



T.C.  
İSTANBUL ÜNİVERSİTESİ – CERRAHPAŞA  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ |



YÜKSEK LİSANS TEZİ

AKCİĞER HASTALIKLARININ TANISINDA KULLANILAN X-IŞINI  
GÖRÜNTÜLERİNE YAPAY ZEKA VE SEGMENTASYON  
UYGULAMASI

Audry SHOMA

Biyomedikal Mühendisliği Anabilim Dalı  
Biyomedikal Mühendisliği Programı

DANIŞMAN

Dr. Öğr. Üyesi Abdurrahim AKGÜNDOĞDU

Şubat, 2021

İSTANBUL

Bu çalışma 10.03.2021 Tarihinde ařağıdaki jüri tarafından  
Biyomedikal Mühendisliğı Anabilim Dalı, Biyomedikal Mühendisliğı Programı Yüksek Lisans  
Tezi olarak kabul edilmiştir.

### TEZ JÜRİSİ

Dr. Öğr. Üyesi Abdurrahim AKGÜNDOĞDU  
İstanbul Üniversitesi – Cerrahpařa  
Mühendislik Fakültesi

Doç. Dr. Yasin ÖZÇELEP  
İstanbul Üniversitesi – Cerrahpařa  
Mühendislik Fakültesi

Doç. Dr. Serap ÇEKLİ  
Maltepe Üniversitesi  
Mühendislik ve Doğa Bilimleri Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa’nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

## ÖNSÖZ

Tezim boyunca beni desteklediđi ve yönlendirdiđi için babam Joseph SHOMA ve annem Brigitte NTUMBA TSHIMANGA'ya içten teşekkürlerimizi iletiyorum. Çalışmalarımı finanse ettikleri ve beni kim olduğum için feda ettikleri için her zaman ve sonsuza dek minnettar olacağım.

Ayrıca danışmanım Dr. Öğr. Üyesi Abdurrahim AKGÜNDOĞDU, çalışmalarımnda rehberlik ve danışmanlık hizmeti için teşekkür ederim.

Şubat 2021

Audry SHOMA



# İÇİNDEKİLER

ÖNSÖZ.....	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ .....	vi
TABLO LİSTESİ .....	vii
KISALTMALAR LİSTESİ .....	viii
ÖZET.....	ix
SUMMARY.....	x
1. GİRİŞ.....	1
2. GENEL KISIMLAR .....	3
2.1. Röntgen.....	3
2.2. Akciğer tüberküloz.....	3
2.3. Makine Öğrenmesi .....	4
2.4. Bölütleme.....	16
3. MATERYAL VE YÖNTEM .....	18
3.1. Veri Setleri .....	19
3.2. Bölütleme.....	19
3.3. Özellik çıkarma.....	21
3.4. Sınıflandırma .....	22
4. SONUÇLAR.....	22
4.1. Bölütleme.....	22
4.2. Özellik çıkarma.....	24
4.3. Sınıflandırma .....	25
5. TARTIŞMA .....	29
6. SONUÇ VE ÖNERİLER .....	30
KAYNAKLAR.....	31
EKLER.....	35

## ŞEKİL LİSTESİ

Şekil 2.1	: Denetimli ML akış diyagramı.....	7
Şekil 2.2	: Eğitim seti için bir karar ağacı örneği.....	8
Şekil 2.3	: İleri beslemeli YSA örneği.....	10
Şekil 2.4	: k-en yakın komşu sınıflandırması.....	10
Şekil 2.5	: İki sınıflı bir veri setinin bölgelerini ayıran sınırlar.....	12
Şekil 2.6	: İki sınıflı bir veri setini ayıran düzlemler.....	13
Şekil 2.7	: Üç küme için üç ağırlık vektörü.....	15
Şekil 2.8	: Kısmi ve Hiyerarşik kümelerin gösterimi.....	18
Şekil 3.1	: Görüntü analizine ait blok diyagram.....	18
Şekil 3.2	: Bölütlendirme adımları.....	19
Şekil 4.1	: RGB ve Gri tonlardaki CXR görüntülerine ait bölütleme uygulanması (a-h).....	23
Şekil 4.2	: Üçüncü seviye Dalgacık ayrışması sonuçları (a-d).	24
Şekil 4.3	: Eğitimde kullanılan veri setinin bir kısmı.....	24
Şekil 4.4	: Farklı sınıflandırıcıların doğruluk oranlarının karşılaştırılması.....	25
Şekil 4.5	: Montgomery veri seti için eğitim sonrası elde edilen ROC eğrileri (a-c).....	28

## TABLO LİSTESİ

Tablo 2.1	: Örnek veri seti.....	8
Tablo 2.2	: Örnekler (x ve y) arasındaki mesafeyi belirleyen yaklaşımlar.....	11
Tablo 4.1	: Holdout doğrulama için sınıflandırma algoritmalarının karışıklık matrisi	26
Tablo 4.2	: 5 kat çapraz doğrulama için sınıflandırma algoritmalarının karışıklık matrisi	26
Tablo 4.3	: Holdout doğrulama ve çapraz doğrulama üzerinde SVM, KNN, Karar Ağaçları ve topluluk alt uzayının sınıflandırma performansı ölçüsü.....	27
Tablo 4.4	: Eğitilmiş modelin duyarlılığı ve özgüllüğü .....	27



## KISALTMALAR LİSTESİ

AUC	: Eğri altında kalan alan
BK	: Koch basili
CAD	: Bilgisayar destekli teşhis
CT:	: Bilgisayarlı Tomografi
CXR:	: Göğüs röntgeni
DICOM	: Tıpta dijital görüntüleme ve iletişim
FN	: Yanlış negatif
FP	: Yanlış pozitif
HIV/AIDS	: İnsan bağışıklık eksikliği virüsü/ edinilmiş Bağışıklık Yetmezlik Sendromu
KNN	: k- En yakın komşuluk
LDA	: Doğrusal diskriminant analizi
ML	: Makine öğrenmesi
RV	: Olasılık oranı
RV-	: Negatif olasılık oranı
RV+	: Pozitif olasılık oranı
Se	: Duyarlılık
Sp	: Özgüllük
SVM	: Destek Vektörleri Makinesi
TB	: Tüberküloz
TN	: Gerçek negatif
TP	: Gerçek pozitif
YCbCr	: Luma bileşeni, mavi fark ve kırmızı fark

## ÖZET

### YÜKSEK LISANS TEZİ

#### AKCİĞER HASTALIKLARININ TANISINDA KULLANILAN X-IŞINI GÖRÜNTÜLERİNE YAPAY ZEKA VE SEGMENTASYON UYGULAMASI

**Audry SHOMA**

**İstanbul Üniversitesi – Cerrahpaşa**

**Lisansüstü Eğitim Enstitüsü**

**Biyomedikal Mühendisliği Anabilim Dalı**

**Danışman: Dr. Öğr. Üyesi Abdurrahim AKGÜNDOĞDU**

Tüberküloz her yıl dünya çapında yaklaşık 10 milyon insanı etkilemektedir. Genel olarak hekim ve özellikle radyolog eksikliği nedeniyle, Göğüs röntgeni (CXR) görüntülerinin yorumlanması bir sorun olmaya devam etmektedir. Bunun üzerine, CXR görüntülerini verimli bir şekilde yorumlayabilen ve tüberkülozu teşhis edebilen bir uygulama geliştirme hedeflerini kendimize belirledik. Montgomery veri setinin kullanıldığı bu çalışmada: 138 CXR görüntüsü (80 normal ve 58 tüberküloz). Bu uygulamanın geliştirilmesi için farklı segmentasyon yöntemleri ve yapay zeka (makine öğrenimi) gerekiyordu. Segmentasyon adımı ile tüberküloz lezyonları izole edildi. Tüberküloz lezyonların ve CXR görüntülerinin özelliklerini çıkarmak için bir özellik çıkarma adımı gerçekleştirildi. Çıkarılan özellikler, makine öğrenimi kullanılarak sınıflandırıldı. Tüberküloz lezyonlar, siyah akciğer yüzeylerinde beyaz olarak görülmektedir. En iyi sınıflandırma, topluluk alt uzay yöntemi ile elde edilmiştir. Sadece eğitim yapıldığında, doğruluk mükemmelliğe denk bir değer olan 1.00'a ve eğri altındaki alan da (AUC) 1.00'a ulaşmıştır. Bununla birlikte holdout (eğitim ve test) yöntemi sonrası doğruluk % 88.2 ve AUC 0.90 olmuştur. Çapraz doğrulama ile elde edilen en iyi sınıflandırıcı SVM (doğruluk=72.50 ve AUC = 0.69) olmuştur. Uygulamamız belirlenen hedeflere ulaşmış ve sorunumuza cevap vermiştir.

**Anahtar kelimeler:** Akciğer, tüberküloz, göğüs röntgen görüntüsü, makine öğrenme, bölütleme, sınıflandırma, topluluk altuzay, makine vektör desteği.

Şubat 2021, 56 sayfa

## **SUMMARY**

**M.Sc. THESIS**

# **APPLICATION OF ARTIFICIAL INTELLIGENCE AND SEGMENTATION TO X-RAY IMAGES USED IN THE DIAGNOSIS OF LUNG DISEASES**

**Audry SHOMA**

**Istanbul University - Cerrahpasa**

**Institute of Graduate Studies**

**Departement of Biomedical Engineering**

**Supervisor: Assist. Prof. Dr. Abdurrahim AKGÜNDOĞDU**

Every year tuberculosis affects nearly 10 million people around the world. With a shortage of medical doctor in general and particularly of radiologists, the interpretation of CXR images remains a problem. Thereupon we set ourselves the goals of developing an application capable of efficiently interpreting CXR images and diagnosing tuberculosis. We used Montgomery dataset: 138 CXR images (80 normal and 58 tuberculosis). The development of this application required different methods of segmentation and artificial intelligence (machine learning). With the segmentation step we isolated the tuberculous lesions. A feature extraction step was required to extract the characteristics of the tuberculous lesions and CXR images. The extracted features has been classified by using machine learning. Tuberculous lesions are shown in white on black lung surfaces. The best classifier has been achieved with ensemble subspace method. Without validation the accuracy reached the perfection 1.00 and an area under the curve (AUC) of 1.00. However with holdout validation accuracy was 88.2% with an AUC of 0.90. The best classifier achieved by cross validation was SVM (accuracy = 72.50 and AUC = 0.69). Our application has achieved the objectives set and responds to our problem.

**Keywords:** Lung, tuberculosis, chest x-ray image, machine learning, segmentation, ensemble subspace, classification, support vector machine.

Febuary 2021, 56 pages

## 1. GİRİŞ

Dünya genelinde her yıl yaklaşık 10 milyon kişi tüberküloz (TB) hastalığına yakalanmaktadır, bunlardan yetişkinlerin sayısı çocuklardan daha fazladır ve cinsiyet bakımından da erkeklerin sayısı kadınlardan daha çoktur. TB insan ölümünün en önemli 10 nedeninden biridir. Aynı zamanda HIV / AIDS 'in üzerinde sıralanan tek bir enfeksiyöz ajanın önde gelen ölüm nedenidir [1, 2]. Dünya nüfusunun önemli bir kısmı akciğer tüberkülozundan etkilenir.

Akciğer tüberkülozu tanısını yapmak için röntgen görüntülerinin yorumu hem uzman radyolog hem de diğer tıbbi personel için kolay değildir.

Çalışmanın amacı yapay zeka uygulaması tasarımıyla akciğerin uygun bir göğüs röntgen görüntülerini elde etmek ve akciğer tüberkülozunu tespit etmektir. Akciğer tüberkülozu tanısını yapmak için röntgen görüntülerinin yorumu hem uzman radyolog hem de diğer tıbbi personel için kolay değildir. Böylece klinisyenler göğüs röntgen görüntüsünden akciğer tüberkülozunu yorumlanması ve tanısında zaman tasarrufu yapacaktır ve herhangi bir radyoloğun yardımı olmadan da doğru bir yorum elde edebilmeye izin verebilecektir.

Yapay zeka uygulamasını tasarlamak için birkaç makine öğrenme tekniklerini bilmemiz gerekir. Makine öğrenmesi, makinelerin programlanmadan öğrenmelerini sağlamak anlamına gelir. Makine öğreniminin amacı makinelerin tahminler yapmasını, ilişkilendirme kurallarını çıkarmasını veya verilen bir veri setinden kararlar almasını sağlamaktır.

Makine öğrenmesini kullanmadan önce birkaç ön işlem vardır, bunlar: Bölütleme (segmentasyon) ve özellik çıkarma. Bölütleme, görüntünün bölgelere veya kategorilere ayrılmasıdır, farklı nesnelere veya nesnelere parçalarına karşılık gelir. Bir görüntüdeki her piksel bu kategorilerden birine tahsis edilir.

Dijital göğüs röntgeninin (CXR) segmentasyonu ve sınıflandırılması, tıbbi uygulamadaki yeri her zaman önemli olmuştur. Van Ginneken ve arkadaşlarına göre, göğüs radyolojisinde bilgisayar destekli tanı (CAD) konusunda yapılan ilk çalışmadan 45 yıl sonra bile, göğüs radyografilerini tam doğru okuyabilen bir sistem yoktur [3]. Otomatik nodül segmentasyonu ve sınıflandırma, CXR ve bilgisayarlı tomografi (CT) için karar destek / otomasyonun daha olgun uygulamalarından biri haline gelmektedir [4, 5]. Sonuç olarak, otomatik nodül segmentasyonu ve sınıflandırma sistemleri, radyologlara akciğer hastalıklarını teşhis etmede başarılı bir şekilde yardımcı olabilir [4].

Akciğer alanının segmentasyonu CXR'lerin uygun bir şekilde değerlendirilebilmesi için herhangi bir CAD sisteminin desteklemesi gereken önemli bir görevdir. Yardımcı olabilecek diğer segmentasyonlar, kaburgaların, kalbin ve klaviküllerin segmentasyonunu içerir [6]. Örneğin, Van Ginneken ve arkadaşları aktif şekiller, kural tabanlı yöntemler, piksel sınıflandırması ve bunların çeşitli kombinasyonlarını içeren akciğer segmentasyonu için çeşitli teknikleri karşılaştırmıştır [7, 8]. Sonuçlar, piksel seviyesi sınıflandırmasının test verileri üzerinde çok iyi performans sağladığını göstermiştir. Dawoud kamuya açık JSRT veri setinde yoğunluk bilgileri ile eğitilmiş olan şekilleri birleştiren yinelemeli bir segmentasyon yaklaşımı sunmuştur [9]. Yine Van Ginneken ve arkadaşları akciğer segmentasyonuna bağlı olarak farklı özellik tipleri ve bunları toplamanın yollarını bildirmişlerdir [6, 7].

Bazı CAD üzerine yazılan makalelerde CXR'lerin anormallikleri herhangi bir spesifik hastalığa odaklanmasından bahsedilmekte olup bir kısmında da CAD sisteminin TB tespitinde uzmanlaşmış olduğu bildirilmiştir [9-12]. Ayrıca, TB taraması sadece normal / anormal sınıflandırma ile sınırlı değildir, aynı zamanda anormalliğin bulunduğu CXR'lerdeki yeri saptamak için de önemlidir [7]. Örneğin, Van Ginneken ve ark. akciğerleri farklı boyutlarda örtüşen bölgelere bölerler ve her bölgeden özellikleri çıkarırlar [9]. Hogeweg ve ark. Yanlış pozitif tepkileri silmek için doku bazlı bir anormallik tespit sistemi ile klavikula tespit aşamasını birleştirdiler [10]. Üst akciğer bölgelerindeki bir boşluk TB'nin oldukça bulaşıcı bir duruma geldiğini gösteren güçlü bir göstergedir. Shen ve ark. bu bölgelerdeki boşlukları otomatik olarak saptamak için melez bilgi tabanlı bir Bayesian yaklaşımı geliştirdiler [11]. Xu ve arkadaşları Hessian matrisine dayanan görüntü geliştirme ile model tabanlı bir şablon eşleştirme tekniği ile aynı soruna yaklaştı [12]. Hogeweg ve arkadaşları, klavikula segmentasyonu için piksel sınıflandırıcıların ve aktif şekil modellerinin bir kombinasyonunu kullanmaktadır [13]. Klavikula bölgesi TB saptaması için oldukça zor bir bölge olmaktadır, çünkü klaviküller akciğerin apeksindeki TB belirtilerini gizleyebilir. Freedman ve ark. yakın zamanda yapılan bir çalışmada, CXR'lerde kaburgaların ve klaviküllerin otomatik olarak sildirilmesinin bir radyoloğun nodül tespiti için performansını önemli ölçüde artırabileceğini göstermiştir [14]. Arzhaeva ve ark. anormalliği bilinen ancak hastalığın kesin yeri bilinmeyen CXR'lerle başa çıkmak için farklılığa dayalı sınıflamayı kullanmaktadır [15]. Bilinen hastalık yerlerine sahip CXR'lerde bölge sınıflandırması ile elde edilen oranlarla karşılaştırılabilir sınıflandırma oranları bildirirler [16].

