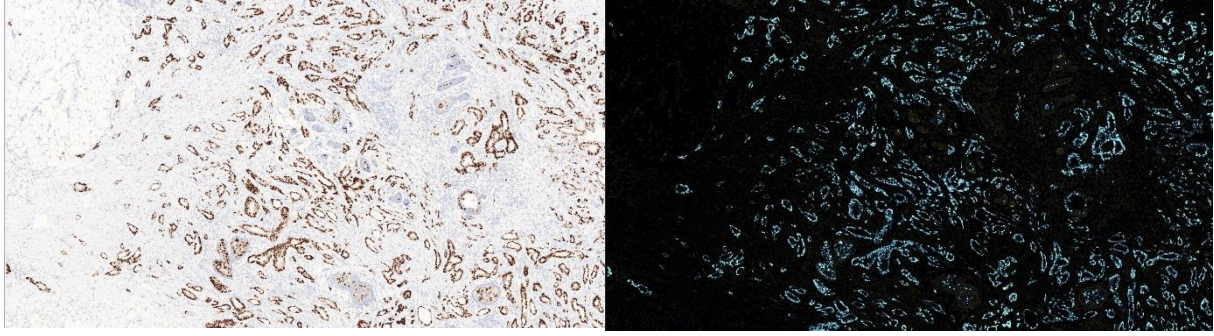
Staja başlarken verilen ilk görev, verilecek diğer görevleri yerine getirebilmek amacıyla, gerekli bilgiye ulaşabilmek için kullanılan yöntemler ve algoritmalar ile ilgili araştırma yapmak ve verilen makaleleri okuyarak bir sunum hazırlamak oldu. İlk hafta boyunca firmada bulunan yazılımcılara ve diğer çalışanlara soru sorarak yardım aldık ve kendimizi geliştirdik. Hazırlanan sunuma <https://ynsgnr.github.io/patolojiSunum.pdf> adresinden ulaşılabilir. Sunum içeriğinde tanı konulmasının adımları, dijital patolojinin tanı konulmasındaki yeri, biyokimyasal boyama metotları, görüntü işlemenin yararları, dijital patolojinin yararları, *gaussian* filtresi, *top-hat* filtresi, *bottom-hat* filtresi gibi görüntü işleme filtreleri, *watershed* algoritması, *graph-cut* segmanlara ayırma algoritması gibi segmanlara ayırma algoritmaları, *Maximally Stable Extremal Regions* ve *Scale-Invariant Keypoints* gibi epipolar geometri ve nesne tanımlama algoritmaları bulunmaktadır. Şirket tarafından verilen makaleler incelenerek ve araştırma yapılarak hazırlanan bu sunum ikinci hafta sunulmuştur. Sunumdan sonra atandığımız bölümün proje müdürü tarafından, bizden görüntü işleme ile ilgili daha fazla makale okumamız ve daha fazla araştırma yapmamız istenmiştir. Yapılan araştırmalar sırasında uygulanan yöntemler incelenmiş ve genel olarak hücre saymada en başarılı yöntemin hücreleri arka plandan ayırdıktan sonra, her bir hücreyi daha önce eğitilmiş bir derin öğrenme ağında işleyerek sınıflandırılması olduğu sonucuna varılmıştır.

Sunumda bulunan bilgiler dışında dijital resimlerde kullanılan renk uzayları, parlaklık, kontrast, histogram gibi bilgiler de araştırılmıştır. Başlıca renk uzaylarından biri olan *RGB* renk uzayı İngilizce kırmızı, yeşil ve mavi kelimelerinin baş harflerinden oluşmaktadır ve bu renk uzayında resimdeki her piksel 0 ile 255 arasında değişen 3 farklı değer ile ifade edilir. İnsan gözündeki kırmızı, yeşil ve mavi ışınları algılayan sinir hücrelerinden esinlenerek oluşturulan bu sistem günümüzde kullanılan çoğu renkli ekranın temelini oluşturmaktadır. Bunun dışında *CMYK* gibi basılı materyallerde daha çok kullanılan; insan gözünün açık renk algılamasını öne çıkaran *LAB* renk uzayı gibi birçok farklı renk uzayı tasarlanmıştır. Ancak en popüler renk uzayı *RGB* olduğu için staj süresince yapılan çalışmalar *RGB* renk uzayı ile yapılmıştır. Histogram ise istatistik biliminde de kullanıldığı gibi, veri dağılımının sütun grafiği ile temsil edilmesidir. Görsellere uyarlandığı zaman, renk uzayındaki her bir boyut için ayrı bir histogram oluşturulur. Bu histogram dağılımlarına ve uygulanan istatistiksel metoda göre görsellerdeki pikseller üzerinde gruplandırma yapılabilir.

