

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

**Bilgisayarlı Patoloji ve Görüntü İşleme**

**STAJ**  
**Duhan Cem Karagöz**  
**150130066**

**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: Duhan Cem Karagöz  
Öğrenci Numarası: 150130066  
Bölüm: Bilgisayar Mühendisliği  
Program: Bilgisayar Mühendisliği (%30 İngilizce)  
E-posta Adresi: karagözdüh@itu.edu.tr  
(Cep) Tel No: +905376565524  
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 ANONİM ŞİRKETİ  
Birimi: Görüntü İşleme  
Web Adresi: virasoft.com.tr  
Kısa Adresi: Reşitpaşa Mahallesi, İTÜ Maslak Kampüsü, 34467 Sarıyer/İstanbul, Türkiye

**Yetkili kiři ile ilgili bilgiler**

Bölümü: Görüntü İşleme  
Unvanı: Proje Yöneticisi ve Görüntü İşleme Uzmanı  
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.....	1
3. STAJ PROJESİNİN TANIMI VE ANALİZİ.....	1
4. SONUÇ.....	8
5. REFERANSLAR.....	8

150130066 numaralı, Duhan Cem Karagöz adlı öğrencinin, yukarıda “İçindekiler” bilgisi verilen staj raporu, görülmüş ve uygun bulunmuştur.

Formu Dolduran Firma Yetkilisinin Adı ve Soyadı: *Sakir Karavi*

Yetkilinin Unvanı: *Görüntü İşleme Uzmanı, Proje Yöneticisi*

Müdür-İmza-Kaşe: *Gökhan HATİPOĞULLI*

(Kurumsal) E-posta: *info@virasoft.com.tr*  
(Kurumsal) Tel. No. : *0338 949 6125*

**VİRASOFT YAZILIM TİC.A.Ş**  
Bağcıbaşı Mh. Katar Cd. Kosgeb  
Blok No: 2/184/1 Sarıyer/İST.  
Sarıyer Tel. No: 0528222  
Ticaret Sicil No: 16725

## 1.KURULUŞ HAKKINDA BİLGİLER

Virasoft 2015 yılında hali hazırda bulunan patoloji tekniklerini bilgisayarlı hale getirmeye odaklanmış bir firmadır. 2017 yılına kadar piyasaya sürdükleri ürünler arasında LBYS (Laboratuvar Bilgi Yönetim Sistemi) bulunmaktadır. *LBYS* hastane içinde iletişimin ve laboratuvar da protokollerin izlenmesini kolaylaştırılması amaçlanmış bu sayede hem hastaya hemde hastaneye katkı sağlanması amaçlanmıştır. Firmanın başka bir ürünü ise *Telepath Mobile*dir. *Telepath* doktorlar arası vaka, anlık görüntü, muayene ve mikroskopik inceleme paylaşılması hedeflenmiş ve bu sayede doktorların birbirine danışmasını kolaylaştırarak hastalara daha iyi hizmet verilmesi istenilmiştir.

İşyerinde çalışmalar projeden projeye değışsede görevler proje başlangıcında proje yöneticisi tarafından dağıtılmaktadır. Her takımın kendi proje yöneticisi bulunmakta ve kendine takımının görevlerini dağıtmaktadır.

KOSGEB ofislerinde yer alan Virasoft 10 çalışana ev sahipliği yapmaktadır. Bahsi geçen on kişi görüntü işleme, android yazılım geliştirme, web tasarım, masaüstü uygulama geliştirme gibi çeşitli bölümlerde çalışmaktadır. Benim görev aldığım görüntü işleme takımının görevi doktorların kanser hastalarına teşhis koymak için kullandığı teknikleri bilgisayarlı hale getirmektir.

Virasoft KOSGEB ofislerinde iki ayrı ofise sahiptir. Ofislerden birisi toplantı, planlama gibi olaylarda kullanılmakta olup diğer ofis toplu çalışma ve yazılım geliştirme ofisi olarak kullanılmaktadır. Bu ofisler T.C. KOSGEB tarafından karşılık beklemeden Virasofta tahsis edilmiştir. Virasoft ayrıca İTÜ Arı Teknokent tarafından desteklenmektedir.