## 2. GENEL KISIMLAR

### 2.1. Röntgen

Röntgen X ışınlarının kullanımına dayanmaktadır. Işın sabit ve dönmeyen bir kaynaktan (bir tüp) yayılır. Bu ışınlar, hastanın arka tarafından yerleştirilmiş bir ışığa duyarlı film üzerinde toplanmadan önce yoğunluğuna göre dokular tarafından az ya da çok emilir. Fotoğrafta geçen dokuların yoğunluğuna göre opaklık ve netlik olarak görünür [17].

X ışınlarının insan vücudu tarafından bu şekilde emilmesi, 4 tip temel yoğunluk tanımaya mümkün kılar: kalsiyum yoğunluğu, su yoğunluğu, yağ yoğunluğu ve hava yoğunluğu. Hava ve yağlı yoğunluk siyah görüntüler verir, çünkü hava X-ışınlarını absorbe etmez.

Bununla birlikte, kalsiyum ve su yoğunlukları X ışınlarının emilim derecesine göre beyaz görüntüler verir, burada opaklıktan söz edilmektedir. Akciğer hava yoğunluğundan dolayı radyografik görüntüsü siyah iken kemikler (klaviküller, omuz bıçakları, kaburgalar ...) kalsiyum yoğunlukları nedeniyle beyaz bir radyografik görüntüye sahiptir. Bronşların pulmoner seviyeye yerleştirilmesi kalın yapılarını oldukları için opaklık (beyaz görüntü) verir [18].

### 2.2. Akciğer tüberküloz

Tüberküloz (TB), Bacillus Mycobacterium (Koch'un Bacillus'u, BK) tüberküloz tarafından kaynaklanan bulaşıcı bir hastalıktır. TB dünyanın en ölümcül bulaşıcı hastalıklarından biridir. 2013 yılında tahmini 9.000.000 kişi tüberküloz hastası olduğu bilinmektedir. Bunlardan 1.100.000 (% 13) HIV-pozitifdir. Rapora göre, 1.500.000 kişi tüberküloz hastalığından ölmüştür [19].

TB genellikle akciğeri etkiler, ancak kemikler ve diğer yumuşak dokular gibi diğer organları da etkileyebilir. Aktif TB hastaları öksürür, hapsirir veya başka türlü bulaşıcı bakterileri dışarı çıkardığında havaya yayılır. WHO'ya göre, TB yükündeki artışın en büyük etkisi gelişmekte olan ülkelerdeki yetersiz kontrol ile ilgilidir, özellikle endemik HIV enfeksiyonu oranları yüksek olanlarda HIV / AIDS hastalarında fırsatçı enfeksiyonların sorunu daha da kötüleşmesine sebep olmaktadır [2, 20].

TB, Sahra altı Afrika ve Güneydoğu Asya'da en yaygın olanıdır; HIV'e ek olarak yaygın yoksulluk ve yetersiz beslenme de hastalığa karşı direnci azaltır. Afrika Bölgesi, her beş HIV-pozitif TB vakasından dördünden ve HIV pozitif olan insanlar arasında TB ölümünden sorumludur [19].

Klinik olarak iki hasta grubunu ayırt edilir: aktif TB'li hastalar ve aktif TB olmayanlar (maruz kalanlar). Son gruptaki hastalarda vücutta özellikle akciğerde BK olabilir. Eğer bağışıklık sistemi tehlikeye girerse tüberkülozun aktivasyon riskini taşır. HIV pozitif kişilerde durum budur.

Aktif tüberkülozda, aşağıdaki radyografik belirtileri ayırt ederiz [21]:

TB Miliyer: Her iki akciğerde dağılan ince tanelere benzer. Yaygın bilateral infiltrat olarak tanımlanır, ayrıca mikronodül olarak adlandırılır [22];

Hava sahası konsolidasyonu: Genellikle pnömoni olarak tanımlanan lobar opasitesi;

Kavite oluşumu veya kavitasyon: sürekli veya süreksiz radyopak kenarları olan oyuklar. Göreceli veya tam merkezi netliği nedeniyle bir küleden farklıdır;

Bronşektazi veya hava yollarının genişlemesi tübüler halkalar veya düzensiz çaplı silindirler şeklinde olabilir.

Adenopati: Lenf düğümlerinin büyümesi. Mediasten yakınında iyi tanımlanmış dağılımı olan kütle;

Plevranın kalınlaşması: Plevra kalınlığını artırır;

Plevral efüzyon: Postero-anterior radyografide lateral ve medial kostofreniklerin görselleştirme kaybı (belirsiz).

Bunlara ek olarak, TB maruziyetinin klasik özellikleri kalsifiye granülomları ve apikal plevranın kalınlaşmasını içerir. Bu özelliklerin çoğu tüberkülozda ortaya çıkar, aynı zamanda pnömoni gibi diğer birçok akciğer hastalığında da ortaya çıkar.

Göğüs röntgeni yukarıda açıklandığı akciğer lezyonlarını vurgulayan önemli bir muayenedir ve böylece tüberküloz tanısına izin verir [23]. Ancak yorumlanması bir radyolog gerektirir ve radyologlar dünyanın pek çok yerinde kolay bulunmamaktadır özellikle Sahra altı Afrika, Orta Doğu gibi ülkelerde.

### **2.3. Makine Öğrenmesi**

Makine öğrenmesi (ML) son zamanlarda önemli ölçüde gelişmeler kaydetmiştir.

ML'nin tarihi: 1943 yılında Warren McCulloch ve Walter Pitts ilk sinir ağı modelini tanıttıklarında başlamıştır. 1955'te Oliver Selfridge örüntü tanıma için bilgisayar kullanılmasını önerdi.

1956'da John McCarthy, Marvin Minsky, IBM'den Nathan Rochester, ve Claude Shannon ilk yapay zeka konferansını Amerika Birleşik Devletleri Dartmouth College'de düzenledi. İkinci Dartmouth konferansında, yapay zeka terimi ilk kez kullanıldı. Bilişsel bilim terimi 1956'da Amerika Birleşik Devletleri MIT'de bilgi biliminde bir sempozyumda ortaya çıktı.

Paul Werbos, geri yayılım ile ilk verimli sinir ağı modelini önerdi. Fakat 1986'da David Rumelhart, Geoffrey Hinton ve Ronald Williams, bir ağın doğrusal olmayan ayrılabilir sınıflar arasında ayırım yapmayı öğrenmelerine izin veren bir yöntem keşfetti ve onu geri yayılım olarak adlandırdılar [24, 25].

1997'den günümüze kadar olan süreç, pekiştirici öğrenme, doğal dil işleme, duygusal anlayış, bilgisayarlı görü ve bilgisayar duyma konularında hızlı gelişmelere tanık olmuştur. Makine öğrenmesinde günümüzdeki araştırmaları bilgisayar görme, işitme, doğal dil işleme, görüntü işleme ve örüntü tanıma, bilişsel hesaplama, bilgi sunumu üzerine odaklanmaktadır.

Dört genel makine öğrenme yöntemi vardır: denetimli, denetimsiz, yarı denetimli ve pekiştirici öğrenme yöntemleri [24-26]. Burada denetimli ve denetimsiz yöntemleri tarif edeceğiz.

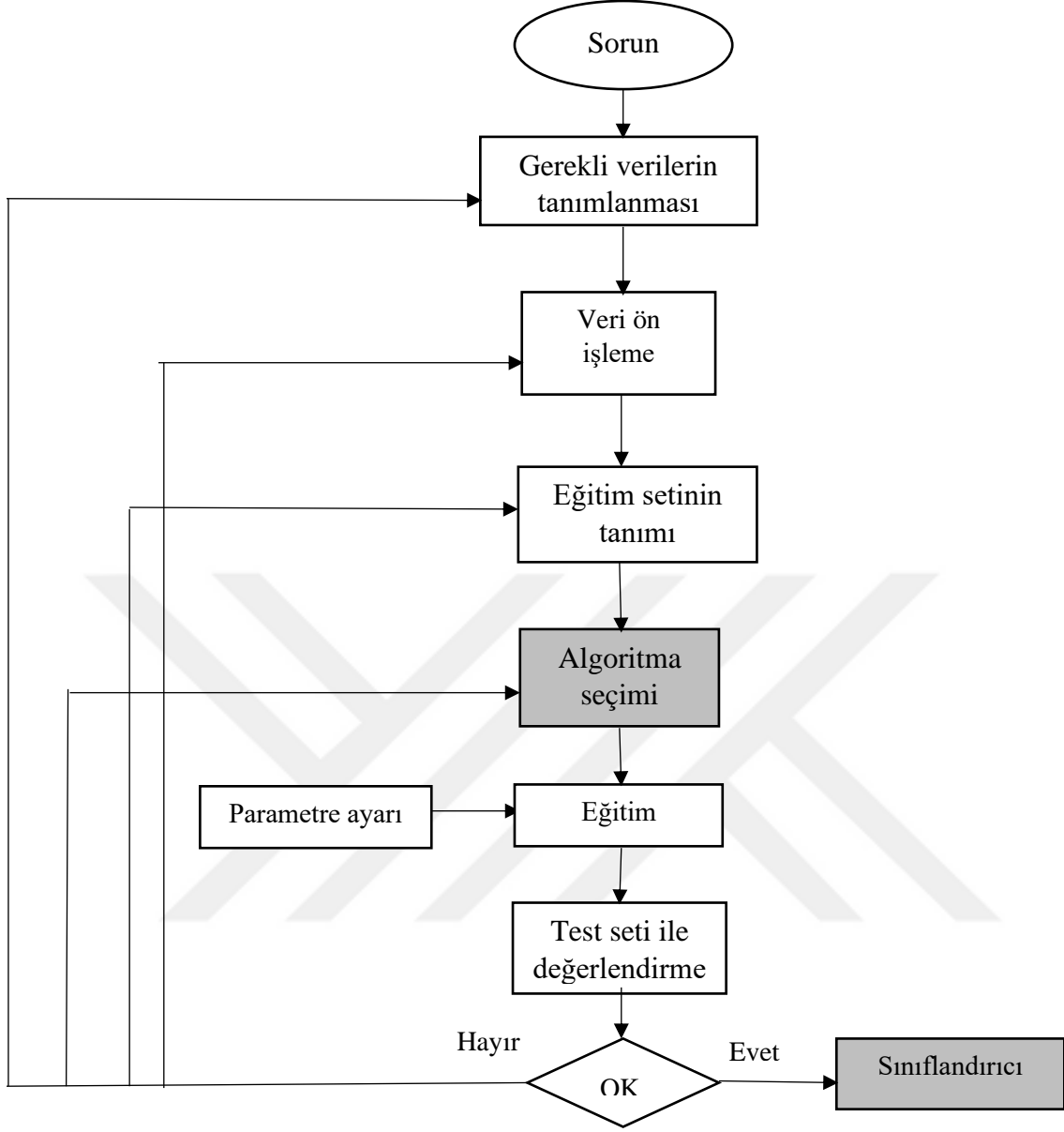
**Denetimli ML**, genel olarak hipotezler üretmek için dışarıdan verilen örneklerden kaynaklanan algoritmalar araştırmasıdır. O zaman gelecekteki örnekler hakkında tahminler yapılabilir [26].

Denetimli öğrenme görevinin standart bir formülasyonu sınıflandırma problemidir: Öğreticinin fonksiyonun birkaç girdi-çıkış örneğine bakarak bir vektörü birkaç sınıftan birine eşleştiren bir fonksiyonu öğrenmesi gerekir (to approximate the behavior of).

Endüktif makine öğrenmesi örneklerden bir seti kural (eğitim setindeki örnekler) öğrenme sürecidir veya daha genel olarak konuşursak, yeni örneklerden genellemek için kullanılacak bir sınıflandırıcı oluşturmaktır [27]. Denetimli ML'yi gerçek dünyadaki bir soruna uygulama süreci, Şekil 2.1'de açıklanmaktadır.

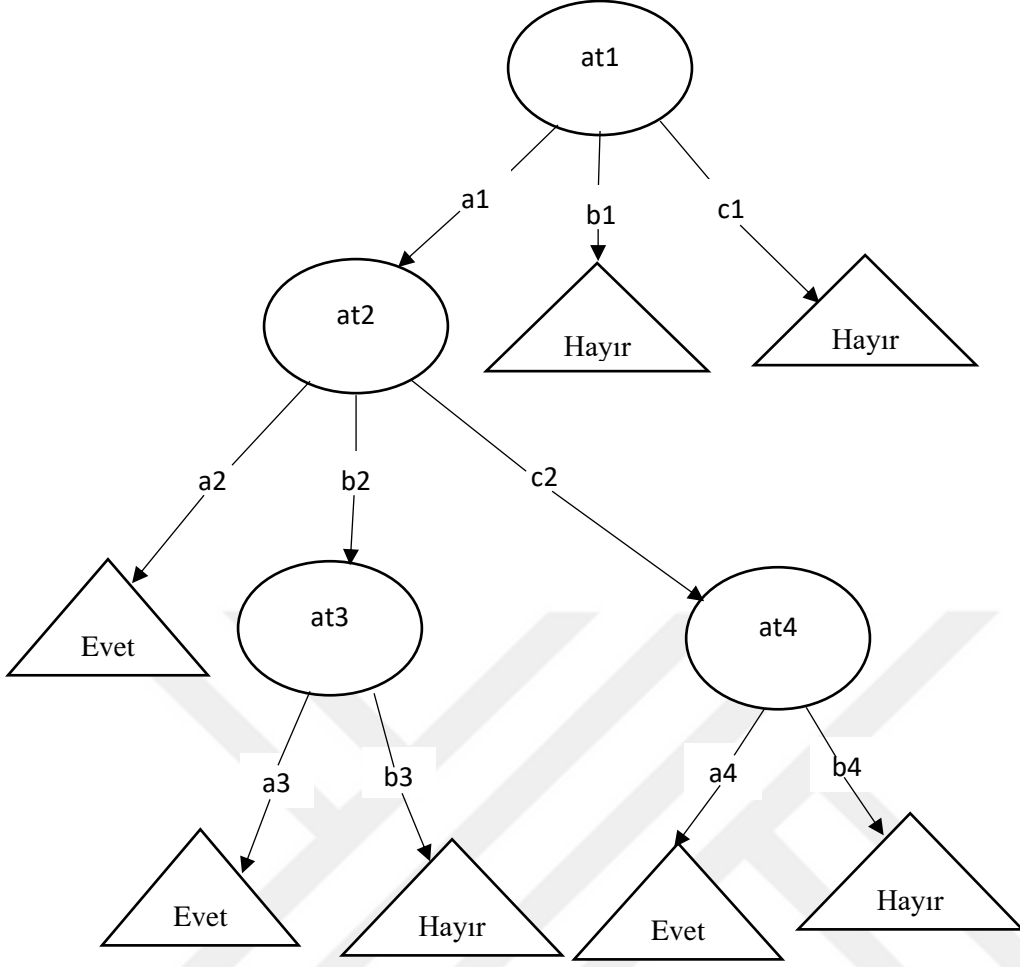
Denetimli sınıflandırma akıllı sistemler tarafından en sık gerçekleştirilen görevlerden biridir. Böylece Yapay Zekaya dayanan çok sayıda teknik geliştirilmiştir: Mantık tabanlı teknikler (Karar ağaçları), Perceptron tabanlı teknikler (Yapay sinir ağları), İstatistiksel öğrenme teknikleri (Bayesian Networks), Örnek tabanlı teknikler (k-Nearest Neighbor) ve destek vektör makineleri bunlara örnek olarak verilebilir. Sonraki bölümlerde denetimli bazı makine öğrenme tekniklerine daha fazla odaklanacağız.