### 3.2 Beer Lambert Yasası

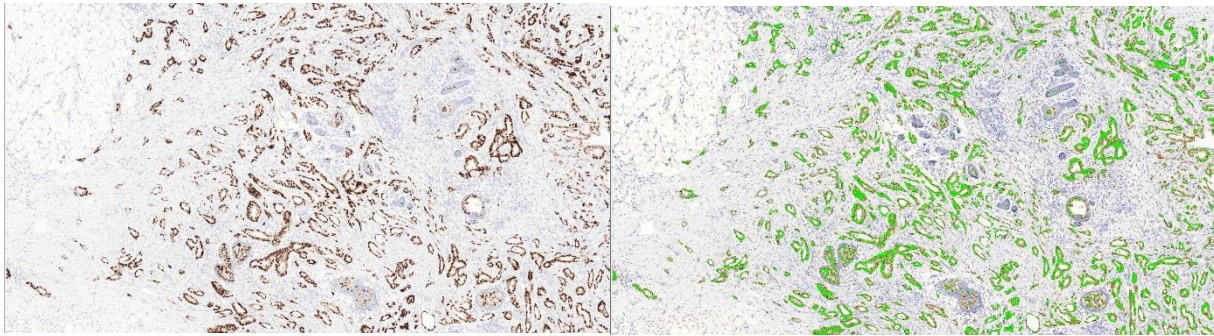
Araştırmalar sırasında *Deep Learning Based Automatic Immune Cell Detection for Immunohistochemistry Images* (Chen, Chefd’hotel, 2014) adlı kaynakta kullanılan Beer Lambert yasası uygulanarak *ImageJ* için bir eklenti programlanması yapılmıştır. Bu yasa ışığın materyal içerisinde ilerken gittiği uzaklık, ışığın soğrulma katsayısı ve materyalin içinden

geçebilen ışık miktarı arasında logaritmik bir bağlantı olduğunu öne sürmektedir. RGB renk uzayı düşünüldüğünde, her pikselde bulunan üç renk değerinin büyüklüğü ile bu yasada materyalin içinden geçebilen ışık miktarı arasında bir bağlantı kurulabilir. Beer Lambert yasası belirtilen kaynakta bulunan yöntem kullanılarak  $O_c = -\log\left(\frac{I_c}{I_0}\right)$  olarak uygulanmıştır;  $c$  harfi RGB uzayındaki indeksini,  $I_0$  ise arka plan piksel değerini belirtmektedir; sonuç olarak  $O$  optik yoğunluk resmi elde edilmiştir (Chen, Chefd'hotel, 2014). Elde edilen optik yoğunluk resmi ve orijinal resim şekil 6'da görülebilir.



Şekil 6 - Beer Lambert yasasının RGB renk uzayındaki bir resme uygulanması (Sol – Orijinal resim, Sağ – işlenmiş resim)

Optik yoğunluğun hesaplanması ile üretilen yeni görsel üzerinde belirli değerlerden büyük piksellerin seçilmesi ile, görsel üzerinde gruplandırma yapılarak hücrelerin ayrılması sağlanmıştır. Bu yöneme eşikleme (*thresholding*) ismi verilmektedir. Eşikleme ve Beer Lambert yasasının birlikte uygulanması sonucu elde edilen sonuç şekil 7'de görülebilir. Öncelikle *MATLAB* ortamında geliştirilerek eşikleme yapılmak istenen değerlerin belirlenmesi tamamlandıktan sonra, algoritma *Java* dilinde yazılarak *ImageJ* platformunda kullanılabilir hale getirilmiştir. Bu algoritma ile ayrılan hücreler daha sonra uzaklık haritalaması kullanan *watershed* algoritması kullanılarak hücre sayısının elde edilmesi sağlanmıştır. Tamamen doğru sonuç vermeyen algoritma düşük hata payı ile uygulanarak geliştirilme sonucu bir üründe kullanabilme potansiyelini ortaya koymuştur.