## 2. GİRİŞ

Virasoft Yazılım Tic. A.Ş.'de görüntü işleme takımında yaptığım 20 günlük stajım boyunca projelerin nasıl gerçekleştirildiğini, görev dağılımının nasıl yapıldığını, projenin bitirilmeden önce nasıl test edildiğini ilk elden görme imkanım oldu. Bir projenin nasıl seçildiğini, bir takım toplantısının nasıl gerçekleştiğini görme imkanım oldu. Daha önce yapılmayan bir programı teorik bir aşamadan nasıl gerçekleyebileceğimi öğrendim. Staj süresince tecrübe ettiğim olaylar bana ileriki çalışma hayatımda yardımcı olacaktır. Ayrıca bu staj ileriki çalışma yaşamımda hangi bölümde uzmanlaşmayı istediğimi seçmemi kolaylaştırdı.

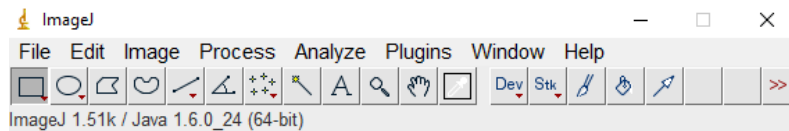
## 3. STAJ PROJESİNİN TANIMI VE ANALİZİ

Staja başladığımda ilk olarak yapılan projelere daha iyi uyum sağlamam, anlamam ve projeye daha verimli bir şekilde katkı sağlamam için firma tarafından kısa bir eğitim ve araştırma sürecine tabi tutuldum. Daha sonra tarafımdan bir sunum yapılması istendi ve bu sunuma göre şirket içinde bulunan görüntü işleme takımına atandım. Bu takımdan farklı olarak şirket içinde masaüstü uygulama geliştirme ve android uygulama geliştirme takımlarında bulunmaktadır. Görüntü işleme takımı ben staja başladığımda kanserli hücreleri normal

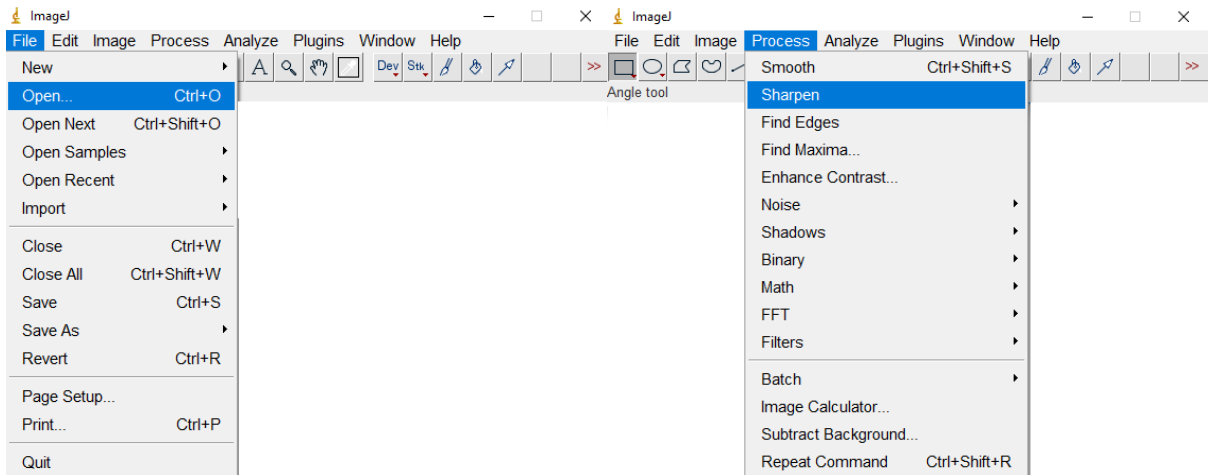
hücrelerden ayırıp, hastalığa bilgisayar yardımı ile tanı koymaya çalışıyordu. Bu nedenle proje sorumlusu stajyerlere projede yardım olabilecekleri araştırma konuları verip bu konularla ilgili görev üstlenmemizi istedi. Şirkette yazılan görüntü işleme projeleri Java diliyle üretilmiş olan *ImageJ* programı ve kütüphaneleri kullanılarak yazılmaktadır.