**Karar ağaçları**, örnek alanının özyinelemeli bir bölümü olarak ifade edilen mantıksal tabanlı bir sınıflandırıcıdır. Karar ağacı köklü bir ağaç oluşturan düğümlerden oluşur. Yani gelen kenarları olmayan “kök” adı verilen bir düğümü olan yönlendirilmiş bir ağaçtır. Diğer tüm düğümlerin tam olarak bir gelen kenarı vardır. Diğer tüm düğümlerin tam olarak bir gelen kenarı vardır. Giden kenarları olan bir düğüme iç veya test düğümü denir. Diğer tüm düğümlere yaprak adı verilir (ayrıca terminal veya karar düğümleri olarak da bilinir). Bir karar ağacında, giriş öznitelik değerlerinin belirli bir ayrık fonksiyonuna göre her bir iç düğüm örnek alanını iki veya daha fazla alt alana böler [28]. En basit ve en sık görülen durumda, her test tek bir özellik dikkate alır. Örnek alanı özniteliğin değerine göre bölümlenecek şekilde. En basit ve en sık görülen durumda, her test tek bir öznitelik dikkate alır örnek alanı, özniteliğin değerine göre bölümlenecek şekilde olur. Sayısal niteliklerde, koşul bir aralığa işaret eder. Her yaprak en uygun hedef değeri temsil eden bir sınıfa atanır. Alternatif olarak, yaprak belirli bir değere sahip olan hedef özniteliğinin olasılığını gösteren bir olasılık vektörü tutabilir [27].



**Şekil 2.1:** Denetimli ML akış diyagramı.

Örnekler, yol boyunca yapılan testlerin sonucuna göre, ağacın kökünden bir yaprağa doğru ilerletilerek sınıflandırılır [27]. Şekil 2.2 ve Tablo 2.1’de eğitim için örnek veri setini ve bir karar ağacını göstermektedir. İç düğümler daireler olarak gösterilirken, yapraklar üçgen olarak belirtilir. Her bir düğüm, test ettiği öznelik ile etiketlenir ve dalları ilgili değerlerle etiketlenir.



Şekil 2.2: Eğitim seti için bir karar ağacı örneği.

Tablo 2.1: Örnek veri seti.

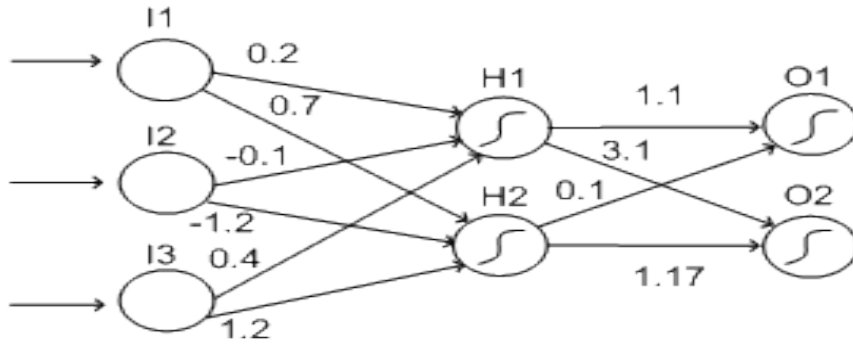
at1	at2	at3	at4	Class
a1	a2	a3	a4	Evet
a1	a2	a3	b4	Evet
a1	b2	a3	a4	Evet
a1	b2	b3	b4	Hayır
a1	c2	a3	a4	Evet
a1	c2	a3	b4	Hayır
b1	b2	b3	b4	Hayır
c1	b2	b3	b4	Hayır

Karar ağaçları kökten ağaçtaki bir yaprağa her yol için ayrı bir kural oluşturarak bir dizi kuralla çevrilebilir [29]. Bununla birlikte, kurallar ayrıca çeşitli kural tabanlı algoritmalar kullanılarak eğitim verilerinden doğrudan elde edilebilir.

**Kural tabanlı sınıflandırıcılar** eğitim verileriyle tutarlı olan en küçük kural kümesini oluşturmayı hedefler. Çok sayıda öğrenilen kural genellikle öğrenme algoritmasının onu yöneten varsayımları keşfetmek yerine, eğitim setini hatırlamaya çalıştığının bir işaretidir. Eğitim örneklerinin bir bölümünü açıklayan bir kural için ayrı ve fethetme algoritması (kapsayan algoritmalar) arayışı, bu örnekleri ayırır ve hiçbir örnek kalmayana kadar daha fazla kural öğrenerek kalan örnekleri tekrar tekrar fetheder. Kural öğrenmesi için sezgisel tarama ile karar ağaçları için sezgisel tarama arasındaki fark, ikincisinin bir takım ayrık setlerin (test edilen özelliğin her değeri için bir tane) ortalama kalitesini değerlendirirken, kural öğrenenleri yalnızca aday kural tarafından kapsanan örneklerin kalitesini değerlendirir. Daha gelişmiş kural öğrenenler, bu basit sözde koddan, çoğunlukla eğitim verilerinin aşırı sığmasını önlemek için ek mekanizmalar ekleyerek farklılık gösterir, Mesela, uzmanlık sürecini bir kalite önlemi kullanarak durdurarak veya aşırı özel kuralları ayrı bir budama aşamasında genelleştirerek [30].

Algılayıcılar yalnızca doğrusal olarak ayrılabilen örnek setlerini sınıflandırabilir. Giriş örneklerini doğru kategorilere ayırmak için düz bir çizgi veya düzlem çizilebilirse, giriş örnekleri doğrusal olarak ayrılabilir ve algılayıcı çözüm bulur. Örnekler doğrusal olarak ayrılmazsa, öğrenme hiçbir zaman tüm örneklerin doğru bir şekilde sınıflandırıldığı bir noktaya ulaşmaz. Bu sorunu çözmeye çalışmak için çok katmanlı algılayıcılar (Yapay Sinir Ağları) yaratılmıştır [30]. Çok katmanlı bir sinir ağı, bir bağlantı düzeninde bir araya getirilmiş çok sayıda birimden (nöron) oluşur (Şekil 2.3). Bir ağdaki birimler genellikle üç sınıfa ayrılır: giriş birimleri, işlenecek bilgiyi alan; çıkış birimleri ise işlemin sonuçlarının bulunduğu ve aralarındaki gizli birimler olarak bilinen birimler. İleri beslemeli YSA'lar, sinyallerin girişten çıkışa yalnızca bir yönde gitmesine izin verir.

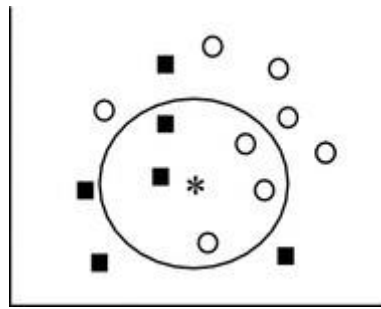
İlk olarak, ağ giriş-çıkış eşlemesini belirlemek için bir çift eşleştirilmiş veri seti üzerinde eğitilir. Nöronlar arasındaki bağlantıların ağırlıkları daha sonra sabitlenir ve ağ yeni bir veri setinin sınıflandırılmasını belirlemek için kullanılır. Sınıflandırma sırasında giriş birimlerindeki sinyal, tüm çıkış birimlerindeki aktivasyon değerlerini belirlemek için ağın her yerine yayılır. Her giriş ünitesinde, ağın dışındaki bazı özellikleri temsil eden bir aktivasyon değeri vardır [31]. Ardından her giriş birimi aktivasyon değerini bağlı olduğu gizli birimlerin her birine gönderir. Her alıcı birimin aktivasyon değeri basit bir aktivasyon fonksiyonuna göre hesaplanır.



Şekil 2.3: İleri beslemeli YSA örneği.

İstatistiksel yöntemler başlığı altındaki diğer bir kategori örnek tabanlı öğrenmedir. Örnek tabanlı öğrenme algoritmaları, sınıflandırma yapılandırma kadar induksiyon veya genelleme sürecini geciktirdikleri için tembel öğrenme algoritmalarıdır [32]. Tembel öğrenme algoritmaları, eğitim aşamasında istekli öğrenme algoritmalarına göre (karar ağaçları, sinir ve Bayes ağları gibi) daha az hesaplama süresi gerektirir ancak sınıflandırma sürecinde daha fazla hesaplama süresi gerektirir. En basit örnek tabanlı öğrenme algoritmalarından biri en yakın komşuluk (k-NN ya da KNN) algoritmasıdır.

**En yakın komşuluk (kNN)** bir dataset içindeki örneklerin genellikle benzer özelliklere sahip diğer örneklere yakın olarak var olacağı prensibine dayanmaktadır [33]. Örnekler bir sınıflandırma etiketi ile etiketlenirse sınıflandırılmamış bir örneğin etiketinin değeri en yakın komşusunun sınıfı gözlenerek belirlenebilir. kNN sorgu örneğine k en yakın örneklerini bulur ve en sık tek sınıf etiketini tanımlayarak sınıfını belirler (Şekil 2.4).



Şekil 2.4: k-en yakın komşu sınıflandırması.

Bir özellik uzayında A ve B noktaları arasındaki mesafeyi ölçmek için, öklidyen gibi mesafe fonksiyonunun en yaygın kullanılanlarından çeşitli mesafe fonksiyonları kullanılır [34]. A ve B;

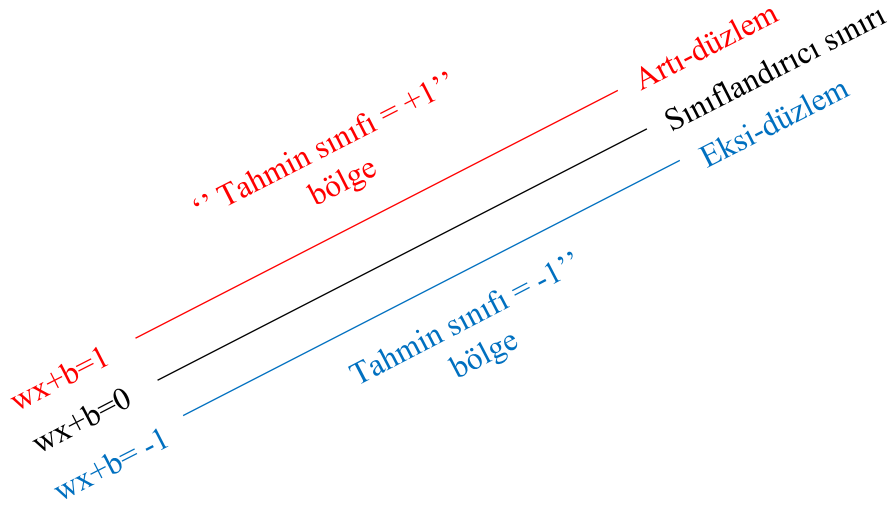
$A = (x_1, x_2, \dots, x_m)$  ve  $B = (y_1, y_2, \dots, y_m)$  özellik vektörleriyle temsil edilir,  $m$  özellik uzayının boyutudur. A ve B arasındaki mesafeyi hesaplamak için normalize edilmiş öklidyen metriği genellikle kullanılır. Tablo 2.2’de çeşitli mesafe hesaplama formülleri gösterilmiştir.

**Tablo 2.2:** Örnekler (x ve y) arasındaki mesafeyi belirleyen yaklaşımlar [27].

Minkowsky: $D(x,y) = \left( \sum_{i=1}^m  x_i - y_i ^r \right)^{1/r}$
Manhattan: $D(x,y) = \sum_{i=1}^m  x_i - y_i $
Chebychev: $D(x,y) = \max_{i=1}^m  x_i - y_i $
Euclidean: $D(x,y) = \left( \sum_{i=1}^m  x_i - y_i ^2 \right)^{1/2}$
Camberra: $D(x,y) = \sum_{i=1}^m \frac{ x_i - y_i }{ x_i + y_i }$
Kendall's Rank Correlation: $D(x,y) = 1 - \frac{2}{m(m-1)} \sum_{i=j}^m \sum_{j=1}^{i-1} \text{sign}(x_i - x_j) \text{sign}(y_i - y_j)$

**Destek vektör makineleri** (SVM) iki veri sınıfını ayıran bir hiper düzlemin her iki tarafındaki marj nosyonunun etrafında döner. Doğrusal bir sınıflandırıcının sınırı bir veri noktasına çarpılmadan önce sınırın arttırılabileceğinden tanımlanır [35]. Marjı maksimize etmek ve böylelikle ayrılan hiper düzlem ile her iki taraftaki örnekler arasında mümkün olan en büyük mesafeyi oluşturmaktadır. Beklenen genelleme hatasının üst sınırını azalttığı kanıtlanmıştır. Şekil 2.5’te sınıflandırıcı çizgisi gösterilmiştir.

Eđitim verileri dođrusal olarak ayrılabilirse, bir çift (w, b) vardır.



Şekil 2.5: İki sınıflı bir veri setinin bölgelerini ayıran sınırlar.

$$w^T x_i + b \geq 1, \text{ hepsi } x_i \in P$$

$$w^T x_i + b \leq -1, \text{ hepsi } x_i \in N$$

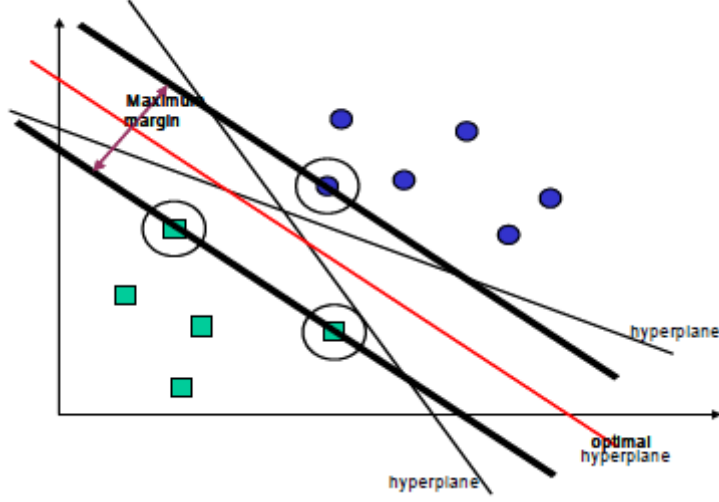
$f_{w,b}(x) = \text{sgn}(w^T x + b)$  Tarafından verilen karar kuralında, w, vektörün ağırlığı olarak adlandırılır ve b önyargılıdır (veya -b eşik olarak adlandırılır).

İki sınıfı dođrusal olarak ayırmanın mümkün olduđu durumlarda bunu göstermek kolaydır, ayırıcı hiper düzlemin kare normunu asgariye indirerek optimum bir ayırıcı hiper düzlem bulunabilir. Minimizasyon, dışbükey karesel ikilenik bir programlama problemi olarak kurulabilir:

$$\text{Küçültmek } \Phi(w) = \frac{1}{2} \|w\|^2$$

$$\text{Bađlı ile } y_i (w^T x_i + b) \geq 1, i=1, \dots, l.$$

Dođrusal olarak ayrılabilir veriler olması durumunda, optimum ayırıştırma hiper düzlemi bulunduđunda, marjında yatan veri noktaları destek vektör noktaları olarak bilinir ve çözüm sadece bu noktaların dođrusal bir kombinasyonu (Şekil 2.6) olarak temsil edilir. Diđer veri noktaları dikkate alınmaz.



Şekil 2.6: İki sınıflı bir veri setini ayıran düzlemler.

Bu nedenle, bir SVM'nin model karmaşıklığı, eğitim verilerinde karşılaşılan özelliklerin sayısından etkilenmez (SVM öğrenme algoritması tarafından seçilen destek vektörlerinin sayısı genellikle azdır). Bu nedenle SVM'ler eğitim örneklerinin sayısına göre özelliklerin büyük olduğu öğrenme görevleriyle başa çıkmak için çok uygundur.

