



**MARMARA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



# **DERİN ÖĞRENME İLE BİYOSİNYAL SINIFLANDIRMA VE HASTALIK TAHMİNİ**

**MURAT ALAN**

**YÜKSEK LİSANS TEZİ**

Mekatronik Mühendisliği  
Anabilim Dalı  
Mekatronik Mühendisliği Programı

**DANIŞMAN**

Prof. Dr. Mustafa Caner AKÜNER

**EŞ-DANIŞMAN**

Doç. Dr. Alper KEPEZ

**İSTANBUL, 2020**



**MARMARA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



# **DERİN ÖĞRENME İLE BİYOSİNYAL SINIFLANDIRMA VE HASTALIK TAHMİNİ**

**MURAT ALAN**  
(523516020)

**YÜKSEK LİSANS TEZİ**  
Mekatronik Mühendisliği  
Anabilim Dalı  
Mekatronik Mühendisliği Programı

**DANIŞMAN**  
Prof. Dr. Mustafa Caner AKÜNER

**EŞ-DANIŞMAN**  
Doç. Dr. Alper KEPEZ

**İSTANBUL, 2020**

# MARMARA ÜNİVERSİTESİ

## FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Murat ALAN'ın "Derin Öğrenme ile Biyosinyal Sınıflandırma ve Hastalık Tahmini" başlıklı tez çalışması, \_\_\_/\_\_\_/\_\_\_ tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

### Jüri Üyeleri

(Danışman)

..... (İMZA).....

(Eş-Danışman)

..... (İMZA).....

(Üye)

..... (İMZA).....

(Üye)

..... (İMZA).....

(Üye)

..... (İMZA).....

### ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun \_\_\_/\_\_\_/\_\_\_ tarih ve \_\_\_ sayılı kararı ile \_\_\_\_\_  
Yüksek Lisans derecesi alması onanmıştır.

**Fen Bilimleri Enstitüsü Müdürü**  
**Prof. Dr. Bülent EKİCİ**

## ÖNSÖZ

Her gün daha da ileriye giden, her gün biraz daha gelişen teknolojiyle birlikte günlük hayatımıza biraz daha yerleşen yapay zekâ, günümüzde ekonomiden mühendisliğe, dil biliminden sağlığa kadar birçok alanda kullanılmaktadır. Sağlık alanında yapılan gelişmeler hastalıklara tanı konulmasında ve tedavisinde büyük öneme sahiptir. Hem uzman hekimlerin işlerini kolaylaştırmakta hem de hastalara erken tanı fırsatı vermektedir. Bu çalışmada, kalp hastalıklarının erken tanısını kolaylaştırmak için gerekli çalışmalar yapılmıştır. Bu çalışmalara göre, eğitilen yapay zekâ sağlıklı bir kalp ile hasta bir kalbi ayırt edebilmektedir.

Bu tez çalışmasında hiçbir desteği esirgemeyen değerli danışman hocam Prof. Dr. Mustafa Caner AKÜNER'e, yine desteklerini esirgemeyen diğer danışman hocam Doç. Dr. Alper KEPEZ'e, yapay zekâ konusundaki eğitimleri için değerli iş arkadaşım Vahit Feryat'a en içten teşekkürlerimi sunarım.

Ayrıca, destekleriyle her zaman yanımda olan arkadaşlarım ve sevdiklerime, beni varlığı ile mutlu eden ve bu süreçte manevi desteklerini esirgemeyen sevgilime, hayatım boyunca beni tüm maneviyatı ile her konuda destekleyen ve hiçbir emeği benden esirgemeyen babama, anneme ve kardeşime en içten duygularıyla teşekkür ederim.