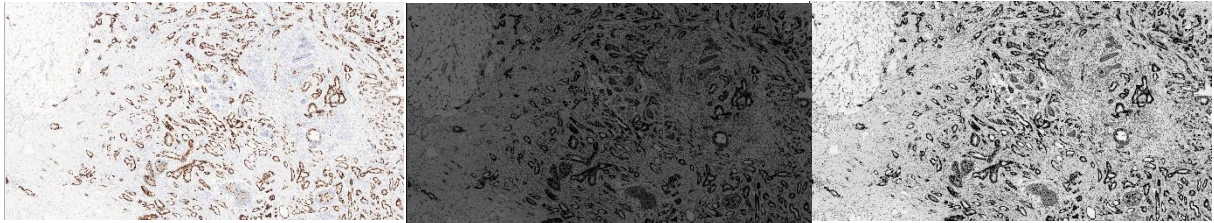


Şekil 7 - Orijinal resim (sol) ve Beer Lambert yasası ile eşikleme uygulanarak elde edilen kahverengi hücreler yeşil renk ile gösterildiği işlenmiş resim (sağ)

### 3.3 Gamma Dağılımı

*An Advanced Deep Learning Approach for Ki-67 Stained Hotspot Detection and Proliferation Rate Scoring for Prognostic Evaluation of Breast Cancer* (Saha, Chakraborty, Arun, Ahmed, Chatterjee, 2017) adlı kaynakta kullanılan yöntemde bulunan kaynak nokta belirleme (*seed point detection*) kısmını uygulayabilmek için görselde bulunan piksellerin histogramına uyum sağlanan birden fazla farklı gamma dağılımı kullanılmıştır. Gamma

dağılımı istatistiksel olasılık olarak  $\frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}$  şeklinde formüle edilir. Verilen formülde  $x$  olasılığın bulunmak istendiği veri değeri,  $k$  ve  $\theta$  istatistiksel dağılımın özelliklerini belirleyen değişkenlerdir, belirtilen gamma fonksiyonu ise  $\Gamma(n) = (n-1)!$  şeklinde formüle edilir. Gamma dağılımının optimize edilerek histograma uyumunun maksimumuma erişebilmesi için *Fast learning of Gamma mixture models with k-MLE* (Schawander, Nielsen, 2013) adlı yöntem kullanılarak gamma dağılımının optimizasyonu yapılmıştır. *Java* dili ile *ImageJ* ortamında yöntemi uygulayabilmek için digamma ve gamma fonksiyonunu yaklaşık olarak hesaplayabilen başka algoritmalar araştırılarak bu algoritmalar yazılan koda entegre edilmiştir. Ayrıca yöntemin uygulanması sırasında üç farklı kanal için üç farklı histogram üzerinde işlem yapılmış ve daha sonra elde edilen olasılıklar toplanarak tek bir olasılık dağılımı elde edilmiştir. Uygulanan yöntem sonucunda elde edilen gamma dağılımları, dağılım sayısından bağımsız olarak birbirine çok yakın olduğu için yararlı sonuçlar elde edilememiştir. Birbirinden bağımsız farklı gamma dağılımları elde edilerek, verilen piksel değerlerine göre bu dağılımlara ait olma olasılıkları hesaplanarak sınıflandırılma yapılabilmesi beklenmiştir. Ancak histogram üzerinde optimize edilen gamma dağılımı, dağılım sayısından bağımsız olarak elde edilen dağılımlar ve her piksel için elde edilen olasılık değerleri birbirine çok yakın olduğu için, sınıflandırma başılamamıştır. Elde edilen olasılık değerlerinin büyüklük olarak gösterimi (büyük değerler beyazla, küçük değerler siyah ile gösterilmiştir, veri sıfır ve iki yüz elli beş arasında normalleştirilmiştir.) şekil 8’de görülebilir.