Java paralel iş yapabilen, güvenli, taşınabilir, nesneye dayalı ve hızlı bir programlama dilidir. Yüksek seviye bir programlama dili olması ve bir çok sistem tarafından desteklenmesi sebebiyle birçok programcı tarafından bilinen java çoğu programın temelini oluşturmaktadır. Ayrıca java tarafından sağlanan çöp toplayıcı kullanıcının işini kolaylaştırdığından ve karmaşık fonksiyonlar ve işlemler için bulunan kütüphaneleri nedeniyle çoğu programcının tercihi olmayı başarmıştır. *Çöp Toplayıcıyı* açıklamak gerekirse java kullanıcının yazdığı programdaki kullanımı biten ve hafızada yer kaplayan değişkenleri kullanıcı etkileşimi olmadan hafızadan kaldırmayı sağlayan bir yardımcı dahili sistemdir. Java dilinin farklı bilgisayarlarda ve mimarilerde çalışmasını sağlayan özelliği ise Java'nın kendine has olan *Java Çalışma Zamanı Ortamıdır*. Bu ortamda java çalıştırma zamanında derlenir ve çalıştırılır. Bu derlenmede java makine koduna değil *bytekoda* çevrilir daha sonra *Java Sanal Makine* ortamında çalıştırılır.

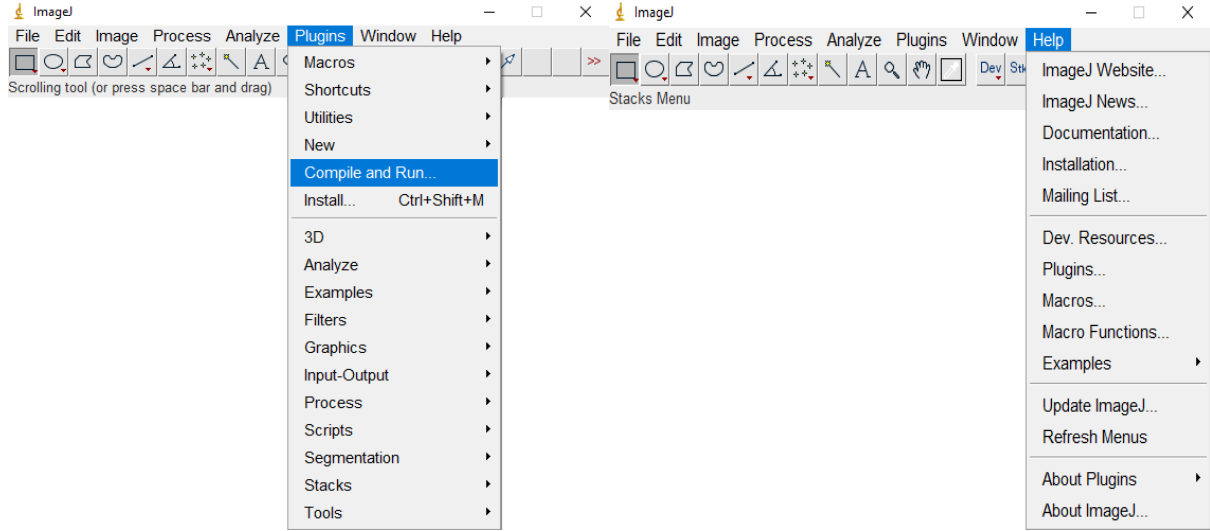
*ImageJ* programı ise java programlama dili ile geliştirilmiş olup bir çok formatta olan medikal görüntülerin işlenmesi için *NIH(National Institutes of Health)* tarafından açık kaynak kodlu olacak bir şekilde geliştirilmiştir. ImageJ, java ile paket şeklinde veya javadan bağımsız bir şekilde indirilebilir. ImageJ içinde bulunan fonksiyonları java kaynak kodu içerisinde makro şeklinde kullanılmasını destekliyor. Ayrıca ImageJ java kodlarını ve yolu belirlenmiş ayrı kütüphaneleri derleyip, çalıştırabiliyor. Medikal görüntü işleme aracı olmasının getirdiği beklenti ile 8-bit, 16-bit, 32-bit gibi ve TIFF,JPEG,GIF,FITS,DICOM gibi birçok resim formatını açıp işleyebilmektedir. Ayrıca paralel işlemleri ve yığın resimleri destekleyen ImageJ birçok yazılımcının görüntü işleme için tercihi olmaktadır. ImageJ kesme, opaklık değiştirme, eşik değiştirme, uzaklık haritası, medyan gibi bir çok görüntü işleme fonksiyonunu kolaylıkla gerçekleyebiliyor (Girish V, Vijayalakshmi A, 2004, ss.47). ImageJ <https://imagej.nih.gov/ij/download.html> adresinden indirilebilir.