**Ekim, 2020**

**MURAT ALAN**

# İÇİNDEKİLER

## SAYFA

ÖNSÖZ .....	i
İÇİNDEKİLER.....	ii
ÖZET .....	iv
ABSTRACT .....	v
SEMBOLLER.....	vi
KISALTMALAR.....	vii
ŞEKİL LİSTESİ .....	viii
TABLO LİSTESİ.....	ix
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KALP ANATOMİSİ .....</b>	<b>5</b>
2.1. Kalbin Yapısı .....	5
2.2. Kalp Sesleri .....	8
<b>3. YAPAY ZEKA .....</b>	<b>11</b>
3.1. Makine Öğrenmesi .....	12
3.2. Yapay Sinir Ağları.....	13
3.2.1. Aktivasyon fonksiyonları .....	16
3.2.1.1. Linear (doğrusal) fonksiyon.....	16
3.2.1.2. Sigmoid fonksiyonu .....	16
3.2.1.3. Tanh (hiperbolik tanjant) fonksiyonu .....	17
3.2.1.4. ReLU (rectified linear unit) fonksiyonu .....	17
3.2.1.5. Sızıntı ReLU (leaky ReLU) fonksiyonu .....	18
3.2.1.6. Softmax fonksiyonu .....	19
3.3. Derin Öğrenme.....	19
3.3.1. Başlıca derin öğrenme mimarileri .....	19
3.3.1.1. Konvolüsyonel sinir ağları (Convolutional neural network).....	19
3.3.1.2. Tekrarlayan sinir ağı (Recurrent neural network) .....	20
3.3.2. Konvolüsyonel sinir ağında kullanılan katmanlar.....	21
<b>4. KALP SESLERİNİN DERİN ÖĞRENME İLE SINIFLANDIRILMASI .....</b>	<b>23</b>
4.1. Veri Setinin Hazırlanması .....	23
4.2. Veri Setinin Etiketlenmesi .....	24
4.3. Konfigürasyon Dosyalarının Ayarlanması.....	25
4.4. Veri Setinin Eğitilmesi.....	27

4.5. Eğitim Sonucunun Gerçekleştirilmesi.....	29
4.5. Derin Öğrenme Analizi Sonuçları.....	32
<b>5. BULGULAR VE TARTIŞMA .....</b>	<b>39</b>
<b>6. SONUÇ .....</b>	<b>43</b>
<b>KAYNAKÇA.....</b>	<b>45</b>
<b>ÖZGEÇMİŞ</b>	



# ÖZET

## DERİN ÖĞRENME İLE BİYOSİNYAL SINIFLANDIRMA VE HASTALIK TAHMİNİ

Günümüzde yapay zekâ her alanda gelişmektedir. Bu alanlardan birisi de sağlık alanıdır. Teknolojik gelişmeler ile birlikte yapay zekâ, sağlık alanında da kendine önemli bir yer edinmeye başlamıştır. Bu sayede hastalıkların erken evrede tespit edilmesi ve tanı konulması hususunda yapay zekâ uzman hekimlerin ve hastaların işlerini kolaylaştırmaktadır.

Kalp hastalıklarının erken tanısına yardımcı olabilmesi için bu çalışmada yapay zekâ kullanılmıştır. Yapay zekâ birçok katmandan meydana gelen yapay sinir ağlarından oluşur. Bu yapay sinir ağları aldığı girdileri çıkışlara dönüştürerek, girdilerin sınıflarını tahmin etmektedir. Bu çalışmada konvolüsyonel sinir ağları kullanılmıştır. Bu yapay sinir ağları görüntü içerisindeki nesnelere sınıflandırmada kullanılır.

Konvolüsyonel yapay sinir ağları, ilk önce sağlıklı ve hasta kalp sesleri ile eğitilmiş ve bu eğitimin sonucu test edilmiştir. Yapılan testler sonucunda eğitilen yapay zekânın sağlıklı ve hasta kalp seslerini ayırt ettiği görülmüştür.

# **ABSTRACT**

## **BIOSIGNAL CLASSIFICATION AND DISEASE PREDICTION WITH DEEP LEARNING**

Artificial intelligence is developing in every field. One of these areas is health. Along with the technological developments, artificial intelligence has started to take an important place in the field of health. In this way, artificial intelligence for the detection and diagnosis of diseases at an early stage facilitates the work of doctors and patients.

Artificial intelligence was used in this study for the early diagnosis of heart diseases. Artificial intelligence consists of artificial neural networks that consist of many layers. These artificial neural networks predict the classes of inputs by converting the inputs to outputs. Convolutional neural networks were used in this study. This artificial neural networks are used to classify the objects in the image.

Convolutional artificial neural networks were first trained with healthy and sick heart sounds and the result of this training was tested. As a result of the tests carried out, it has been seen that trained artificial intelligence distinguishes between healthy and sick heart sounds.