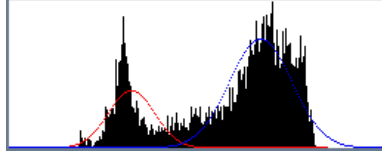
Şekil 8 - Orijinal resim (Sol), birinci gamma olasılık dağılımı (Orta), ikinci olasılık dağılımı (sağ)

Bu başarısız denemeden sonra, aynı yöntem gauss dağılım modeli ile tekrar denenmiş ve başarılı sonuçlar elde edilmiştir.

### 3.4 Gauss Dağılımı

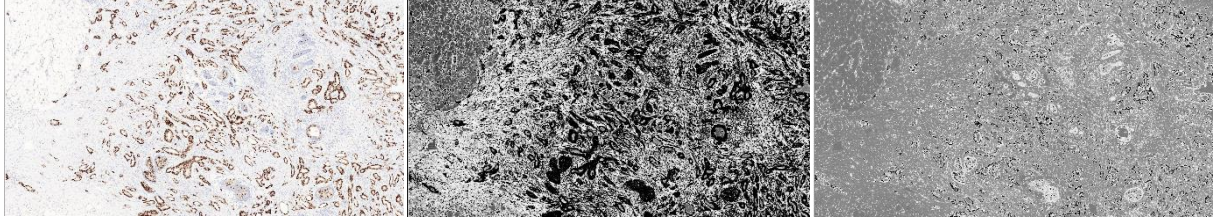
Gauss dağılımı istatistiksel olasılık olarak  $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  şeklinde formüle edilir. Verilen formülde  $x$  olasılığın bulunmak istendiği veri değeri,  $\sigma$  ve  $\mu$  istatistiksel dağılımın özelliklerini belirten değişkenlerdir. İki farklı gauss dağılımı işlenmek istenen görselin piksel değerlerinin histogramı üzerinde optimize ederek sınıflandırma yapılması amaçlanmıştır. Yapılan işlem şekil 9’da görülebilir. Ayrıca yöntemin uygulanması sırasında üç farklı kanal için üç farklı



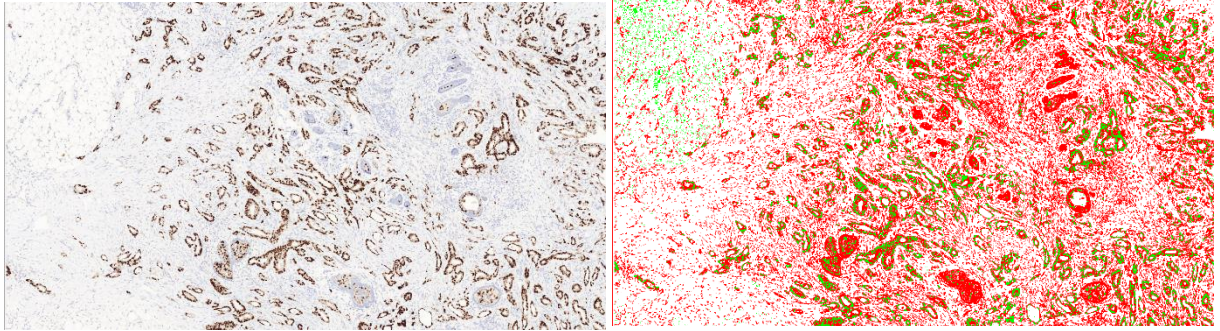


*Şekil 9 - Pikel değerlerinin histogramı üzerinde optimize edilen iki farklı gauss dağılımı (Kırmızı ve Mavi çizgiler gauss dağılımlarını ifade etmektedir)*

histogram üzerinde işlem yapılmış ve daha sonra elde edilen olasılıklar toplanarak tek bir olasılık dağılımı elde edilmiştir. Elde edilen olasılık dağılımları üzerinde verilen piksel değerlerine göre hesaplanan olasılık değerleri kullanılarak eşikleme (*thresholding*) yöntemi ile hücrelerin gruplandırılması tamamlanmıştır. Daha sonra elde edilen görsele uygulanan *watershed* algoritması ile hücrelerin ayrılması ve sayımı tamamlanarak yüksek doğrulukta sonuçlar elde edilmiştir. Elde edilen sonuçlar ve olasılık dağılımları şekil 10 ve şekil 11’de görülebilir.