Şekil-1. ImageJ kullanıcı arayüzü



Şekil-2. ImageJ dosya arayüzü (Sol), ImageJ işlemler arayüzü (Sağ)



Şekil-3. ImageJ Eklenti arayüzü (Sol), ImageJ Yardım ve örnekler arayüzü (Sağ)

Program ilk olarak çalıştırıldığında Şekil-1'deki arayüz açılır. Proje kodunu çalıştırmak için ilk önce resim Şekil-2 (Sol)'de görüldüğü gibi dosya açılır. Daha sonra Şekil-3 (Sol)'te görüldüğü gibi plugins sekmesinde Compile and Run seçeneği seçilir ve projenin bulunduğu konuma gidilerek java dosyası seçilir ve çalıştırılır. Şekil-2 (Sağ)'de görülen seçeneklerin tamamı java kaynak kodu içerisinde makro olarak çağrılıp kullanılabilir. Şekil-3 (Sağ)'te görülen seçenekler programın güncellenmesi ve çeşitli örnekler bakılmak için kullanılabilir.

Staj süresince java programlama diline geçmeden önce proje yöneticisi tarafından önerildiği gibi *MATLAB* kullanarak görüntü işleme algoritmalarını denedim daha sonrasında bu algoritmaları Java dilinde tekrardan gerçekleştirdim. *MATLAB* (*Matrix Laboratory*) MathWorks tarafından geliştirilmiş olup dil içindeki tüm işlemler matris üzerinden gerçekleşir. Matematiksel işlemler için üretilmiş olmasına rağmen MathWorks tarafından sağlanan çeşitli eklentiler (toolbox) sayesinde makine öğrenmesi, sinyal işleme, grafiksel simülasyon gibi çeşitli alanlarda kullanılabilir. Virasoft şirketinde yaptığım stajda *MATLAB 9.0.0.341360-R2016a* versiyonunu kullandım.

Staj başlangıcında sunumu yapmak için *Microsoft Office PowerPoint 2013* kullandım ayrıca şirket tarafından stajyerlerden istenen haftalık raporu yazmak ve *PDF* versiyonuna çevirmek için *Microsoft Office Word 2013* kullandım. Ayrıca yazdığım kodların akış şemasını çizmek ve proje yöneticisine göndermek için <https://www.draw.io> sitesini kullandım. Yaptığım projede yazdığım kodlar şirkete ait olduğundan ve şirket tarafından izin verilmediğinden herhangi bir kod veya akış şeması paylaşamayacağım.

### 3.1 Görüntü İşleme ve Patoloji Eğitimi

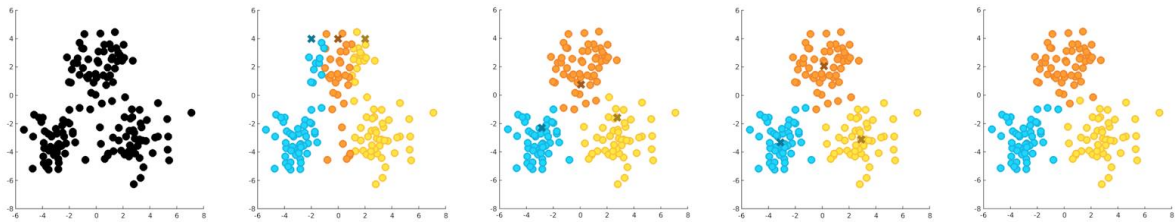
Stajyer olarak bana verilen ilk görev tarafıma sağlanan makaleleri okumam ve bu makalelerden bir sunum hazırlamamdı. Bana verilen makalelerin tamamı patolojik tanı koyma yöntemleri ve bu yöntemlerin nasıl bilgisayarlı ortamda gerçekleştirilebileceği hakkındaydı. Daha sonra görüntü işleme takımındakilere yöntemler, algoritmalar ve programlar hakkında sorular sorup bilgi aldık ve görüntü işleme hakkında bilgimizi arttırdık. Sunum içerisinde tanı konulması hakkında bilgiler, kanser hakkında bilgiler, lamin nasıl işlenip boyanacağı, bilgisayarlı patolojinin patolojideki yeri, bilgisayarlı patolojinin yararları, tanı yöntemleri ve

aşamaları, Gauss filtresi, top-hat filtresi, laplas filtresi, medyan filtresi, morfolojik operasyonlar, watershed algoritması gibi filtreler ve algoritmalar açıklanmaktadır. Sözü geçen sunum <http://cemkaragoz.me/files/Slide.pptx> indirilebilir. Sunum stajın ikinci haftasında pazartesi günü yapılmıştır. Sunumdan sonra görüntü işleme grubuna dahil edildim. Proje yöneticisi görüntü işleme ve java hakkında daha derin araştırma yapmamızı ve tanı koymak için yapılması gereken işlemleri araştırmamızı istedi. Bilgisayarlar ile tanı koymak için yapılması gereken aşamalar; arka ve ön planı ayırıp, geriye kalan resmi gruplayarak hücreleri saymak olmuştur.