## SEMBOLLER

- $y$  : Yapay sinir hücresinin çıktısı  
 $w_i$  : Yapay sinir hücresine verilen ağırlık  
 $x_i$  : Yapay sinir hücresine verilen girdi  
 $b$  : Bias (sapma) değeri



## **KISALTMALAR**

- GPU : Graphical Process Unit  
CPU : Central Process Unit  
TÜİK : Türkiye İstatistik Kurumu  
FKG : Fonokardiyogram  
EKG : Elektrokardiyogram  
ReLU : Rectified Linear Unit  
USB : Universal Serial Bus  
IoU : Intersection over Union  
mAP : Mean Average Precision  
AP : Average Precision  
TP : True Positive  
FP : False Positive  
TN : True Negative  
FN : False Negative

## ŞEKİL LİSTESİ

	<b>SAYFA</b>
Şekil 2.1. Kalbin Yapısı.....	5
Şekil 2.2. Kalp Kapakçıkları.....	6
Şekil 2.3. EKG Dalgası.....	7
Şekil 2.4. Oskültasyon Bölgeleri .....	8
Şekil 2.5. Kalp Sesleri (S1 ve S2).....	9
Şekil 3.1. Yapay Zekâ ve Alt Dalları.....	12
Şekil 3.2. Yapay Sinir Hücresi .....	14
Şekil 3.3. Sigmoid Fonksiyonu.....	16
Şekil 3.4. Hiperbolik Tanjant Fonksiyonu.....	17
Şekil 3.5. ReLU Fonksiyonu .....	18
Şekil 3.6. Sınımlı ReLU Fonksiyonu .....	18
Şekil 3.7. Konvolüsyonel Sinir Ağı.....	20
Şekil 3.8. Tekrarlayan Sinir Ağı .....	20
Şekil 4.1. Python ile Çizdirilen Bir Kalp Sesi Verisi .....	24
Şekil 4.2. Değiştirilmesi Gereken Sınıf Sayısı Satırı.....	25
Şekil 4.3. Değiştirilmesi Gereken Filtre Sayısı Satırı.....	26
Şekil 4.4. Sağlıklı Kalp Sesi .....	29
Şekil 4.5. Yolov3 Algoritması Çıktısının Kamera Görüntüsü Üzerinde Gösterilmesi.....	30
Şekil 4.6. Hasta Kalp Sesi.....	30
Şekil 4.7. Yolov3 Algoritması Çıktısının Kamera Görüntüsü Üzerinde Gösterilmesi.....	31
Şekil 4.8. Yolov4 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi.....	31
Şekil 4.9. Yolov4 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi.....	31
Şekil 4.10. Yolov5 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi.....	32
Şekil 4.11. Yolov5 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi.....	32
Şekil 4.12. Yolov3 Algoritması AP <sub>50</sub> Değerleri .....	34
Şekil 4.13. Yolov4 Algoritması AP <sub>50</sub> Değerleri .....	34
Şekil 4.14. Yolov5 Algoritması AP <sub>50</sub> Değerleri .....	34
Şekil 4.15. Yolov3 Algoritması AP <sub>75</sub> Değerleri .....	35
Şekil 4.16. Yolov4 Algoritması AP <sub>75</sub> Değerleri .....	36
Şekil 4.17. Yolov5 Algoritması AP <sub>75</sub> Değerleri .....	36
Şekil 5.1. Yolov3 Algoritmasında Hata Değerinin Eğitimle Sıfıra Yaklaşması .....	39
Şekil 5.2. Yolov4 Algoritmasında Hata Değerinin Eğitimle Sıfıra Yaklaşması .....	40

## TABLO LİSTESİ

	<b>SAYFA</b>
<b>Tablo 4.1.</b> Algoritmaların AP ve mAP Değerleri (IoU=0.50).....	35
<b>Tablo 4.2.</b> Algoritmaların AP ve mAP Değerleri (IoU=0.75).....	36
<b>Tablo 4.3.</b> Algoritmaların AP ve mAP Değerleri .....	37



# 1. GİRİŞ

Günümüzde teknolojinin insan hayatının bir parçası olması yaşam standartlarını arttırmaktadır. Artık her insanın muhakkak sahip olduğu bir kişisel teknolojik cihazı bulunmaktadır. Bu cihazlar her türlü alanda karşımıza çıkmaktadır. Örneğin, bir üniversite öğrencisi sunumlarını dizüstü bilgisayarında hazırlarken, bir şirketin üst düzey yöneticisi sunumunu tablet üzerinden yapmaktadır. Yine bir fotoğraf sanatçısı tam profesyonel bir dijital fotoğraf makinesi ile çekim yaparken, amatör bir sanatçı cebinde bulunan akıllı telefon ile fotoğraflar çekebilmektedir. Teknolojik cihazları, bu tarzda daha birçok örnekle anlatmak mümkündür. Her alanda karşımıza çıkan teknolojik cihazlar medikal alanda da kullanılmaktadır. Örneğin, bir diyabet hastası birey, kan şekeri düzeyini belirlemek için kan şekeri ölçüm cihazını kullanmaktadır. Yine tansiyon hastası bir birey de dijital tansiyon aleti ile ölçümlerini yapabilmektedir.