*Şekil 10 – Orijinal resim (sol) birinci gauss olasılık dağılımı (orta) ikinci gauss olasılık dağılımı (sağ)*



*Şekil 11 – Orijinal resim (sol) gauss dağılımı ve eşikleme sonucu gruplandırılan pikseller (kırmızı ve yeşil olarak iki gruba ayrılmıştır. (sağ)*

Erişilmek istenen sonuç bu yöntemlerin çeşitli kimyasal boyama ile elde edilen dijital görseller üzerinde uygulanabilirliğini ve başarı oranlarını test etmek olduğu için, uygulanan yöntemlerin daha fazla geliştirilmesine gerek duyulmamıştır. Üzerinde test uygulanan görsellerden bir kısmı ekler bölümünde bulunabilir. Elde edilen sonuçlar ve şirket tarafından sağlanan görseller üzerinde yapılan testlerin sonuçları rapor olarak hazırlanarak staj bitiminde proje yöneticisine ve şirkete teslim edilmiştir.

## 4.SONUÇ

Bilgisayar biliminin benim için yeni olan dijital patoloji dalına adım attığım bu stajımda, görüntü işlemeye giriş bilgilerini elde edindim. Bana yardımcı olan *ViraSoft* çalışanları sayesinde özellikle görüntü işleme ve *ImageJ* programının kullanımı konusunda kendimi çok geliştirdim. Bu stajın hayatımın ilerleyen dönemlerinde mesleğimi şekillendirmem için bana çok yararı dokundu. Yeni kurulan bir firmada organizasyon ve araştırma geliştirme nasıl yapılır gözlemlemeyebildim. Özellikle daha önce uygulanmamış veya açık kaynak olarak internette bulunmayan algoritmaları pratiğe dökerek; hem araştırma hem makale okuma hem de yazılım konusunda kendimi çok geliştirdim. Verimli bir ay boyunca yazılım sektörünü ve bilgisayar biliminde araştırma geliştirme bölümünü yakından tanıma fırsatı bulmuş oldum.

## 5.REFERANSLAR

Girish V, Vijayalakshmi A (2004). *Affordable image analysis using NIH Image/ImageJ*. Indian Journal of Cancer (s. 47). PMID:15105580

Werner B (2000), *Writing ImageJ PlugIns – A Tutorial*

Ting C, Christophe C (2014). *Deep Learning Based Automatic Immune Cell Detection for Immunohistochemistry Images*. Machine Learning in Medical Imaging: 5th International Workshop. Proceedings. DOI: 10.1007/978-3-319-10581-9\_3

Monjoy S, Chandan C, Indu A, Rosina A, Sanjoy C (2017). *An Advanced Deep Learning Approach for Ki-67 Stained Hotspot Detection and Proliferation Rate Scoring for Prognostic Evaluation of Breast Cancer*. DOI:10.1038/s41598-017-03405-5

Olivier S, Frank N (2013). *Fast learning of Gamma mixture models with k-MLE*. Similarity-Based Pattern Recognition: Second International Workshop. DOI: 10.1007/978-3-642-39140-8\_16

## 6. EKLER

Görüntü işleme ve patoloji ile ilgili hazırlanan sunuma aşağıdaki adresten ulaşılabilir:

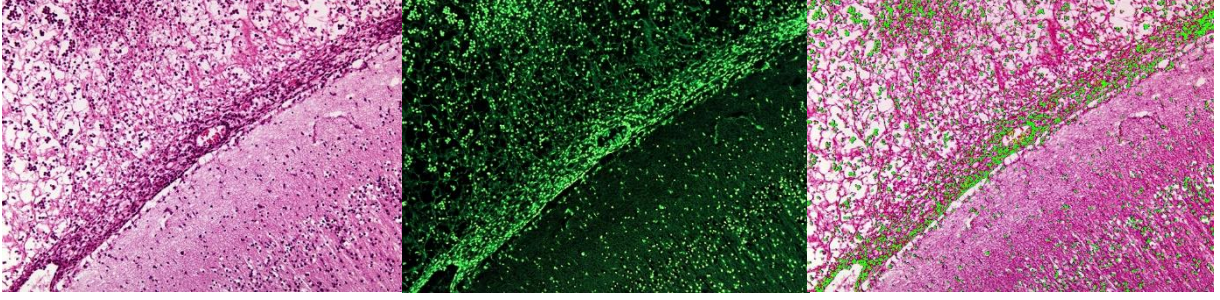
<https://ynsgnr.github.io/patolojiSunum.pdf>