Maksimum marj SVM'nin birden fazla aday hiper düzlem arasından seçim yapmasına izin vermesine rağmen, birçok dataset için veriler yanlış sınıflandırılmış örnekler içerdiğinden SVM hiçbir ayırıcı hiper düzlemi bulamayabilir. Sorun, eğitim durumlarının bazı yanlış sınıflandırmalarını kabul eden yumuşak bir marj kullanarak ele alınabilir [36]. Bu kısıtlamalara  $\xi_i, i = 1, \dots, N$  pozitif durgunluk değişkenleri getirilir;

$$w \cdot x_i - b \geq +1 - \xi \quad \text{for } y_i = +1$$

$$w \cdot x_i - b \geq -1 - \xi \quad \text{for } y_i = -1$$

$$\xi \geq 0,$$

Bu nedenle, bir hatanın oluşması için karşılık gelen  $\xi_i$  birliği geçmelidir, bu yüzden  $\sum_i \xi_i$  eğitim hatalarının sayısına bağlı bir üst sınırdır. Bu durumda Lagrangian:

$$L_p \equiv \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha \{y_i(x_i \cdot w - b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$$

$\mu_i, \xi_i$ 'nin pozitifliğini artırmak için Lagrange çarpanlarıdır.

Bununla birlikte, çoğu gerçek dünyadaki sorunların olumlu olanı eğitim setindeki negatif durumlardan başarıyla ayıran hiçbir hiperdüzlemin bulunmadığı ayrılamaz verileri içerir. Ayrılmazlık sorununa bir çözüm verileri daha yüksek boyutlu bir alana haritaya koymak ve orada ayırıcı bir hiper düzlem tanımlamaktır. Bu duruma eğitim örnekleri tarafından işgal edilen girdi alanına karşılık olarak, yüksek boyutlu alana dönüştürülmüş özellik alanı denir.

Uygun bir şekilde seçilen dönüştürülmüş özellik alanı, yeterli boyutluluğa sahip olduğunda, tutarlı bir eğitim seti ayrılabilir hale getirilebilir. Dönüştürülmüş özellik alanındaki doğrusal bir ayırım, orijinal giriş alanındaki doğrusal olmayan bir ayırma karşılık gelir.

Verileri başka bir Hilbert alanı  $H$ 'a haritaya koyma  $\Phi$  olarak:  $R^d \rightarrow H$ . O zaman eğitim algoritması sadece  $H$ 'deki skaler çarpımlardan elde edilen verilere bağlı olacaktır, Örneğin,  $\Phi(x_i) \cdot \Phi(x_j)$  formunun fonksiyonları hakkında. Eğer  $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$  gibi bir Kernel (çekirdek) fonksiyonu  $K$  olsaydı,  $K$ 'yı sadece eğitim algoritmasında kullanmamız gerekirdi ve asla  $\Phi$  açıkça belirlenmesi gerekmez. Bu nedenle, Kernels yukarıda açıklanan haritalamayı gerçekleştirmeden iç çarpımların doğrudan özellik alanında hesaplanmasını sağlayan özel bir fonksiyon sınıfıdır [37].

Bir hiper düzlem oluşturulduktan sonra Kernel fonksiyonu sınıflandırma için özellik alanına yeni noktaları haritaya koymak için kullanılır. Bazı popüler Kerneller şunlardır:

$$(1) K(x, y) = (x \cdot y + 1)^P,$$

$$(2) K(x, y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}$$

$$(3) K(x, y) = \tanh(kx \cdot y - \delta)^P$$

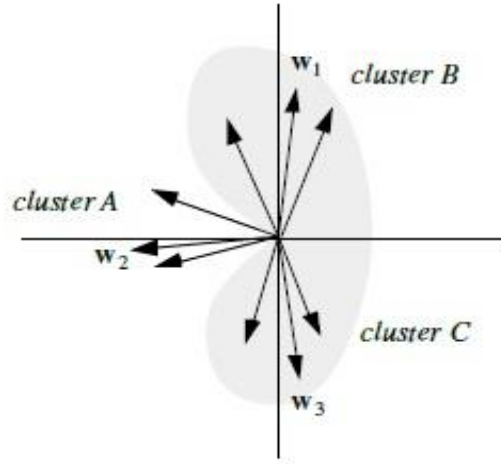
**Topluluk altuzay:** Topluluk öğrenme, tahmin performansını iyileştirmek için çeşitli modelleri birleştirir, örneğin (i) rastgele alt uzay, eğitim algoritmasını gerçekleştirmeden önce rastgele bir özellik alt kümesini seçerek (seçilen alt uzay) öğrenme algoritmasını rastgele hale getirir ve daha sonra model çıktıları çoğunluk oyu ile birleştirilir, (ii) torbalama (Bootstrap Aggregation). Bu rastgele bir veri üzerinde eğitilen bir set model oluşturur, daha sonra tahminler ortalama kullanılarak final tahmin için toplanır / birleştirilir. Ve (iii) güçlendirme birden fazla modelin ortalamasının alınmasına / oylanmasına dayanır, inşa edilen modellerin performanslarına göre ağırlıklandırılır. Alt uzay ayırt edici topluluğu alt uzay öğrenme tekniklerinin önemli bir rolü

vardır; Özellikle doğrusal diskriminant analizi (LDA) yöntemi gibi. Burada düşük boyutlu belirleyici bir ayırt edici alt uzay belirlenmeye çalışılır [38].

**Denetimsiz öğrenme** genel olarak bir gözlem toplama topluluğu kullanma fikri ile ilişkilidir  $X_1, \dots, X_n$   $p(X)$  'un özelliklerini tanımlamak için bir  $p(X)$  dağılımından örneklenmiştir. Bu tanım son derece geneldir ve örneğin tanımlayıcı istatistiklerin herhangi bir prosedürünü tanımlayabilir [39]. Kümeleme, karışım ayrışması, temel bileşen analizi, birleşme kuralı keşfi, çok boyutlu ölçeklendirme gibi isimlerle adlandırılır, burada kümeleme yöntemine odaklanacağız.

Kümeleme tek algılayıcı gibi diğer bazı yöntemlerden daha karmaşık bir sınıflandırmayı çözebilir.

Tek algılayıcıda, P'deki bütün  $p$  vektörler için  $w \cdot p \geq 0$  ve N'deki bütün  $n$  vektörler için;  $w \cdot n < 0$  yerine getiremeyen bir ağırlık vektörü vardır. P ve N iki boyutlu bir problemde iki vektör seti olduğunda. Bu durumda, Şekil 2.7'de gösterilen üç kümenin (A, B ve C) her birinde vektörler için bir tür temsilci olarak işlev görebilecek üç farklı vektör ( $w_1$ ,  $w_2$  ve  $w_3$ ) bulmak mümkündür. Bu vektörlerin her biri, kümesindeki her vektörden çok uzak değildir. Her ağırlık vektörü, yalnızca girdi vektörü kendi ağırlık vektörüne yeterince yakın olduğunda ateşleyen tek bir hesaplama ünitesine karşılık gelir [40].



**Şekil 2.7:** Üç küme için üç ağırlık vektörü.

## 2.4. Bölütleme

Bölütleme algoritmaları, orijinal görüntülere veya dönüşümler ve filtrelerin uygulanmasından sonra uygulanabilir.

Bölütlendirmeye genel yaklaşım:

- Eşikleme,
- Kenar tabanlı yöntemler,
- Genişletme,
- Bölge tabanlı yöntemler,
- Kümeleme yöntemleri,
- Maske bazlı segmentasyon,
- Dalgacık dönüşümü.

Sınıflandırma işleminde eşik değerini değiştirmek için, komşu piksellerin değerleri olan bir pikselin içeriği de kullanılabilir. En çok kullanılan yöntemlerden ikisi:

- Histogramı, benzer şekilde komşu olan piksellerle sınırlandırmak,
- Sınıflandırma sonrası yumuşatma.

Kenar tabanlı bölümlendirmede, görüntüye bir kenar filtresi uygulanır ve pikseller kenar ve kenar olmayan olarak sınıflandırılır. Bunlar genellikle birinci veya ikinci dereceden türev yöntemi ile tespit edilir [41]. Kenar etiketleme şunlar olabilir:

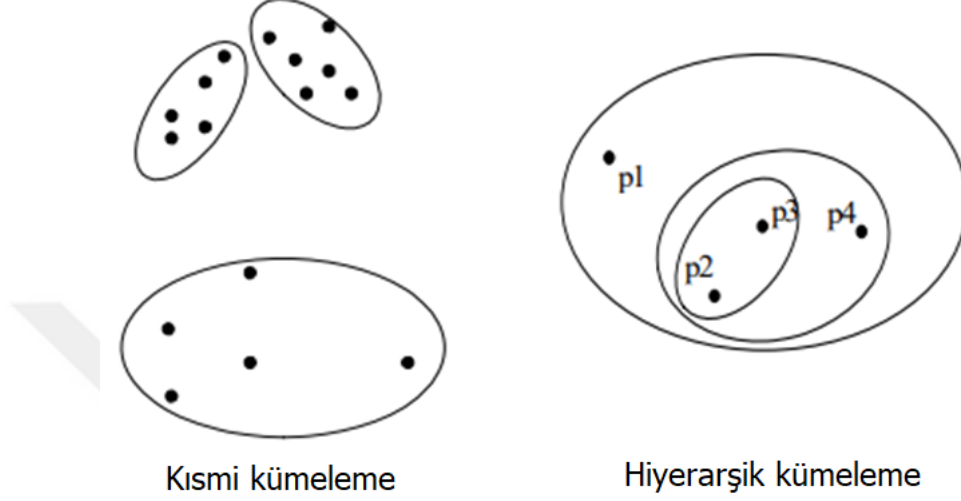
- Elle yapılan, bir ekran imlecini kontrol etmek ve bölgeler arasında sınır çizgileri çizmek için bir bilgisayar faresi kullanarak,
- Otomatik: kenar algılama filtresi kullanarak. Kenarlar bulunabilir:
  - Prewitt gibi bir filtreden gelen çıktılarının bir eşiği aştığı durumlarda,
  - Veya bir Laplasian-Gaussian filtresinden sıfır geçişlerde.

Genişletme, nesnelerin boyutunu artırır. A'nın B tarafından genişlemesi şu şekilde tanımlanır [42]:

$$A \oplus B = \bigcup_{b \in B} A_b$$



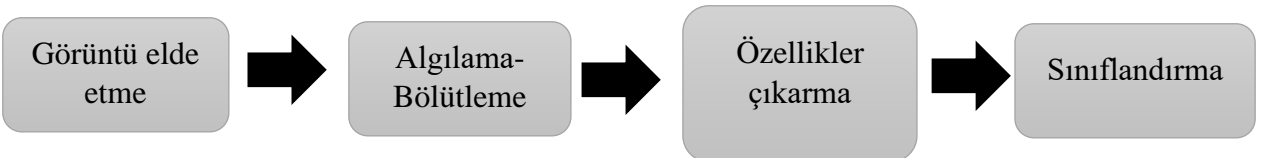
Örüntü tanıma kümelenmesi, bir örüntü vektörünü setler adı verilen alt kümelere bölme işlemidir [44]. Kümeleme Türleri: Bölümlenme ve Hiyerarşik Kümeleme. Hiyerarşik kümeleme hiyerarşik bir ağaç olarak düzenlenmiş iç içe setler setidir ve bölümlenme her veri nesnesinin tam olarak bir alt sette olduğu şekilde üst üste gelmeyen alt setlere (setler) bölünen veri nesnelere (Şekil 2.8).



Şekil 2.8: Kısımî ve Hiyerarşik kümelere gösterimi.

### 3. MATERYAL VE YÖNTEM

Uygulamamız, Şekil 3.1'de gösterilen prosedürün ana adımlarını takip ederek pulmoner tüberkülozun otomatik teşhisine odaklanmaktadır. Ancak kendimizi sadece tanı ile sınırlamıyoruz, akciğerleri normal mi yoksa anormal mi olduklarına göre de sınıflandıracamız. Kullanılan prosedür şu şekilde özetlenmiştir: görüntü elde etme, algılama-bölütlenme, özellikler çıkarma ve sınıflandırma. Algoritmik gelişim eğitim ve test için geniş görüntü veri setleri gerektirmektedir ve Matlab 2018b'de geliştirilmiştir.



Şekil 3.1: Görüntü analizine ait blok diyagram.

### 3.1. Veri Setleri

Tahmin ve sınıflandırma için Montgomery veri seti kullanılmıştır.

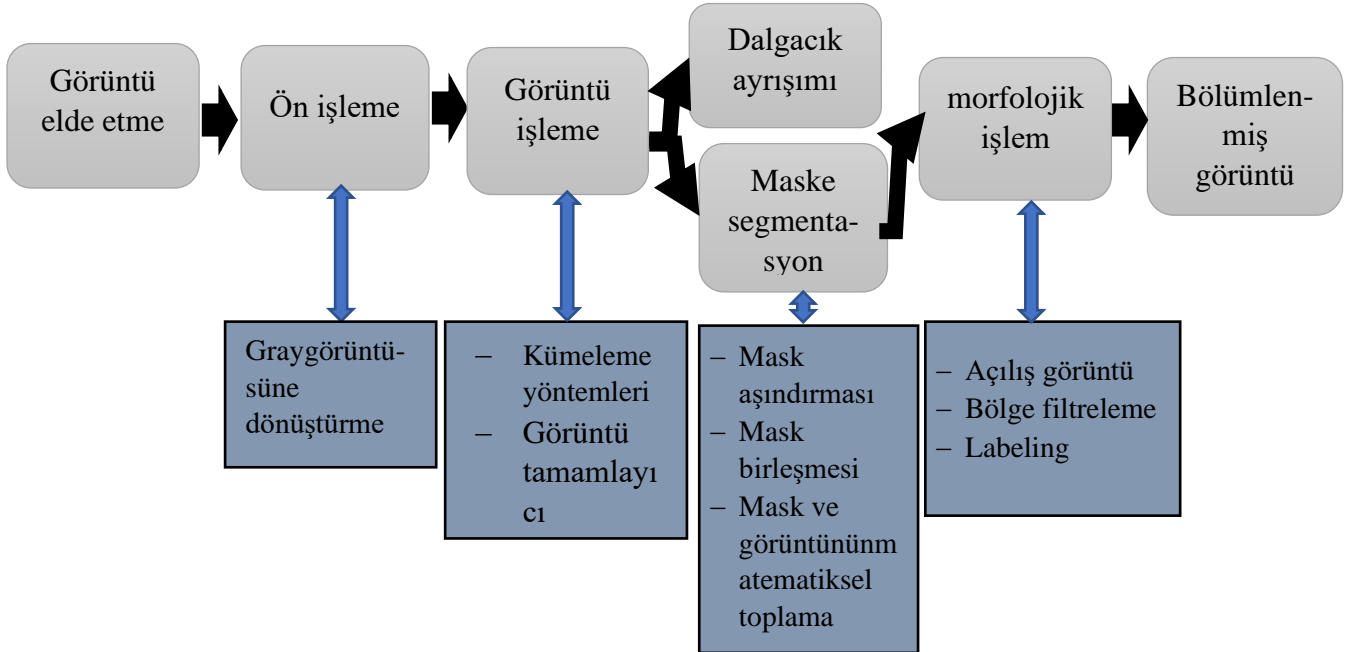
*Montgomery veri seti:* ABD, Maryland, Montgomery ilçesinde Sağlık ve İnsan Hizmetleri Departmanından elde edilmiştir. Bu set Montgomery ilçesi`ndeki Tüberküloz kontrol programı kapsamında toplanan CXR'leri içermektedir. Veri setinde 138 görüntüden 80'i normal ve 58'i tüberküloz belirtileriyle anormal olarak etiketlenmiştir. CXR görüntüleri 12 bit gri tonlama bir Eureka sabit X-ray makinesiyle elde edilmiştir. Veri setindeki görüntüler 4020 piksel x 4892 piksel çözünürlüğe sahiptir. Dikey ve yatay yönlerde piksel aralığı 0.0875 mm'dir.