Medikal alandaki teknolojik gelişmeler hekimlerin görevlerini bir hayli kolaylaştırmaktadır. Piyasada hekimlerin kullanımına sunulması için geliştirilen ve birçok hekim tarafından kullanılan elektronik stetoskoplar bulunmaktadır. Hekimler bu stetoskoplar sayesinde dinledikleri vücut seslerini görselleştirebilir, hatta daha sonra da kullanabilmek için kayıt altına alabilirler. Kayıt aldıkları vücut sesi verilerini, isterlerse yerel hafıza birimlerinde, isterlerse bulut ortamında saklayabilirler.

Teknolojik gelişmelerin hayatımıza bu denli entegre olması, her alanda bireylerin işlerini kolaylaştırmakta ve yaşam standartlarını arttırmaktadır. Özellikle medikal alanda yapılan teknolojik gelişmeler, hastalıkların erken evre teşhisinde önemli rol oynamaktadır. Hastanelerde kullanılan radyolojik görüntüleme cihazlarından, ameliyatlarda kullanılan robotlara kadar medikal alanın her noktasında artık teknolojik cihazlardan yardım alınmaktadır. Klinik ortamda hastalıkların analizinde ve yorumlama sürecinde, her ne kadar teknolojik cihazlar kullanılsa da bu analiz ve yorumlama uzman hekimin bilgi ve mesleki tecrübesine dayanmaktadır. Günümüzde teknolojik cihazların elverişli olduğu ölçüde geliştirilen algoritmalar ve yazılımlar sayesinde, klinik olmayan ortamlarda da bazı durumlarda hastaların kendi bireysel verilerini takip etmelerine olanak sağlanmıştır. Bu tip cihazların geliştirilmesi, teknolojik ilerlemenin bir sonucudur.

Geliştirilen yeni uygulama ve algoritmaların, verilerin takibine olan katkısı yadsınamaz bir gerçektir. Fakat artık teknolojik gelişmeler, bu tip cihazlara takip yeteneğinin yanı sıra yorumlama yeteneği de kazandırmıştır. Yorumlama yeteneğinin kazandırılması için yapay zekâ algoritmaları ve yapay sinir ağları kullanılmaktadır.

Önceleri donanımsal kısıtlamalar nedeniyle pek fazla kullanılmayan yapay zekâ algoritmaları ve yapay sinir ağları, günümüzde teknolojinin birçok alanında kullanılmaya başlanmıştır. Bu kullanımın artmasındaki en büyük etken GPU (Graphics Processing Unit) tarafında yaşanan teknolojik gelişmeler olmuştur. CPU'lar (Central Processing Unit) yüksek hıza sahip işlem birimleri barındırırken, GPU'lar yavaş hıza sahip binlerce işlem birimi barındırır. Bu sayede GPU'lar, yapılan işlemlerde CPU'lara göre daha performanslı çalışmaktadır. GPU tarafında yaşanan gelişmelerle yapay sinir ağları, sığ ağlardan derin ağlara geçiş yapmıştır. Bu sayede derin öğrenme algoritmaları geliştirilmeye başlanmıştır. Derin öğrenme algoritmaları görüntü işlemeden, doğal dil işlemeye, aktivite tanımadan, medikal uygulamalara kadar geniş bir yelpazede başarı ile kullanılmaktadır [1].

Medikal birçok alanda kullanılan derin öğrenme, bu tez içerisinde kalp rahatsızlıklarının erken safhada belirlenmesine yardımcı olmak amacı ile hasta ve sağlıklı kalp seslerini ayırt etmede kullanılacaktır. Bu konuda yapılan bir çalışmaya göre karşılaşılan en büyük zorluğun çok düşük sinyal kalitesi sergileyen kalp sesleriyle başa çıkabilecek doğru ve sağlam algoritmalar oluşturulmasının gerekliliği olarak ortaya çıkmıştır [2].