Test yapılan görsellerden bazıları:

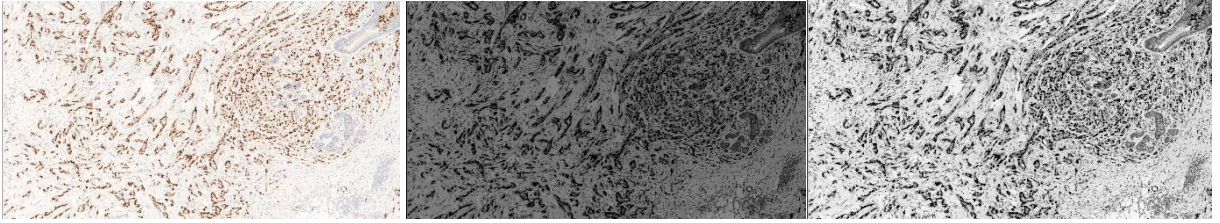


Ek 1 – Orijinal resim (sol) Beer Lambert yasası ile elde edilen görsel (orta) Beer Lambert yasası ve eşikleme ile elde edilen kahverengi hücreler (yeşile boyanmış pikseller) (sağ)

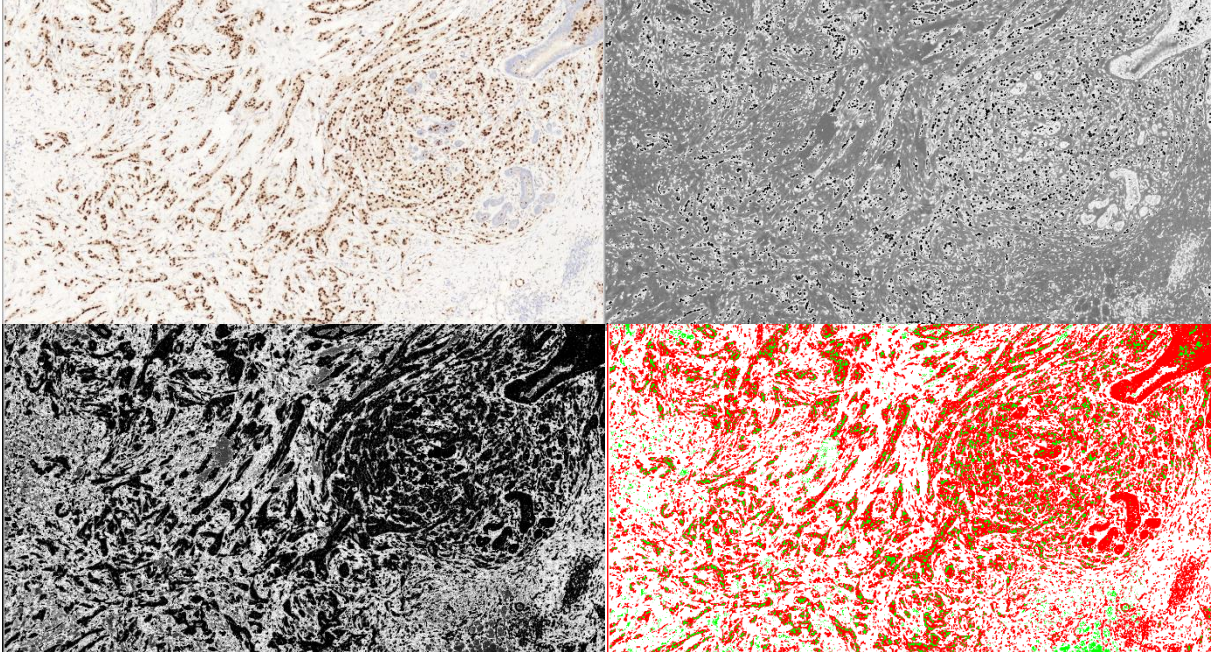




*Ek 2 – Orijinal resim (sol) Beer Lambert yasası ile elde edilen görsel (orta) Beer Lambert yasası ve eşikleme ile elde edilen kahverengi hücreler (yeşile boyanmış pikseller) (sağ)*