### 3.2. Bölütleme

Dijital bir CXR görüntüsünde üst üste görünen birçok anatomik detay vardır. Bu klavikula, kaburga, kalp ve akciğer alanlarının durumudur. Akciğer alanları kendileri homojen değildir, ancak CXR'de birleştirilebilir pikseller öneren bir yapıya sahiptir.

Bilgisayar destekli bir CXR tarama sistemindeki ilk adımın anahtarı akciğer alanlarının otomatik algılanması ve bölütlenmesidir [45].

CXR görüntüler için yeni bir otomatik akciğer lezyonu algılama ve bölütleme algoritması geliştirdik (Şekil 3.2). Bölütlememiz birkaç yöntemi birleştirdi: Görüntü elde etme, Ön işleme, Görüntü işleme, Mask segmentation, Sonraki işleme ve Bölümlenmiş görüntü.



Şekil 3.2: Bölütlendirme adımları.

Dalgacık yöntemi ile özelliklerin çıkarımı adımı sırasında kullanım için resmimizi RGB'den gri görüntüye dönüştürerek ön işlem yapıldı.

Birkaç renk işlemlerin uygulandığı kümeleme yöntemini kullandık. Böylece tüm görüntüleri RGB'den YCbCr ve HSV renk alanına dönüştürdük ve bu bileşenlerin bazılarını izole ettik.

YCbCr renk uzayı renk yoğunluğu (parlaklık) ve iki renk farkı sinyali olarak temsil eder. RGB rengi üç farklı bileşenden (kırmızı, yeşil ve mavi) oluşmuştur. YCbCr renk alanı insan gözünün özelliklerini kullanır. YCbCr renk alanı görüntü işlemede yaygın olarak kullanılır çünkü Y bileşenindeki parlaklık, Cb ve Cr bileşenleriyle açıklanan renklilikten ayrılır. Parlaklık bileşeni yüksek çözünürlükte depolama için ayrı olarak kullanılabilir ve diğer iki krominans bileşeni performansı artırmak için değerlendirilebilir [45].

Böylece RGB CXR görüntüsünü YCbCr renk uzayına dönüştürdük ve iki farklı matristeki Y'i (yoğunluk) ve Cr'i (ikinci renk farkı sinyali) izole ettik. Yoğunluğu temsil eden  $Y = \text{Kırmızı} + \text{Yeşil} + \text{Mavi}$ , birinci renk farkı sinyalini temsil eden  $Cb = Y - \text{mavi}$  ve ikinci renk farkı sinyalini ise temsil eden  $Cr = Y - \text{kırmızı}$ dır [46].

RGB renk alanı ve YCbCr renk alanı arasındaki dönüşüm, aşağıdaki denklemlerle açıklanmaktadır:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Aynı CXR görüntüsünü de RGB'den HSV renk alanına dönüştürdük ve V (değer) bileşenini bir matriste izole ettik. H: Ton, S: Doygunluk, V: Değer. HSV renk alanı renkler arasındaki insan farklılaşmasına uyan Ton, Doygunluk ve Değer parametrelerine dayanmaktadır [47, 48]. Bu alanın bir avantajı kromatik (Ton ve Doygunluk) ve akromatik (Değer) bilgilerin ayrılmasıdır. Renk bilgisini Değer bilgisinden bağımsız olarak işleme imkânı sunar.

Sonra yeni bir matrisi ‘‘features (:,3)’’ (öznitelik) oluşturduk. Birinci sütunun Y (YCbCr renk alanının yoğunluğu) olduğu, ikincisinin Cr (YCbCr renk alanının ikinci renk farkı sinyali) ve üçüncü sütunun V (HSV renk alanının değeri veya parlaklığı) olduğu yer.

K-means clustering (K-Ortalama kümeleme) bir bölümlenme yöntemidir. K-means clustering aktüel gözlemler üzerinde çalışır ve tek bir küme düzeyi oluşturur. K-means clustering genellikle büyük miktarlarda veri için hiyerarşik kümelenden daha uygundur [49].

Böylece, görüntü piksellerimizi (yeni matrisimiz “features”) özel kümelere ayırdık ve her gözlemi atadığı kümenin indeksini elde ettik. Kümelenmiş görüntü vektörünün boyutunu tanımlamak için RGB görüntü boyutu vektörünü kullanarak indeksin vektörünü yeniden şekillendirdik ve kümelenmiş görüntü vektörünü normalize ettik.

Elde edilen görüntü tümlendi ve genellikle bir gri tonlamalı görüntünün tümleyicisi olan  $J$ ,  $255-J$ 'ye eşittir. Aynı prensibi uyguladık, fakat akciğer alanlarını siyah lezyonları beyaza almamıza yardımcı olacak görüntünün ne zaman tamamlanmayıp tamamlanmayacağını seçmek için belirleyen bir algoritma gerçekleştirdik.

Sonra eşikli görüntü genişletildi. *Imdilate* fonksiyonu nesnelere boyutunu artırır ve böylece en küçük ayrıntıları daha görünür kılar. Daha sonra ilgilendiğimiz nesnelere tutulması gereken minimal bir yüzey tanımlayarak bir bölge filtreleme kullanıldı. Alan minimumu 250 olarak belirlendi. Üçüncü bir dalgacık ayrıştırma düzeyi kullanılarak CXR görüntümüzün dalgacık dönüşümü analizi uygulanmıştır. Bu ayrışma sonucu aşağıda bahsedilen tekstür özelliklerinin çıkarılması için kullanılmıştır.

### 3.3. Özellik çıkarma

Görüntüleri yeterli hale getirmek için bölütleme ve ön-işleme işlemlerinden sonra özellikleri çıkardık. Bizim durumumuzda, izole ettiğimiz yapılar tuberküloz lezyonları çıkarma durumudur. Bu adım tüberküloz lezyonları karakterize etmeyi ve sonrasında bir sınıflandırma yapmayı sağlar. Çeşitli özellik tiplerini, tanımlanacak nesnelere bağlı olarak çıkartılabilir, Bunlar: morfolojik, tekstür, geometrik ve görünüm özelliği, bölge özellik özellikleri, şekil tanımlayıcı özellikleri, entropi değerleri ve renktir [50, 51].

Bizim çalışmamızda morfolojik, şekil, tekstür ve geometrik koordinat özelliklerine odaklandık. Morfolojik ve şekil özellikleri olarak çıkarıldı, bunlar: boyut, her görüntüdeki lezyon sayısı, alan, sınır özellikleri (çevre), eliptik özellikler (ana ve küçük eksen uzunluğu, eksantriklik), euler sayısı, sağlamlık, genişlik ve ağırlık merkezidir. Tekstür özellikleri için entropi, kontrast, enerji, korelasyon, homojenlik, ortalama, standart sapma, RMS, varyans, pürüzsüzlük, çarpıklık, basıklıktan yararlandık. Ve geometrik özellikler olarak her biri CXR görüntüsündeki her lezyonun ağırlık merkezlerinin (centroid) ortalamasını hesapladık ve görüntünün lezyonların konumu olarak tanımladık. Yani bu konumdan apsiler, ordinatlar ve bunların standart sapmaları tanımlanmıştır. Output için görüntülerin sırasıyla teşhisini ekledik. Çıkarılan özelliklerle bir eğitim ve test matrisi oluşturulmuştur. Bu matris, eğitim ve test tablolarına dönüştürüldü.

### 3.4. Sınıflandırma

CXR görüntü analizi için bir sınıflandırıcı seçimindeki ana hususlardan biri çok yoğun veri kümeleriyle ve ayrıca geniş bir özellik vektörüyle başa çıkabilmesidir. Sınıflandırma ile ilgili üç adım attık. İlk adım segmentasyon ve özellik çıkarımıdır, bölüm 3.2 ve 3.3'te açıklanmıştır. Eğitim matrisi çıkarılan özelliklerden oluşmuş ve her sınıfın etiketli örnekleri verilmiştir. İkinci adımda bir eğitim setinde sınıflandırma modelleri (tahmin fonksiyonları) eğittik ve Matlab Classification Learner'i kullanarak bağımsız geçerlilik sınavı ile performans değerlendirmelerini yaptık.

Modellerimizin doğrulanması için holdout doğrulama ve çapraz doğrulama kullanmıştır. Holdout veri kümesini bir "eğitim" ve "test" kümesine ayırdığımız zamandır. Eğitim seti, modelin eğitildiği şeydir ve test seti, modelin görünmeyen veriler üzerinde ne kadar iyi performans gösterdiğini görmek için kullanılır [52]. Eğitim için verilerimizin % 75'ini ve test için verilerin kalan% 25'sini kullanmıştır. Çapraz doğrulama yönteminde k eğitim seti eşit parçalara bölünür ve bunlardan biri doğrulama seti olarak kullanılır. Kalan k-1 parçaları eğitim setini oluşturur. Her seferinde doğrulama seti olarak farklı bir bölüm kullanılır. Bu işlem k kez tekrarlanır ve her seferinde doğrulama setindeki başarısı hesaplanır. Böylece k başarı değeri elde edilir. Elde edilen ortalama k başarı değerleri modelin başarısı olarak kabul edilir [53].

Üçüncü adım, topluluk altuzay ayırt edici yöntem olarak (denetimli makine öğrenmesi) adım 2'den en iyi performans gösteren sınıflandırıcı olarak kullanan sınıflandırma geliştirmiştir.

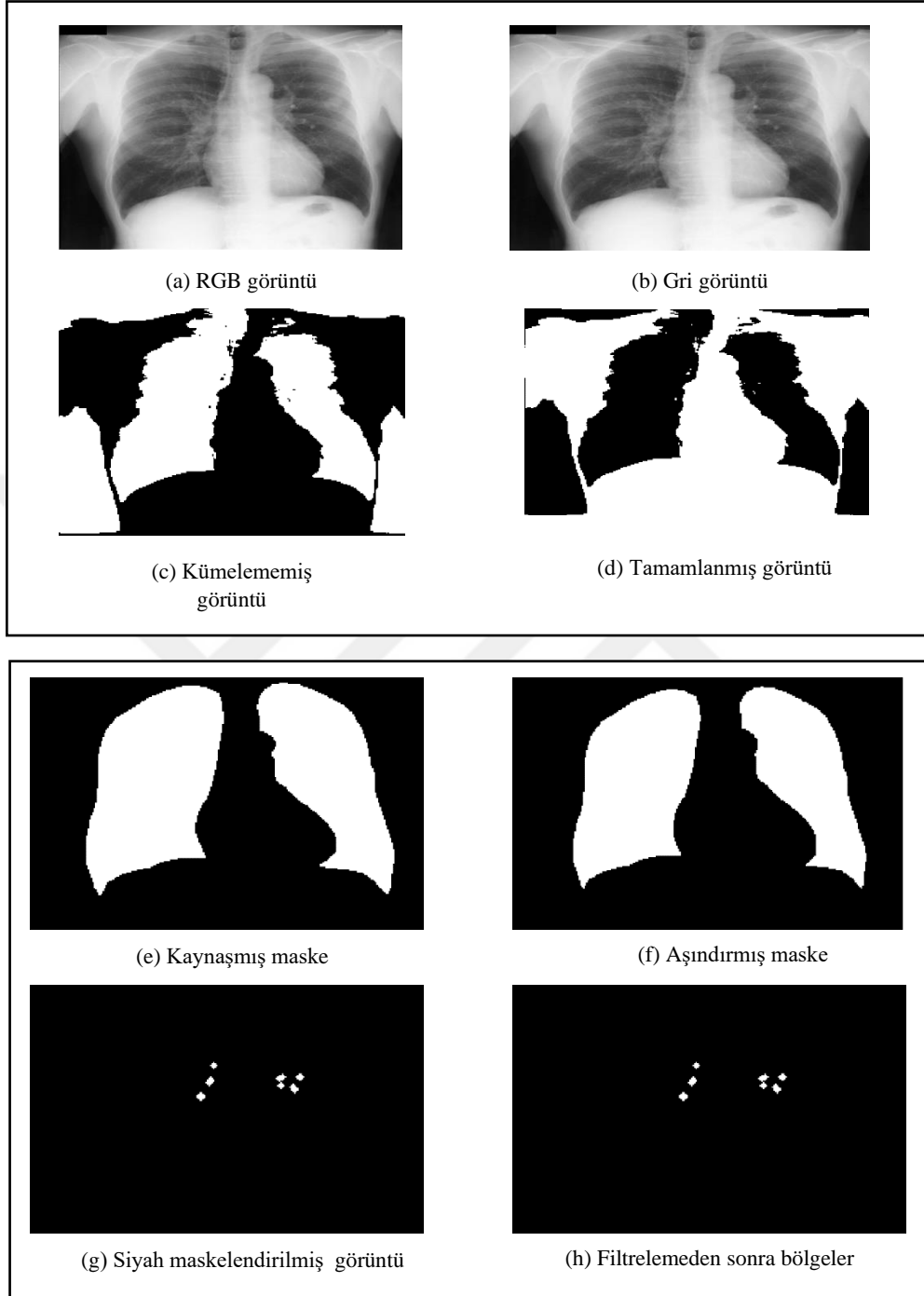
Testimizin başarı oranlarını sunmak için duyarlılığı (Se), özgüllüğü (Sp) ve olasılık oranını (RV), AUC ve doğruluk hesaplamamız gerekir. Pozitif bir olasılık oranı (RV+) ve negatif bir olasılık oranı (RV-) vardır.

## 4. SONUÇLAR

### 4.1. Bölütleme

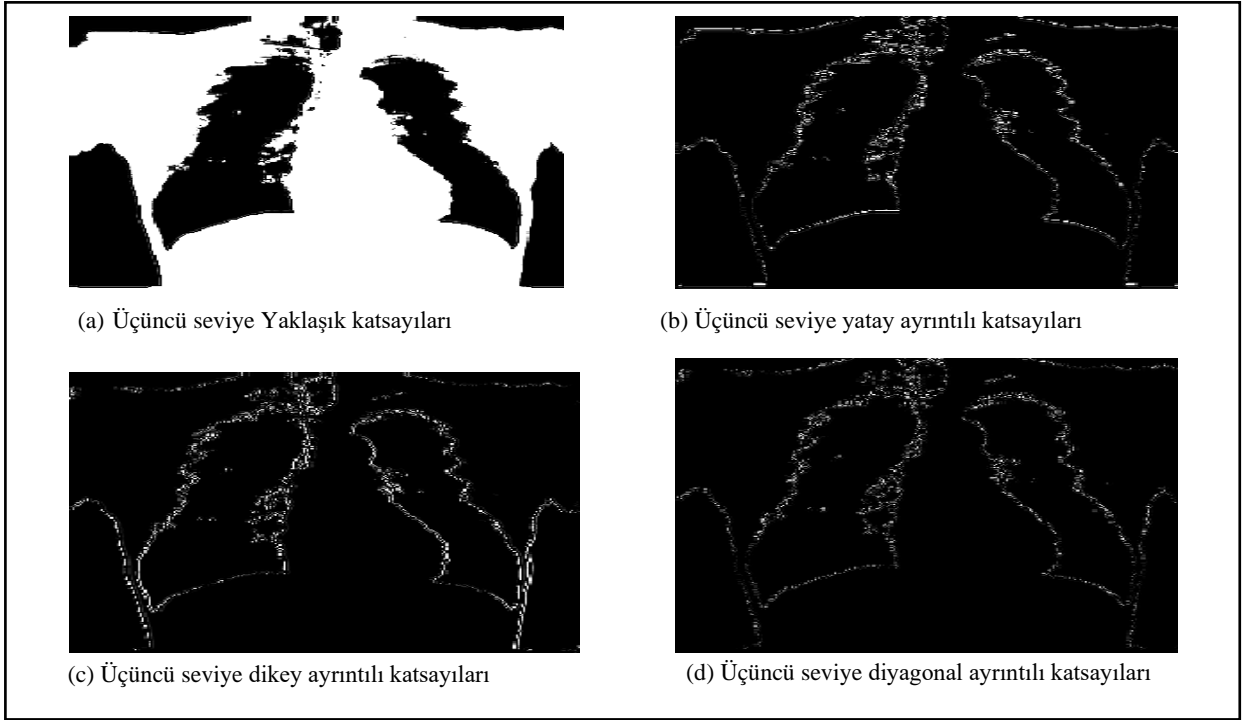
Bir renk işleme ve k-means kümelemesi içeren bir kümeleme bölütlemesi algoritması geliştirdik. CXR görüntüsüne uygulanan kümeleme algoritmasının sonucu 0 ve 255 değerlerinde iki grup piksel değerine sahip gri tonlamalı bir görüntüdür. 0 piksel değerleri Şekil 4.1.c'de gösterildiği gibi siyah alanları ve 255 ise beyaz alanları göstermektedir.