TÜİK'in 2017 yılındaki istatistik verilerine göre Türkiye'deki dolaşım sistemi kaynaklı ölümlerin %39,7'sine iskemik kalp hastalığı, %23'üne ise diğer kalp hastalıklarının neden olduğu açıklanmıştır [3]. Dünya Sağlık Örgütü'nün yaptığı araştırmaya göre dünyada kalp hastalıklarından ölenlerin sayısı toplam hastalıklardan ölenlerin sayısının %29'unu oluşturur. Bu nedenle kalp hastalıklarının ön teşhisi önemli hale gelmiştir. Klinik ortamda kalp sesinin analiz ve yorumla süreci tamamen hekimin bilgi ve mesleki tecrübesine dayanmaktadır. Fakat günümüzde yapay zekâ algoritmalarının ve telefon uygulamalarının gelişmesi, klinik olmayan ortamlarda insanların kendi kalp sesi ile ilgili analiz sonucunu otomatik olarak takip etmesini mümkün hale getirmiştir. Bu durumda erken kalp hastalığı teşhisi için bu tip gelişmeler kritik bir öneme sahiptir [4]. Bu konuda yapılan bir çalışmada, kalp seslerini dinlemenin ve yorumlamanın hataya

açık doğası nedeni ile otomatik sınıflandırmanın önemine vurgu yapılmıştır [5]. Yine bu konuda yapılan başka bir çalışmada, kalp seslerinin dinlenmesi ile yapılan muayenelerin klasik bir prosedür olup, sesleri dinlemenin ve yorumlamanın kazanılması gereken bir deneyim olduğundan bahsedilmiştir. Bu nedenle akıllı sınıflandırmanın, tıbbi pratisyenlere deneyim geliştirmede yarar sağlayacağına değinilmiştir [6]. Ayrıca fonokardiyogram (FKG) sinyali bulunan verilerin bilgisayar ortamında derin öğrenme ile sınıflandırılması hekimlerin de işini kolaylaştıracaktır.

Bu çalışma ile özellikle kardiyoloji hekimlerine erken tanı koymada yardımcı olacak bir yazılım geliştirmek amaçlanmıştır. Bu yazılım sayesinde hekimler kalp seslerini hem dinleyecek hem de mobil cihazları veya bilgisayarları gibi kişisel teknolojik cihazlarla bile kalp seslerinin analizini yapabileceklerdir.

Bundan sonra yer alan bölümde canlılar için hayati öneme sahip organ olan kalbin görevleri ve yapısı hakkında bilgiler verilmiştir. Bu bilgiler arasında, kalpte bulunan iki ana bölüm olan karıncık ve kulakçıkların görevleri açıklanmıştır. Yine bu bilgiler arasında, karıncık ve kulakçıkların kasılması ile oluşan S1 ve S2 sesleri ile ilgili bilgilerle, bu seslerin göğüste hangi bölgelerden stetoskop ile dinlendiğine kadar bilgiler verilmiştir.

Üçüncü bölümde yapay zekaya giriş yapılmış ve yapay zekanın alt dalları hakkında bilgiler verilmiştir. Yapay zekanın bir alt dalı makine öğrenmesidir. Makine öğrenmesinin bir alt dalı ise yapay sinir ağlarıdır. Birçok katmandan oluşan yapay sinir ağlarının oluşturduğu yapıya ise derin öğrenme denmektedir. Üçüncü bölüm içerisinde yine bu çalışmada da kullanılan, derin öğrenme içerisinde bulunan konvolüsyonel sinir ağları hakkında bilgiler verilmiştir.

Bir sonraki bölümde çalışmanın hazırlıkları, yöntemi ve deneyleri hakkında bilgiler bulunmaktadır. Bu bölümde, kullanılan veri setinin özellikleri, nasıl ayrıştırıldığı, düzenlendiği ve etiketlendiği hakkında detaylı bilgiler verilmiştir. Yine bu bölümde, kullanılan derin öğrenme algoritmaları ve modellerinin veri seti ile nasıl eğitildiği hakkında da detaylı bilgiler bulunmaktadır. Veri seti ile eğitilen algoritmaların ve modellerin, test verileri ile nasıl test edildiği ve sonuçlarının neler olduğu yine bu bölümde yer almaktadır.

Beşinci bölümde, elde edilen bulgular ve sonuçlar tartışılmıştır. Ayrıca bu konuda yapılan diğer çalışmalarla ilgili veriler, bu çalışmada veri seti ile eğitilen algoritma ve modellerin test sonuçları ile karşılaştırılmıştır. Bu karşılaştırmaların bulguları ve sonuçları, sonuç bölümünde karara bağlanmıştır.