Slayt dahilinde olmasa bile görüntü işleme ile ilgili başka konularda araştırmaya dahil edilmiştir. *RGB (Red, Green, Blue)* renk uzayı kırmızı, yeşil ve mavi renklerin ingilizce baş harflerinden oluşmakta olan bir renk uzayıdır (Hunt, R. W. G, 2004). Bu uzayda her piksele kırmızı, yeşil ve mavi renk kuvvetine göre 0 ile 255 üç farklı değer atanmaktadır. RGB uzayı en çok kullanılan uzaylardan biridir. Görüntü işleme işlemlerinde ikinci bir renk uzayı ise LAB renk uzayıdır. LAB renk uzayı parlaklık, yeşil-kırmızı, mavi-sarı değerlerini kullanıp pikselleri temsil eder (Hunter, Richard Sewall, 1948). Ayrıca slaytta yer bulamasada histogramda araştırmaya dahil edilmiştir. Histogram ise bir görüntünün renklerinin grafiksel olarak tablo halinde temsildir (Ed Sutton, n.d.).

### 3.2 K-Means Kümeleme

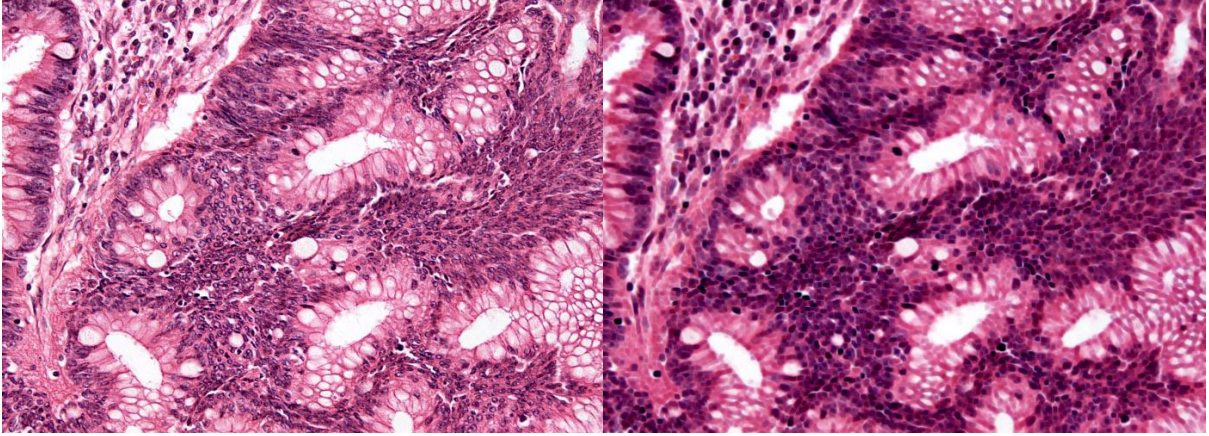
Proje yöneticisi tarafından proje takımına dahil olduktan sonra bana verilen görev k-means kümeleme yöntemini araştırmam ve ImageJ için bir eklenti yazmam oldu. *K-Means kümeleme* yöntemine göre birbirine yakın özellikte bulunan data noktaları aynı gruba ait olmalıdırlar. Daha basit olarak *K-Means kümeleme* basit olarak bir data grubunu belirli bir değere göre K adet kümeye ayırmaktır (Nameirakpam D, Khumanthem M and Jina C, 2015).



Şekil-4. K-Means örneği. Data grubu kordinatlarına göre 3 farklı gruba ayrılmıştır.