Tamamlanmış görüntü, akciğer alanlarını siyah ve akciğer lezyonlarını beyaz olarak görebildiğimiz Şekil 4.1.d'de gösterilmektedir.



**Şekil 4.1:** RGB ve Gri tonlardaki CXR görüntülerine ait bölütleme uygulanması (a-h).

Şekil 4.2 dalgacık ayrışmasının üçüncü seviyesini göstermektedir. Yaklaşık katsayıları, yatay ayrıntılı katsayıları, dikey ayrıntılı katsayıları ve diyagonal ayrıntılı katsayıları sırasıyla Şekil 4.2 a, b, c ve d'de sunulmuştur



Şekil 4.2: Üçüncü seviye Dalgacık ayrışması sonuçları (a-d).

## 4.2. Özellik çıkarma

Çıkarılan özelliklerden bir matris elde edildi ve ardından yeniden gruplandı. Daha sonra sınıflandırma işlemi sırasında daha uygun bir tabloya dönüştürülür. Bu tabloda sütunlar farklı değişkenleri temsil eder ve satırlar görüntüye karşılık gelir. İlk yirmi beş sütun girişleri (input) ve son sütun çıkışı göstermektedir (output). Şekil 4.3’de bu veri setinin bir kısmı gösterilmektedir. Tüm inputlar sayısal değerlerdir ve karşılık gelen output bir karakter değeridir (etiket).

	1	2	3	4	5	6	7	8	9
	TrainArea	TrainPerimeter	TrainMajorAxisLength	TrainMinorAxisLength	TrainEccentricity	TrainEulerNumber	TrainSolidity	TrainExtent	TrainAbs
1	20.3750	10.1633	4.4398	2.1429	0.1716	0.3750	0.3478	0.1987	
2	0	0	0	0	0	0	0	0	
3	19.8000	10.7416	4.4703	2.4212	0.3361	0.4000	0.3602	0.2151	
4	19.8000	10.7416	4.4703	2.4212	0.3361	0.4000	0.3602	0.2151	
5	28	14.2416	5.8817	3.5878	0.3409	0.6364	0.6140	0.3624	
6	31.6000	18.8896	8.3938	3.1554	0.4593	0.6000	0.5457	0.3243	
7	34.7273	17.0393	7.2968	3.9047	0.4652	0.6364	0.5967	0.3501	
8	34.8889	17.5636	6.8157	3.5819	0.2258	0.5556	0.5061	0.2980	
9	28	18.4680	8.2720	4.6616	0.8261	1	0.9655	0.5000	
10	21.6667	13.0562	4.9290	3.8109	0.2813	0.5556	0.6398	0.3851	
11	13.2500	6.3475	2.3644	1.8910	0.1501	0.2500	0.2366	0.1472	

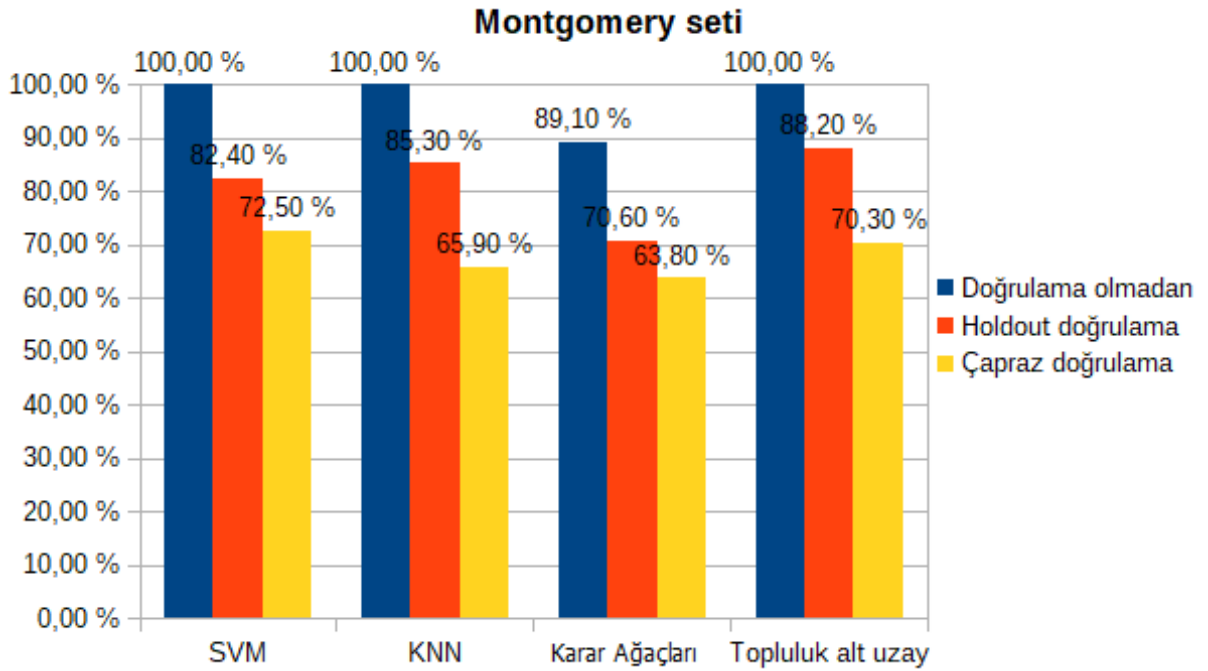
Şekil 4.3. Eğitimde kullanılan veri setinin bir kısmı.

### 4.3. Sınıflandırma

Eğitim adımlarımızın sonucu üç farklı yöntemle değerlendirilmiştir, bunlar: doğrulamasız (sadece eğitim), holdout doğrulama ve çapraz doğrulamadır.

Şekil 4.4 eğitim sonrası sonuçları ve doğrulama ile elde edilen test işlemlerine ait en verimli yöntemleri sunmaktadır.

Sadece eğitimin olduğu test işleminin yapılmadığı yöntemlere bakıldığında destek vektör makinesi, k-en yakın komşular (KNN) ve topluluk alt uzay yöntemleri için % 100 doğruluk elde edildi. Ağaç yönteminin % 89.10 doğruluğu gözlemlendi. Doğrulama başarıları için, holdout doğrulamada topluluk alt uzay yönteminin en verimli doğruluğa sahip olduğu (% 88.20) görüldü, bundan sonra KNN (% 85.30), SVM (% 82.40), Karar Ağaçları (% 70.60) yöntemi gelmiştir. Ancak çapraz doğrulamada SVM, topluluk alt uzay, KNN ve ağaç yöntemleri için sırasıyla % 72.50,% 70.30,% 65.90 ve% 63.80 bir doğruluk sağlanmıştır.



Şekil 4.4: Farklı sınıflandırıcıların doğruluk oranlarının karşılaştırılması.

**Tablo 4.1.** Holdout doğrulama için sınıflandırma algoritmalarının karışıklık matrisi

		<b>SVM</b>		$\Sigma$			<b>KNN</b>		$\Sigma$
		Tahmin					Tahmin		
		Tüberküloz	Normal				Tüberküloz	Normal	
Gerçek	Tüberküloz	10	4	14	Gerçek	Tüberküloz	16	2	18
	Normal	2	18	20		Normal	3	13	16
$\Sigma$		12	22	34	$\Sigma$		19	15	34

		<b>Karar Ağaçları</b>		$\Sigma$			<b>Topluluk altuzay</b>		$\Sigma$
		Tahmin					Tahmin		
		Tüberküloz	Normal				Tüberküloz	Normal	
Gerçek	Tüberküloz	8	6	14	Gerçek	Tüberküloz	11	3	14
	Normal	4	16	20		Normal	1	19	20
$\Sigma$		12	22	34	$\Sigma$		12	22	34

**Tablo 4.2.** 5 kat çapraz doğrulama için sınıflandırma algoritmalarının karışıklık matrisi

		<b>SVM</b>		$\Sigma$			<b>KNN</b>		$\Sigma$
		Tahmin					Tahmin		
		Tüberküloz	Normal				Tüberküloz	Normal	
Gerçek	Tüberküloz	28	30	58	Gerçek	Tüberküloz	26	32	58
	Normal	8	72	80		Normal	15	65	80
$\Sigma$		36	102	138	$\Sigma$		41	97	138

		<b>Karar Ağaçları</b>		$\Sigma$			<b>Topluluk altuzay</b>		$\Sigma$
		Tahmin					Tahmin		
		Tüberküloz	Normal				Tüberküloz	Normal	
Gerçek	Tüberküloz	32	26	58	Gerçek	Tüberküloz	24	34	58
	Normal	24	56	80		Normal	7	73	80
$\Sigma$		56	82	138	$\Sigma$		31	107	138

Tablolar 4.1 ve 4.2 sırasıyla farklı sınıflandırma yöntemleri (SVM, KNN, Karar Ağaçları ve topluluk altuzay) ile holdout doğrulama ve çapraz doğrulaması ile karışıklık matrisini temsil etmektedir. Bu karışıklık matrisi gözlemleri, gerçek pozitif (TP) oranları, gerçek negatif (TN) oranları, yanlış pozitif (FP) oranları ve yanlış negatif (FN) oranları vermektedir.

**Tablo 4.3.** Holdout doğrulama ve çapraz doğrulama üzerinde SVM, KNN, Karar Ağaçları ve topluluk alt uzayının sınıflandırma performansı ölçüsü.

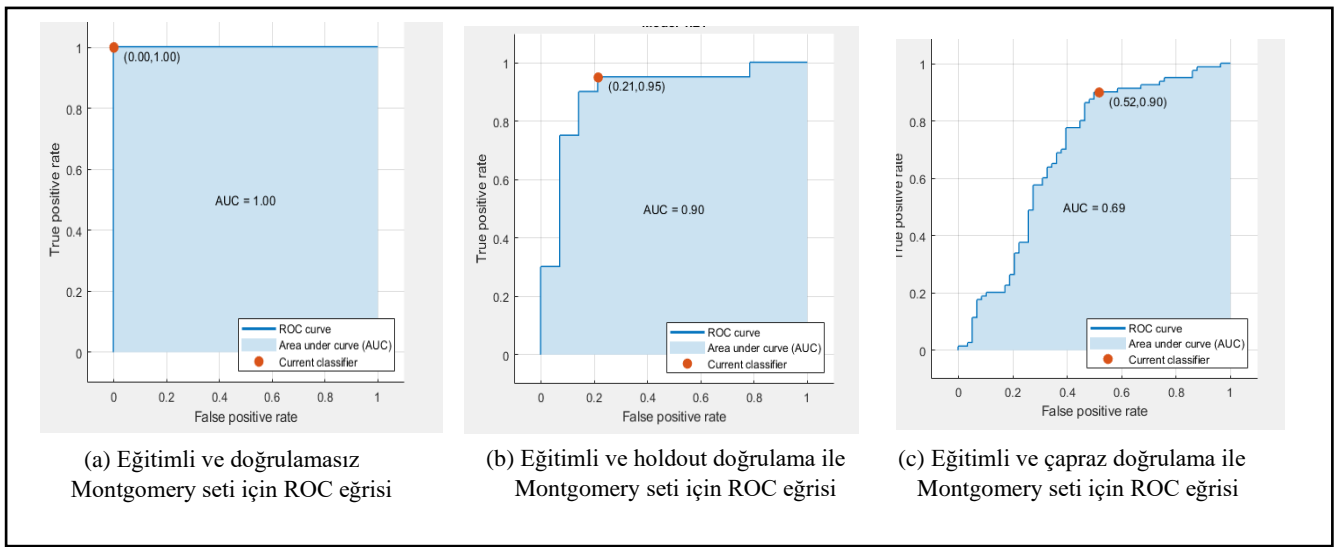
Sınıflandırıcı	Doğrulama	Duyarlılık	Özgüllük	RV+	RV-
SVM	Holdout doğrulama	0.83	0.82	4.58	0.20
	Çapraz doğrulama	0.78	0.71	2.64	0.31
	Doğrulasız	1.00	1.00	-	0
KNN	Holdout doğrulama	0.84	0.87	6.32	0.18
	Çapraz doğrulama	0.63	0.67	1.92	0.55
	Doğrulasız	1.00	1.00	-	0
Karar Ağaçları	Holdout doğrulama	0.67	0.73	2.44	0.46
	Çapraz doğrulama	0.57	0.68	1.80	0.63
	Doğrulasız	0.91	0.88	7.43	0.09
Topluluk altuzay	Holdout doğrulama	0.92	0.86	6.72	0.10
	Çapraz doğrulama	0.77	0.68	2.44	0.33
	Doğrulasız	1.00	1.00	-	0

Makine öğrenimi (SVM, KNN, Ağaç ve Topluluk altuzay) yöntemlerinden elde edilen sınıflandırma sonuçları Tablo 4.3'de gösterilmektedir. Bu sonuç, doğrulamalı ve doğrulasız da sunmaktadır.

Tablo 4.3'deki doğrulamalı sonuçlara göre Topluluk altuzay holdout doğrulamada en başarılı sonuçları ( $Se=0.92$ ,  $Sp=0.86$ ,  $RV+=6.72$  ve  $RV-=0.10$ ) vermektedir. Çapraz doğrulama ile elde edilen en iyi sınıflandırıcı SVM olmuştur ( $Se=0.78$ ,  $Sp=0.71$ ,  $RV+=2.64$  ve  $RV-=0.31$ ).

Doğrulama olmadan en iyi sonuçlar SVM, KNN ve Topluluk altuzay ile elde edilir ( $Se=1.00$ ,  $Sp=1.00$  ve  $RV=0$ ).

Şekil 4.5 a, b ve c, doğrulama olsun veya olmasın eğitilmiş modelimizin (Topluluk altuzay) ROC eğrilerini göstermektedir. Doğrulamasız modeli Şekil 4.5.a'dadır ve Şekil 4.5.b ve c sırasıyla holdout doğrulaması ve çapraz doğrulama ile ROC eğrisini gösterilmektedir. Doğrulama olmadan eğrinin altında kalan alan (AUC) 1.00'dir ve en yüksek olan değerdir. Çapraz doğrulama (cross validation) ( $AUC = 0.69$ ) ile karşılaştırıldığında, holdout doğrulaması 0.90 ile en büyük AUC'yi vermiştir.



Şekil 4.5: Montgomery veri seti için eğitim sonrası elde edilen ROC eğrileri (a-c).

Tablo 4.4: Oluşturulan modelin eğitim ve tahmin süreleri.

Eğitim/Test	Montgomery seti	
	Tahmin hızı (obs/saniye)	Eğitim Süresi (saniye)
Doğrulama olmadan	36000	0.15
Holdout doğrulama	580	0.61
Çapraz doğrulama	7300	0.29

Topluluk alt uzayını (elde edilen en iyi yöntem) yöntem olarak kullandıktan sonra, tahmin hızı ve eğitim süresi Tablo 4.2'de verilmiştir. Bu tablo, doğrulaması olmayanları da göstermektedir. En iyi sonuç doğrulama olmadan elde edilmiştir (tahmin hızı = -36000 gözlem / sn; eğitim süresi = 0,15 sn).

## 5. TARTIŞMA

Segmentasyon, bir CXR bilgisayar destekli teşhisi uygulamak için başarının anahtarı olmaya devam etmektedir. Makine öğrenimi sonucu kullanabileceğimiz yöntemler ne olursa olsun, segmentasyon adımına bağlıdır.