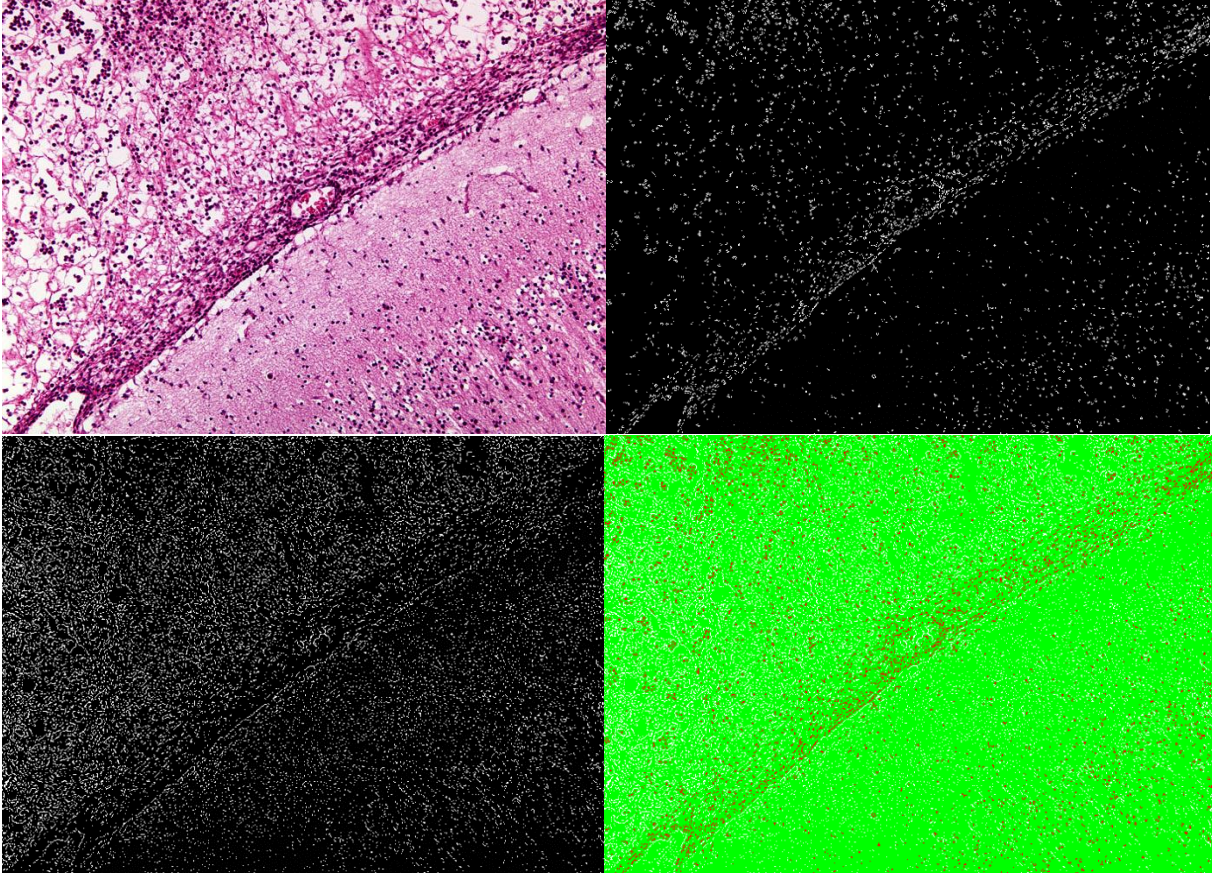


*Ek 3 – Orijinal resim (sol) birinci gamma olasılık dağılımı (orta) ikinci gamma olasılık dağılımı (sağ)*



*Ek 4 – Orijinal resim (sol üst) birinci gauss olasılık dağılımı (sağ üst) ikinci gauss olasılık dağılımı (sol alt) sınıflandırılmış pikseller (kırmızı ve yeşil olarak) (sağ alt)*





*Ek 5 – Orijinal resim (sol üst) birinci gauss olasılık dağılımı (sağ üst) ikinci gauss olasılık dağılımı (sol alt) sınıflandırılmış pikseller (kırmızı ve yeşil olarak) (sağ alt)*