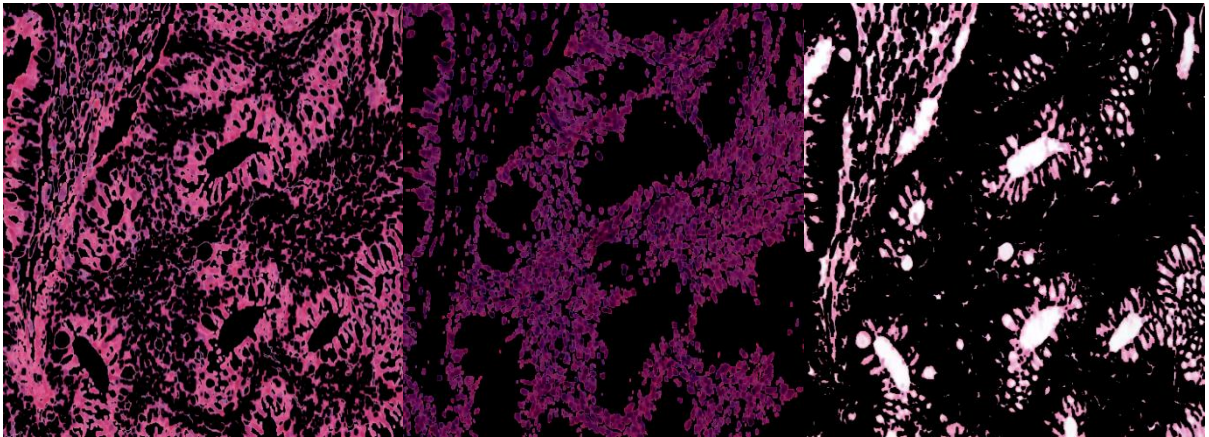
Resimlere *K-Means Kümeleme* yöntemini uygulamak için en iyi data grubu her pikselin sahip olduğu *RGB* değeri ile en yakın değerlere sahip olan noktalar aynı gruba atanarak *K-Means* yöntemi uygulanabilir. Yöntemi resim üstünde uygulamak için ilk önce küme sayısı kadar rastgele veya daha önceden belirlenmiş bir şekilde küme merkezleri atanır. Daha sonra her nokta ile her kümenin merkezine olan öklit uzaklıkları  $\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$  formülü kullanılarak hesaplanır. Böylece uzaklık matrisi oluşturulur. Daha sonra her nokta olan uzaklıklarına göre en yakın oldukları gruplara atanır. Daha sonra grup merkez noktaları tekrardan hesaplanarak işleme tekrardan başlanır. Bitirme koşulu projeden projeye değişikliğe uğrasada bu projede kullanılan bitirme koşulu merkez noktaların döngü içinde sabitlenmesidir (Nameirakpam D, Khumanthem M and Jina C, 2015). K-Means algoritması Şekil-4 üzerinde 3 farklı küme oluşturacak şekilde uygulanırsa ortaya keskin bir şekilde ayrılmış üç farklı küme oluşur.





Şekil-5. Orjinal resim (Sol), medyan ve keskinlik filtresi uygulandıktan sonraki resim (Sağ)

Medikal resimlerde lam boya bulaşma hatası olabileceğinden ve elimizdeki resmi en iyi haline çevirebilmek için Şekil-5’de görülen resme ilk olarak medyan filtresi uygulanarak resimdeki uç pikseller yumuşatılır. Daha sonra resme keskinlik filtresi uygulanarak resmin renkleri daha kuvvetlendirilir. Daha sonra K-Means algoritması resim üzerine uygulanır ise Şekil-6’de görülen üç farklı kümeye ayrılır.



Şekil-6. Birinci küme elemanları (Sol), İkinci küme elemanları (Orta), Üçüncü Küme elemanları (Sağ)

### 3.3 Fuzzy C-Means Kümeleme

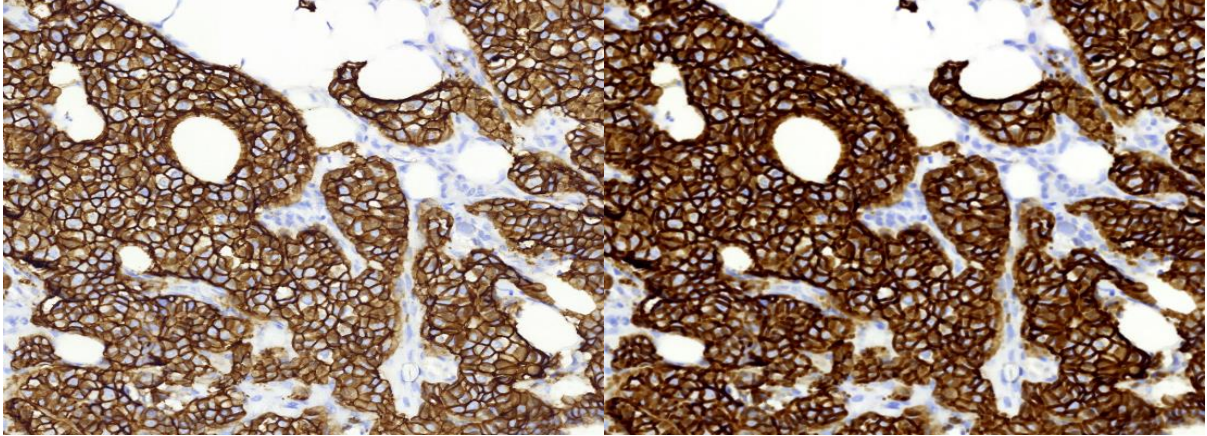
K-Means Kümelemeyi tamamladıktan sonra proje yöneticisi *Fuzzy C-Means Kümeleme* yöntemini araştırmamı ve ImageJ programına yönelik bir eklenti yazmamı istedi. *Fuzzy C-Means Kümeleme* yöntemi *K-Means Kümeleme* yöntemine temelde benzerse de birçok noktada farklılık göstermektedir. En temel farklılıktan bahsetmek gerekirse *K-Means Kümeleme* yönteminde her piksel sadece bir kümeye ait olabilirken *C-Means Kümeleme* yönteminde bir piksel aitlik değerine göre birden fazla kümeye ait olabilir. Ayrıca *C-Means Kümeleme* yönteminde uzaklık tek başına grup aitliğini belirlemeye yetmiyor. Bunun yerine  $u_{ik} = \frac{1}{\sum_{l=1}^c \left( \frac{|p_i - v_k|}{|p_i - v_l|} \right)^{\frac{2}{m-1}}}$  fonksiyonu her pikselin her gruba aitlik değerini bulmak için kullanılıyor.