Birçok çalışmanın aksine tüberküloz lezyonlarını bu çalışmada izole etmeyi başardık. Ginneken V. ve diğerleri [9] gibi görüntüyle bağlantılı özelliklerin (görüntünün morfolojik özelliği, entropi vb.) çıkarılmasıyla kendimizi sınırlamaktan kaçındık. Pulmoner lezyonların morfolojik özelliklerini ve yerlerini çıkarabildik (materyal ve yönteme bakınız). Bunlar sınıflandırmamızı daha uygun hale getirmemizi ve daha iyi doğruluklar elde etmemizi sağlamıştır.

Performans doğrulaması olmadan elde ettiğimiz sonuçlar, aynı veri seti (Montgomery veri seti) kullandığımız için Sameer'in sonucundan [54] daha iyidir. % 95,57 olarak bulunduğu SVM yöntemi için %100 doğruluğa ulaşılmıştır. Seçtiğimiz sınıflandırıcı yöntemimiz (topluluk alt uzay) 1.00 eğrisinin altında bir alan göstermiş olup Sameer A.'nın bulunduğu değer 0.99 olmuştur.

Diğer çalışmaların aksine, performans doğrulamalı sonuçları da bu çalışmada sunduk. Holdout doğrulamasıyla % 88,20'lik bir doğruluk ve 0,90'lık bir AUC elde ettik. Bu sonuçlar Sameer A. tarafından sunulan doğrulamasız sonucuna da yakındır (Doğruluk =% 95,57 ve AUC = 0,99) [54].

Sonuçlarımızda, kalıcı doğrulama ve çapraz doğrulama sonrasında elde edilen doğruluklar arasında önemli bir fark bulduk. Bu farklılıklar neredeyse %20'ye ulaşmaktadır. Holdout doğrulama (tek veri setinin %25'i) çapraz doğrulamadan (5 kat çapraz doğrulama = veri setinin %20'si. Bütün veri seti kullanır ) daha küçük bir veri setinin bölümünü kullanmasına neden olabilir.

Performans doğrulaması olmadan test performansımızın duyarlılığı ve özgüllüğü mükemmelliğe ulaşmıştır, 1.00 gibi. Ancak Sameer'in çalışmasında duyarlılık 0,99'dur. Doğrulama durumunda, holdout doğrulama bize daha yüksek duyarlılık değeri 0.92 ve özgüllük 0.86 verdi.

Doğrulama performansının olmaması durumunda en yüksek tahmin hızları ve eğitim süreleri rapor edilmiştir. Bu durum, veri setinin teste katılacak bölümünü doğrulamak ve seçmek için gereken süreye neden olacaktır.

## 6. SONUÇ VE ÖNERİLER

Sonuç olarak, bu çalışma akciğer hastalığının (özellikle tüberküloz) sınıflandırılmasını ve yapay zekanın makine öğrenimi ile uygulanmasını geliştirmek için yeni bir segmentasyon şeması sunmaktadır. Elde edilen sonuçlar tatmin edicidir ancak uygulamanın daha büyük veri tabanlarında test edilmesi gerekir ve bu nedenle bazı optimizasyonlar yapılabilir. Bu da çalışmada önemli bir sınırlama kabul edilmelidir.

Zaman ve kaynak kısıtlaması nedeniyle bazı akciğer hastalıkları dahil edilememiştir. Bunun üzerine CXR görüntülerini analiz edebilen ve özellikle tüberkülozu teşhis edebilen bir uygulama hayata geçirdik. Bu uygulama bir CXR görüntüsünde tüberkülozu okuyabilen ve teşhis edebilen sağlık personeli eksikliği sorununa cevap vermemizi sağlar. Özellikle kaynakları sınırlı ve tüberkülozdan çok etkilenen bölgelerde, en mütevazı hastanelerin ulaşabileceği düşük maliyetli bir uygulamadır.

Bu tezin hazırlanmasında kullanılan kaynak kodları Ekler bölümünde mevcuttur. Segmentasyon, öznitelik çıkarımı ve sınıflandırma yapılırken kullanılan kodlar orijinal halleriyle eklenmiştir.

**KAYNAKLAR**

- [1]. World Health Org., Global Tuberculosis Report 2018.
- [2]. World Health Organization. Tuberculosis Prevalence Surveys: A Handbook. 2011.
- [3]. Lodwick G, Keats T, Dorst J. The coding of Roentgen images for computer analysis as applied to lung cancer. *Radiology*, 81(2):185, 1963.
- [4]. Sakai S., Soeda H., Takahashi N. et al. Computer-aided nodule detection on digital chest radiography: Validation test on consecutive T1 cases of resectable lung cancer. *J. Digital Imaging*, 19(4):376–82, 2006.
- [5]. Kakeda S, Moriya J, Sato H. et al. Improved detection of lung nodules on chest radiographs using a commercial computer- aided diagnosis system. *Am. J. Roentgenol.*, 182(2):505-10, 2004.
- [6]. Van Ginneken B., Stegmann M,Loog M. Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database. *Med. Image Anal.*, 10(1):19–40, 2006.
- [7]. Van Ginneken B., Ter Haar Romeny B. Automatic segmentation of lung fields in chest radiographs. *Med. Phys.*, 27(10):2445-55, 2000.
- [8]. Dawoud A. Fusing shape information in lung segmentation in chest radiographs. *Image Anal. Recognit.*, pp. 70–8, 2010.
- [9]. Van Ginneken B., Katsuragawa S., Ter Haar Romeny B. et al. Automatic detection of abnormalities in chest radiographs using local texture analysis. *IEEE Trans. Med. Imag.*, 21(2):139–49, 2002.
- [10]. Hogeweg L., Dawson R., Van Ginneken B. et al. Fusion of local and global detection systems to detect tuberculosis in chest radiographs. *Proc. MICCAI*, pp. 650–57, 2010.
- [11]. Shen R., Cheng I., Basu A. A hybrid knowledge-guided detection technique for screening of infectious pulmonary tuberculosis from chest radiographs. *IEEE Trans. Biomed. Eng.* 57(11): 2646–56, 2010.
- [12]. Xu T., Cheng I., Mandal M. Automated cavity detection of infectious pulmonary tuberculosis in chest radiographs. *Int. IEEE Eng. Med. Biol. Soc.*, pp. 5178–81, 2011.
- [13]. Hogeweg L., Sánchez CI., van Ginneken B. et al. Clavicle segmentation in chest radiographs. *Med. Image Anal.* 16(8):1490–1502, 2012.
- [14]. Freedman M., Lo S., Seibel J. et al. Lung nodules: Improved detection with software that suppresses the rib and clavicle on chest radiographs. *Radiology*, 260(1): 265–73, 2011.
- [15]. Arzhaeva Y, Tax D, van Ginneken B, “Dissimilarity-based classification in the absence of local ground truth: Application to the diagnostic interpretation of chest radiographs. *Pattern Recognition*, 42(9):1768–76, 2009.

- [16]. Jaeger S, Karargyris A, Candemir S, Siegelman J, Folio L, Antani S, Thoma G. Automatic screening for tuberculosis in chest radiographs: a survey. *Quant Imaging Med Surg.* 2013; 3(2):89-99.
- [17]. Lelo T, Ndoma K. *Cours d'imagerie médicale*, p. 9-11, 2010.
- [18]. Mukaya J. *Guide pratique d'imagerie médicale d'auto-apprentissage*, p. 5-6, 2005.
- [19]. World Health Org., *Global Tuberculosis Report 2014*.
- [20]. Stop TB Partnership, World Health Organization. *The Global Plan to Stop TB 2011–2015*.
- [21]. Lee KS, Song KS, Lim TH, et al. Adult-onset pulmonary tuberculosis: findings on chest radiographs and CT cans. *AJR Am J Roentgenol.* 160:753-8, 1993.
- [22]. Lelo T, Ndoma K. *Cours d'imagerie médicale*, p. 67-68, 2010.
- [23]. Leung C. Reexamining the role of radiography in tuberculosis case finding. *Int. J. Tuberculosis Lung Disease*, 15(10):1279, 2011.
- [24]. Mohssen M, Muhammad B, Eihab B. *Machine learning algorithms and applications.* 13:978-1-4987-0538-7,2017.
- [25]. Judith H., Daniel K. *Machine Learning.* 978-1-119-45495-3, 2018.
- [26]. Osisanwo F.Y., Akinsola J.E.T, Awodele O., et al. Supervised machine learning classification and comparaison. *Int. J. of Computer Trends and Technology-Volume 48 Number 3 June 2017.*
- [27]. Kotsiantis. S.B. *Supervised Machine Learning: A Review of Classification Techniques, Informatica 31 (2007) 249-268.*
- [28]. Murthy S.K. *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, Data Mining and Knowledge Discovery 2:345–389, 1998.*
- [29]. Quinlan, J.R. *C4.5: Programs for machine learning.* Morgan Kaufmann, San Francisco,1993.
- [30]. Furnkranz, J. *Pruning algorithms for rule learning. Machine Learning 27: 139-171,1997.*
- [31]. Camargo, L. S. & Yoneyama, T. *Specification of Training Sets and the Number of Hidden Neurons for Multilayer Perceptrons. Neural Computation 13: 2673–2680, 2001.*
- [32]. Mitchell, T. *Machine Learning.* McGraw Hill. 1997.
- [33]. Cover, T., Hart, P. *Nearest neighbor pattern classification. IEEE Transactions on Information, Theory, 13(1): 21–7, 1967.*

- [34]. Li-Yu H., Min-Wei H., Shih-Wen K., et al. The distance function effect on k-nearest neighbor classification for medical datasets, DOI 10.1186/s40064-016-2941-7, 2016.
- [35]. Cristianini, N. & Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, University Press, Cambridge, 2000.
- [36]. Veropoulos K., Campbell, C. & Cristianini, N. Controlling the Sensitivity of Support Vector Machines. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99), 1999.
- [37]. Scholkopf, C., Burges, J. C. & Smola, A. J. Advances in Kernel Methods. MIT Press, 1999.
- [38]. Amira S. Ashour, Yanhui Guo. (2018). Ensemble of subspace discriminant classifiers for schistosomal liver fibrosis staging in mice microscopic images, Health Information Science and Systems, DOI: 10.1007/s13755-018-0059-8, pp. 2-3.
- [39]. R. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996.
- [40]. Smola A., Vishwanathan S.V.N. Introduction to Machine Learning, 0 521 82583 0, 2010.
- [41]. Gamarra M., Zurek E., San-juan H. A study of image analysis algorithms for segmentation, feature extraction and classification of cells. Journal of information systems engineering and management, 2017, 2(4), 20. 2468-4376, 2017.
- [42]. MathWorks. Types of morphological operations. 2020.
- [43]. Nikhil R.P., Sankar K.P. A review on image segmentation techniques. Pattern recognition vol 26, No 9. Pp.1277-1294, 1993.
- [44]. Tonti, S., Di Cataldo, S., Bottino, A. and Ficarra, E. (2015). An Automated Approach to the Segmentation of HEp-2 Cells for the Indirect Immunofluorescence ANA Test. Computerized Medical Imaging and Graphics: The Official Journal of the Computerized Medical Imaging Society, 40, pp. 62–69.
- [45]. Candemir, S., Jaeger, S., Palaniappan, K., Musco, J. P., Singh, R. K., Xue, Z., Karargyris A., Antani S., Thoma G., McDonald, C. J. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. Medical Imaging, IEEE Transactions on, 33(2), 577-590, 2014.
- [46]. Chitra S, Balakrishnan G. Comparative Study of Two Color Spaces HSCbCr and YCbCr in skin color detection. Appl Math Sci. 2012; 6:4229–38.
- [47]. Joblove, G. H., Greenberg, D.: Color Spaces for Computer Graphics. ACM SIGGRAPH Computer Graphics, Vol. 12, No. 3. ACM (1978).

- [48]. Plataniotis, K. N., Venetsanopoulos, A. N.: *Colour Image Processing and Applications*. Springer (2000).
- [49]. Matworks. K-Means Clustering R2019B.
- [50]. Liu, Q. and Liu, X. (2013). Feature Extraction of Human Viruses Microscopic Images using Gray Level Co-Occurrence Matrix. In 2013 International Conference on Computer Sciences and Applications. IEEE, pp. 619–622.
- [51]. Dogantekin E., Avci E. and Erkus, O. (2013). Automatic RNA Virus Classification using the Entropy-ANFIS Method. *Digital Signal Processing*, 23(4), pp. 1209–1215.
- [52]. Awwalu J. ve ark. Validation A Comparison between Neural Network and Support Vector Machine. *International Journal of Trend in Research and Development*, Volume 6(2), ISSN: 2394-9333, p.235 April 2019.
- [53]. Akgündoğdu A., Comparaison of Deep Learning and Machine Learning Methods to Classify Retinal Images. *International conference on engineering technologies (ICENTE'20)*, p.6. Kasım 2020.
- [54]. Sameer A. ve ark. (2015). Automated detection of lung diseases in chest x-rays. A report to the board of scientific counselors, pp 8-12.

## EKLER

### Clustering image processing

```
% image segmentation using K-means clustering in matlab
function clustim=imseg(rgbim,k)
rgbim=imresize(rgbim,[180,270]);
[r,c,~]=size(rgbim);
ycb=rgb2ycbcr(rgbim);
hsv=rgb2hsv(rgbim);
im2=ycb(:,:,1); % peut etre modifier(1,2 ou 3) selon les détails qu'on veut
sortir de l'image
im3=ycb(:,:,3); % peut etre modifier(1,2 ou 3) selon les détails qu'on veut
sortir de l'image
im21=hsv(:,:,3); % peut etre modifier(1,2 ou 3) selon les détails qu'on
veut sortir de l'image
features=[];
features(:,1:2)=[im2(:),im3(:)];
features(:,3)=im21(:);
[ind,~]=kmeans(features,k);
clustim=reshape(ind,[r,c]);
clustim=(clustim-min(clustim(:)))/(max(clustim(:))-min(clustim(:)))*255;
% data between 0 and 1. multiplier par 255 pour obtenir un echelle gray
donc ainsi
% notre data va varier de 0 à 255
clustim=uint8(clustim);
```

### Image processing second step

```
function [GrayImage,smoothGradient,seg2]=ImageProcessing1(rgbim)

% % % % % % % % % % % % % % % % % % % % % % % % Clustering

% imseg=clustering
seg=imseg(rgbim,2);% clustered image and k(2) might be modified according
wanted classes number( black and white or black, white and gray=3, and so
more)
figure,imshow(seg);title('seg before imcomplement')
intensity_value_left1=seg(70,70);
intensity_value_left2=seg(90,90);
intensity_value_right=seg(180,60);
if intensity_value_left1==255 || intensity_value_left2==255 &
intensity_value_right==255
seg1=imcomplement(seg);
else seg1=seg;
end
seg2=imcomplement(seg1); % seg2 has been creating for being using
in similarity value (ssimval) calculation.

cent=findcen(seg);
figure;
figure,imshow(seg1);title('seg1 after imcomplement')

hold on
plot(cent(1,2),cent(1,1),'r. ');
% features(total_images,:)=cent; % La matrice features contiendra les
coordonées du centre 'cent' de chaque image
```



```

Mean = mean2(G);
Standard_Deviation = std2(G);
Entropy = entropy(G);
RMS = mean2(rms(G));
Variance = mean2(var(double(G)));
a = sum(double(G(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(G(:)));
Skewness = skewness(double(G(:)));
end

```

### Features extractions: find centroid of each lésions

```

function center=findcen(bwimage)
a=imresize(bwimage,[180,270]);
[r,c]=size(a);
cor=[];
count=1;
for i=1:r
    for j=1:c
        if a(i,j)==255
            cor(1,count)=i;
            cor(2,count)=j;
            count=count+1;
        end
    end
end
center=[sum(cor(1,:))/count,sum(cor(2,:))/count];
center=uint16(center);

```

### fuse mask

```

function [fusedmask]=fusemask(maskR_Gray,maskL_Gray)

```

```

maskR_Gray=double(maskR_Gray);
maskL_Gray=double(maskL_Gray);