Daha sonra önceden belirlenen aitlik sınırına göre pikseller gerekli gruplara dağıtılır. Daha sonra aitlik değerlerine göre dağıtılmış pikseller  $J = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m \times |p_i - v_k|^2$  formülü kullanılarak hata hesaplanır ve belirlenmiş hatanın altında ise döngü sonlandırılır. Eğer hata beklenen değerden yüksek ise döngü hata belirli bir düzeyin altına inene veya döngü sınırına

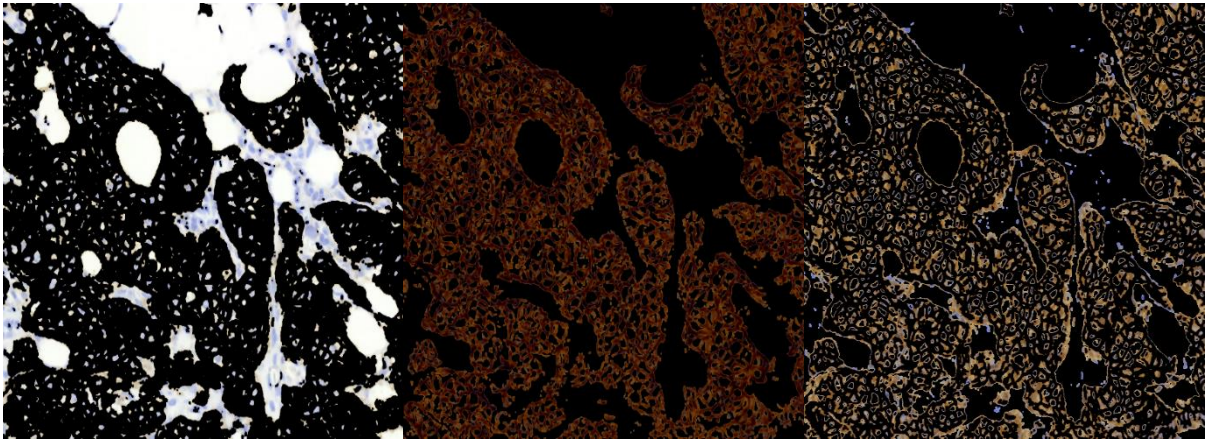


gelene kadar devam eder. Kümelerim merkez noktalarını tekrardan hesaplamak için ise K-Means Kümeleme yönteminden farklı olarak  $v_k = \frac{\sum_{i=1}^n u_{ik}^m \times p_i}{\sum_{i=1}^n u_{ik}^m}$  formülü kullanılır. Formüllerde kullanılan semboller aşağıda açıklanmıştır:

- j hata fonksiyonu,
- n E resminde bulunan piksel sayısı,
- c küme sayısı,
- u fuzzy aitlik değeri,
- m fuzziness değeri ( $m > 1$ ),
- $p_i$  resimdeki i. piksel,
- $v_k$  k. kümenin merkez noktası.



Şekil-7. Orjinal resim (Sol), keskinlik filtresi uygulandıktan sonraki resim (Sağ)



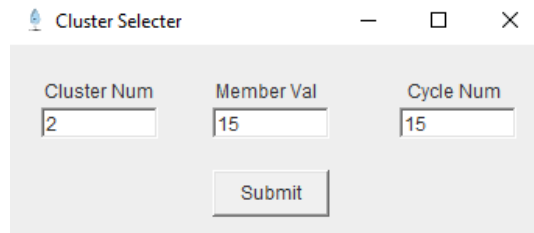
Şekil-8. Fuzzy C-Means uygulandıktan sonraki ilk küme (Sol), ikinci küme (Orta), üçüncü küme (Sağ)

Şekil-7 (Sol)'daki resime keskinlik filtresi uygulandıktan sonra Şekil-7 (Sağ)'daki resim elde edilir. Sonrasında resime yukarıda bahsi geçen formüller uygulanır daha sonra pikseklerin aitlik değerlerini tutan c boyutlu matris değerlerine ve aitlik sınırına göre Şekil-10'da görüldüğü gibi gruplara atanır. Daha sonra Şekil-8'daki gibi kümeler yan yana kullanıcıya gösterilir ve sonuç kayıt edilir.

Daha sonra proje kapsamında C-Means Kümeleme kullanılmak üzere kullanıcı arayüzünü oluşturmak amaçlı Java *Swing* kütüphanesini kullandım. Bu arayüzde küme sayısı, aitlik sınırı ve döngü sınırını kullanıcı tarafından kolayca değiştirilebilmesini amaçladım.

Swing kütüphanesi Java dilinde temel grafik arayüz tasarlama aracıdır. Swing kütüphanesinde bir pencere JFrame, JPanel, JLabel, JButton ve JTextField gibi yapı bloklarından oluşmaktadır. Her bir blok ayrı göreve sahiptir. JFrame ve JPanel barındırıcı bloklar, JLabel, JButton ise arayüz bloklarıdır. Bu arayüz bloklarının kullanıcı ile etkileşimini izlemek için Swing kütüphanesi ActionListener fonksiyonunu kullanıma sunmuştur. Bu fonksiyon kullanılarak kullanıcı JButton'a bastığında kod içerisinde belirlenen bir başka fonksiyona çağrı yapılabilir veya bir geri dönüş verilebilir. Swing kütüphanesini java kaynak kodunda kullanmak için ilk önce javax.swing ve gerekli JFrame gibi kullanılan sınıfların kaynak koda eklemek gerekiyor.

Şekil-9'de ise bu kütüphaneleri kullanıp oluşturulan kullanıcı arayüzünü gösterilmektedir. Bu ara yüzden küme sayısı, minimum aitlik değeri ve maksimum döngü sayısı alınmakta olup ekrana varsayılan değerler ile gelmektedir.



Şekil-9. Kullanıcı arayüzü Küme, Aitlik ve Döngü seçim ekranı

#### 4. SONUÇ

Staj sonrasında daha önce uğraşmadığım bir alan olan görüntü işleme ile ilgili çok şey öğrendim. Birçok çalışanla konuşup meslek geleceğim ile ilgili tavsiyeler aldım. Girişim seviyesindeki bir şirket nasıl yönetildiğini ve bu şirketteki projeler nasıl ele alındığını gözlemledim. Girişim seviyesindeki bir şirketin pazara nasıl açıldığını gözlemledim. Daha önceden bulunmayan algoritma ve prosedürlerin koda dökülüp nasıl gerçekleştirildiğini ilk elden deneyimledim.

#### 5. REFERANSLAR

Ed Sutton. (n.d.). Histograms and the Zone System, in Illustrated Photography.

Girish V, Vijayalakshmi A. (2004). Affordable image analysis using NIH Image/ImageJ, in Indian Journal of Cancer (ss. 47).

Hunter, Richard Sewall. (July 1948). Photoelectric Color-Difference Meter. JOSA. 38 (7): 661. (Proceedings of the Winter Meeting of the Optical Society of America).

Hunt, R. W. G (2004). The Reproduction of Colour (6th ed.). Chichester UK: Wiley–IS&T Series in Imaging Science and Technology.

Nameirakpam Dhanachandra, Khumanthem Manglem and Yambem Jina. (2015), Chanu Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm, in Procedia Computer Science, Volume 54, (ss. 764-771).

Nikhil R. Pal, James M. Keller, and James C. Bezdek. (2005). A Possibilistic Fuzzy c-Means Clustering Algorithm in IEEE TRANSACTIONS ON FUZZY SYSTEMS, Volume 13, (ss. 517-530).