```

```

fusedmask=maskR_Gray+maskL_Gray;
fusedmask=logical(fusedmask);
end

```

### Mask segmentation

```

function [blackMaskedImage]=MaskSegmentationTrain(GrayImage,BinaryMask)

```

```

blackMaskedImage = GrayImage;
blackMaskedImage(~BinaryMask) = 0;
figure,imshow(blackMaskedImage,[],),title('Masked Outside Image');

end

```

### Features computing, mask segmentation and extractions

```

close all,clear all,clc

```

```

% Montgomery
image_folder='F:\THESE
AUDRY\Database04112020\MontgomeryCXRSet\MontgomerySet\CXR_png';
mask_folderR='F:\THESE
AUDRY\Database04112020\MontgomeryCXRSet\MontgomerySet\ManualMask\rightMask'
;
mask_folderL='F:\THESE
AUDRY\Database04112020\MontgomeryCXRSet\MontgomerySet\ManualMask\leftMask';

% Reading

% Images
filenamesImg=dir(fullfile(image_folder, '*.*png'));
total_images=numel(filenamesImg);

% Mask
% Right mask

filenamesMskR=dir(fullfile(mask_folderR, '*.*png'));
total_maskR=numel(filenamesMskR);

% Left mask

filenamesMskL=dir(fullfile(mask_folderL, '*.*png'));
total_maskL=numel(filenamesMskL);

% Training and testing matrix
rowIndex=0;
train_index=1;

for i=1:total_images
    f=fullfile(image_folder, filenamesImg(i).name);
    rgbim=imread(f);
% Pre-processing
    Agray = rgb2gray(rgbim);
    img2 = im2bw(Agray);

% First image processing
    [GrayImage, smoothGradient, seg2]=ImageProcessing1(rgbim); % First step
    image processing

% Read and resize mask
    [Ir Ic]=size(GrayImage)
    g=fullfile(mask_folderR, filenamesMskR(i).name);
    maskR_BW=imread(g);
    maskR_BW=imresize( maskR_BW, [Ir Ic]);
    figure, imshow( maskR_BW)
    p=fullfile(mask_folderL, filenamesMskL(i).name);
    maskL_BW=imread(p);
    maskL_BW=imresize( maskL_BW, [Ir Ic]);
    figure, imshow( maskL_BW)

% Operation on mask (fuse, resize and erode mask)
    [fusedmask]=fusemask(maskR_BW, maskL_BW);
    figure, imshow(fusedmask)
    BinaryMask=imresize( fusedmask, [Ir Ic]);
    SE1=strel('disk', 3);

```

```

BinaryMask=imerode(BinaryMask,SE1);

figure, imshow(BinaryMask),title('BinaryMask')
LogicGrayImage=double(seg2);
LogicBinaryMask=double(BinaryMask);
[ssimval] = ssim(LogicGrayImage,LogicBinaryMask); % test similarity
fprintf('The SSIM value is %0.4f.\n',ssimval);

figure,imshow(smoothGradient,[], title('smoothGradient'))

% Mask segmentation
[blackMaskedImage]=MaskSegmentationTrain(GrayImage,BinaryMask);
SE2=strel('disk',2);
blackMaskedImage=imdilate(blackMaskedImage,SE2);
bmibw=im2bw(blackMaskedImage); % black masked image converted in binary

BW_out=bmibw;
figure,imshow(BW_out);

% % Features extraction

% Labeling
[Ilabel]=Labeling_2(rgbim,BW_out);

% Compute features
RegionArea = regionprops(Ilabel,'Area');%find Area
T_Area=struct2table(RegionArea);
M_Area=T_Area.Area;
% meanArea=mean(T_Area.Area);
% Area=meanArea;

RegionPerimeter = regionprops(Ilabel,'Perimeter');%find
Perimeter
T_Perimeter=struct2table(RegionPerimeter);
M_Perimeter=T_Perimeter.Perimeter;
% meanPerimeter=mean(T_Perimeter.Perimeter);
% Perimeter=meanPerimeter;

RegionMajorAxisLength =
regionprops(Ilabel,'MajorAxisLength');%find MajorAxisLength
T_MajorAxisLength=struct2table(RegionMajorAxisLength);
M_MajorAxisLength=T_MajorAxisLength.MajorAxisLength;
% meanMajorAxisLength=mean(T_MajorAxisLength.MajorAxisLength);
% MajorAxisLength=meanMajorAxisLength;

RegionMinorAxisLength =
regionprops(Ilabel,'MinorAxisLength');%find MinorAxisLength
T_MinorAxisLength=struct2table(RegionMinorAxisLength);
M_MinorAxisLength=T_MinorAxisLength.MinorAxisLength;
% meanMinorAxisLength=mean(T_MinorAxisLength.MinorAxisLength);
% MinorAxisLength=meanMinorAxisLength;

```

```

RegionEccentricity = regionprops(Ilabel, 'Eccentricity');%find
Eccentricity
T_Eccentricity=struct2table (RegionEccentricity);
M_Eccentricity=T_Eccentricity.Eccentricity;
% meanEccentricity=mean(T_Eccentricity.Eccentricity);
% Eccentricity=meanEccentricity;

RegionOrientation = regionprops(Ilabel, 'Orientation');%find
Orientation
T_Orientation=struct2table (RegionOrientation);
M_Orientation=T_Orientation.Orientation;
% meanOrientation=mean(T_Orientation.Orientation);
% Orientation=meanOrientation;

RegionConvexArea = regionprops(Ilabel, 'ConvexArea');%find
ConvexArea
T_ConvexArea=struct2table (RegionConvexArea);
M_ConvexArea=T_ConvexArea.ConvexArea;
% meanConvexArea=mean(T_ConvexArea.ConvexArea);
% ConvexArea=meanConvexArea;

RegionFilledArea = regionprops(Ilabel, 'FilledArea');%find
FilledArea
T_FilledArea=struct2table (RegionFilledArea);
M_FilledArea=T_FilledArea.FilledArea;
% meanFilledArea=mean(T_FilledArea.FilledArea);
% FilledArea=meanFilledArea;

RegionEulerNumber = regionprops(Ilabel, 'EulerNumber');%find
EulerNumber
T_EulerNumber=struct2table (RegionEulerNumber);
M_EulerNumber=T_EulerNumber.EulerNumber;
% meanEulerNumber=mean(T_EulerNumber.EulerNumber);
% EulerNumber=meanEulerNumber;

RegionEquivDiameter = regionprops(Ilabel, 'EquivDiameter');%find
EquivDiameter
T_EquivDiameter=struct2table (RegionEquivDiameter);
M_EquivDiameter=T_EquivDiameter.EquivDiameter;
% meanEquivDiameter=mean(T_EquivDiameter.EquivDiameter);
% EquivDiameter=meanEquivDiameter;

RegionSolidity = regionprops(Ilabel, 'Solidity');%find Solidity
T_Solidity=struct2table (RegionSolidity);
M_Solidity=T_Solidity.Solidity;
% meanSolidity=mean(T_Solidity.Solidity);
% Solidity=meanSolidity;

RegionExtent = regionprops(Ilabel, 'Extent');%find Extent

```

```

    T_Extent=struct2table(RegionExtent);
    M_Extent=T_Extent.Extent;
    % meanExtent=mean(T_Extent.Extent);
    % Extent=meanExtent;

%Find Lesion coordinates
    stat_Centroid = regionprops(Ilabel,'centroid');
    XY=[stat_Centroid.Centroid];
    imshow(Ilabel); hold on;

    % Trouvons le transposé de la matrice XY

    T=XY';

    % Selectionnons les abscisses (représentés par i impairs) et les
ordonées
    % (représentés par i pairs) car la matrice T (qui est la transposé de
la matrice XY 1 x i)
    % est une matrice i x 1

    A =T(1:2:end,:); % odd matrix = les i impairs de matrice T =
abscissas
    O = T(2:2:end,:); % even matrix = les i pairs de la matrice T =
ordonées

    A=round(A); % round all values of A matrix
    O=round(O); % round all values of O matrix

% Assign zero to all abscissa and ordinate out of the region of
interesting

%     matrix_index=1;
storagematrix=zeros(20,14);
    for s = 1: numel(stat_Centroid)
        % XX=A(s,1); % get lesion's abscissa
        % YY=O(s,1); % get lesion's ordinate

plot(stat_Centroid(s).Centroid(1),stat_Centroid(s).Centroid(2),'ro');
    XX1=(34:118); % abscissa condition matrix XX1
    XX2=(145:195); % abscissa condition matrix XX2
    YY=(53:130 ); % ordinate condition matrix YY

    XX1_position=find(XX1==A(s,1));% finding the position of the right
lung lesion's abscissa A(s,1) in the abscissa condition matrix XX1
    XX2_position=find(XX2==A(s,1)); % finding the position of the left
lung lesion's abscissa A(s,1) in the abscissa condition matrix XX2
    YY_position=find(YY==O(s,1)); % finding the position of the
lesion's ordinate O(s,1) in the ordinate condition matrix YY

    RRX1=XX1(1,XX1_position); % result:lesion's abscissa A(s,1) finded
in the condition matrix XX1 (if not finded the result is zero)
    RRX2=XX2(1,XX2_position); % result:lesion's abscissa A(s,1) finded
in the condition matrix XX2 (if not finded the result is zero)
    RRY=YY(1,YY_position); % result:lesion's ordinate O(s,1) finded in
the condition matrix YY (if not finded the result is zero)

```

```

if A(s,1)==RRX1|A(s,1)==RRX2& O(s,1)==RRY; % Help to exclude all
residual lesions which are out of region of interest

    s_Area=M_Area(s,1);
    s_Perimeter=M_Perimeter(s,1);
    s_MajorAxisLength=M_MajorAxisLength(s,1);
    s_MinorAxisLength=M_MinorAxisLength(s,1);
    s_Eccentricity=M_Eccentricity(s,1);
    s_Orientation=M_Orientation(s,1);
    s_ConvexArea=M_ConvexArea(s,1);
    s_FilledArea=M_FilledArea(s,1);
    s_EulerNumber=M_EulerNumber(s,1);
    s_EquivDiameter =M_EquivDiameter(s,1);
    s_Solidity = M_Solidity(s,1);
    s_Extent = M_Extent(s,1);
    X=A(s,1);
    Y=O(s,1);
else
    s_Area=0;
    s_Perimeter=0;
    s_MajorAxisLength=0;
    s_MinorAxisLength=0;
    s_Eccentricity=0;
    s_Orientation=0;
    s_ConvexArea=0;
    s_FilledArea=0;
    s_EulerNumber=0;
    s_EquivDiameter =0;
    s_Solidity =0;
    s_Extent =0;
    X=0;
    Y=0;
end

% Buildind matrix for storage all values by respectively parameters
storagematrix(s,1)=s_Area;
AreaMatrix=storagematrix(1:s,1); % selected the s first value of
AreaMatrix

storagematrix( s,2)=s_Perimeter;
PerimeterMatrix=storagematrix(1:s,2); % selected the s first value
of PerimeterMatrix

storagematrix(s,3)=s_MajorAxisLength;
MajorAxisLengthMatrix=storagematrix(1:s,3); % selected the s first
value of MajorAxisLengthMatrix

storagematrix(s,4)=s_MinorAxisLength;
MinorAxisLengthMatrix=storagematrix(1:s,4); % selected s first
value of MinorAxisLengthMatrix

storagematrix(s,5)=s_Eccentricity;
EccentricityMatrix=storagematrix(1:s,5); % selected the s first
value of EccentricityMatrix

storagematrix(s,6)=s_Orientation;
OrientationMatrix=storagematrix(1:s,6); % selected the s first
value of OrientationMatrix

```

```

    storagematrix(s,7)=s_ConvexArea;
    ConvexAreaMatrix=storagematrix(1:s,7); % selected the s first value
of ConvexAreaMatrix

    storagematrix(s,8)=s_FilledArea;
    FilledAreaMatrix=storagematrix(1:s,8); % selected the s first value
of FilledAreaMatrix

    storagematrix(s,9)=s_EulerNumber;
    EulerNumberMatrix=storagematrix(1:s,9); % selected the s first
value of EulerNumberMatrix

    storagematrix(s,10)=s_EquivDiameter;
    EquivDiameterMatrix=storagematrix(1:s,10); % selected the s first
value of EquivDiameterMatrix

    storagematrix(s,11)=s_Solidity;
    SolidityMatrix=storagematrix(1:s,11); % selected the s first value
of SolidityMatrix

    storagematrix(s,12)=s_Extent;
    ExtentMatrix=storagematrix(1:s,12); % selected the s first value of
ExtentMatrix

    storagematrix(s,13)=X;
    XMatrix=storagematrix(1:s,13); % selected the s first value of
XMatrix

    storagematrix(s,14)=Y;
    YMatrix=storagematrix(1:s,14); % selected the s first value of
YMatrix

%     matrix_index=matrix_index+1; % repeat the loop till getting all
lesions in an image

%     Compute mean of differents parameters
Area=mean(AreaMatrix);
Perimeter=mean(PerimeterMatrix);
MajorAxisLength=mean(MajorAxisLengthMatrix);
MinorAxisLength=mean(MinorAxisLengthMatrix);
Eccentricity=mean(EccentricityMatrix);
Orientation=mean(OrientationMatrix);
ConvexArea=mean(ConvexAreaMatrix);
FilledArea=mean(FilledAreaMatrix);
EulerNumber=mean(EulerNumberMatrix);
EquivDiameter=mean(EquivDiameterMatrix);
Solidity=mean(SolidityMatrix);
Extent=mean(ExtentMatrix);
AbscPosition=mean(XMatrix);
OrdinPosition=mean(YMatrix);
StdAbscPosition=std(XMatrix);
StdOrdinPosition=std(YMatrix);
numberOflesions=nnz(XMatrix); % Total number of lesions in an image
(it's exclude all lesions out of our region of interest). nnz:count all non
zero values in a matrix

```

```

end

% Wavelet Features extraction

[TrainingContrast,TrainingCorrelation,TrainingEnergy,TrainingHomogeneity,Tr
ainingMean,TrainingStandard_Deviation,TrainingEntropy,TrainingRMS,TrainingV
ariance,TrainingSmoothness,TrainingKurtosis,TrainingSkewness]=WaveletFeatur
esExtraction(img2);

% Create matrix

% Training matrix

TrainingMatrix( train_index,1)=Area;
TrainingMatrix( train_index,2)=Perimeter;
TrainingMatrix( train_index,3)=MajorAxisLength;
TrainingMatrix( train_index,4)=MinorAxisLength;
TrainingMatrix( train_index,5)=Eccentricity;
TrainingMatrix( train_index,6)=EulerNumber;
TrainingMatrix( train_index,7)=Solidity;
TrainingMatrix( train_index,8)=Extent;
TrainingMatrix( train_index,9)=AbscPosition;
TrainingMatrix( train_index,10)=OrdinPosition;
TrainingMatrix( train_index,11)=StdAbscPosition;
TrainingMatrix( train_index,12)=StdOrdinPosition;
TrainingMatrix( train_index,13)=numberOflesions;
TrainingMatrix( train_index,14)=TrainingContrast;
TrainingMatrix( train_index,15)=TrainingCorrelation;
TrainingMatrix( train_index,16)=TrainingEnergy;
TrainingMatrix( train_index,17)=TrainingHomogeneity;
TrainingMatrix( train_index,18)=TrainingMean;
TrainingMatrix( train_index,19)=TrainingStandard_Deviation;
TrainingMatrix( train_index,20)=TrainingEntropy;
TrainingMatrix( train_index,21)=TrainingRMS;
TrainingMatrix( train_index,22)=TrainingVariance;
TrainingMatrix( train_index,23)=TrainingSmoothness;
TrainingMatrix( train_index,24)=TrainingKurtosis;
TrainingMatrix( train_index,25)=TrainingSkewness;

train_index=train_index +1;

TrainingMatrix(isnan(TrainingMatrix))==0; % All Nan value will be
equal to zero

% close all;

end

% Create training table

TrainArea = TrainingMatrix(:,1);
TrainPerimeter = TrainingMatrix(:,2);
TrainMajorAxisLength = TrainingMatrix(:,3);
TrainMinorAxisLength = TrainingMatrix(:,4);
TrainEccentricity = TrainingMatrix(:,5);

```