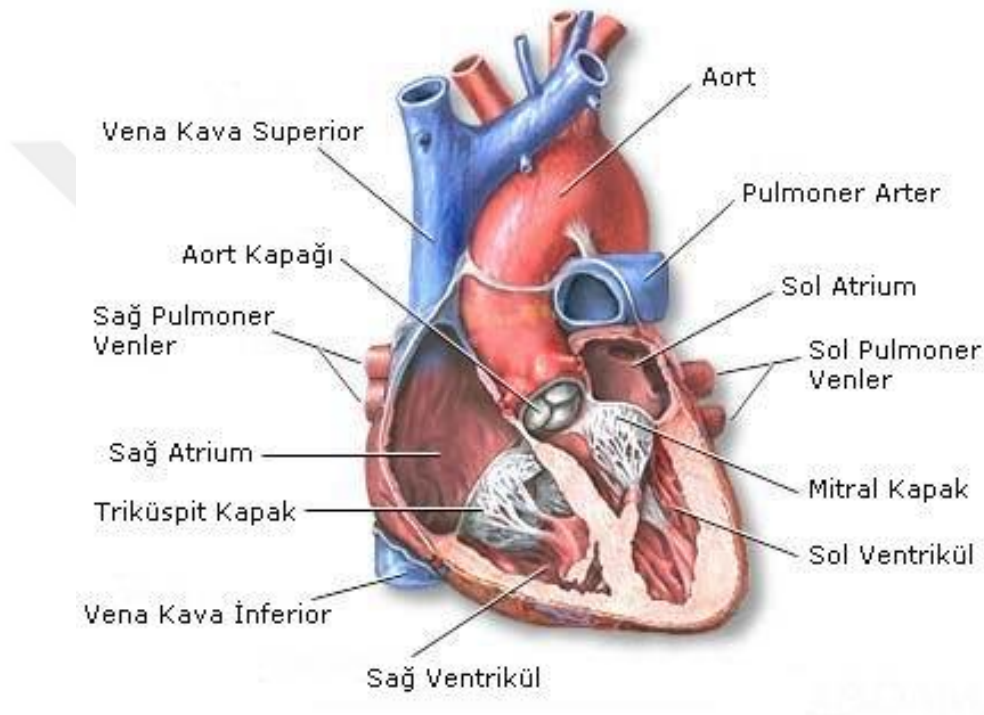




## 2. KALP ANATOMİSİ

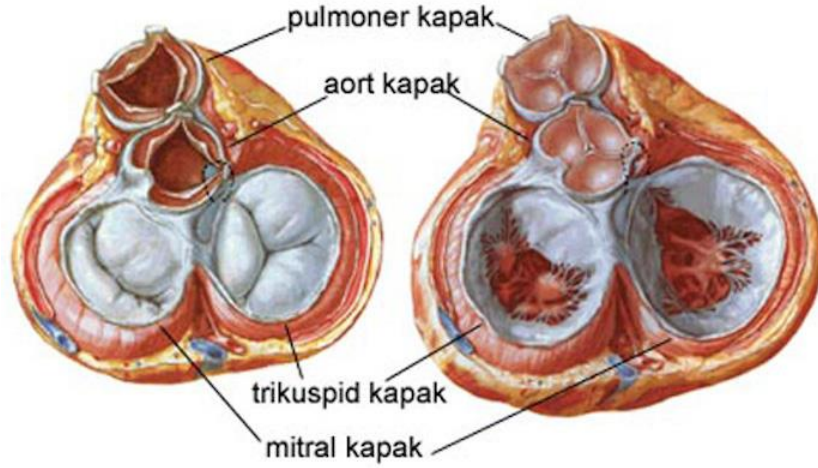
### 2.1. Kalbin Yapısı

Kalp insan vücudu için en önemli organlardan biridir. Göğüs kafesinin hemen arkasında, iki akciğeri ayıran orta boşlukta ve sol tarafta bulunur. Kardiyovasküler sistemle vücut için gerekli olan kanı pompalar.



**Şekil 2.1.** Kalbin Yapısı [<https://ahmetalpman.com/genel-anatomi/>]

Kalp yapısal olarak iki ana bölüme ayrılır. Bunlarda üst tarafta bulunana atriyum (kulakçık), alt tarafta bulunana ise ventrikül (karıncık) adı verilmektedir. Karıncığın uç kısmına apeks adı verilir. Karıncık ve kulakçık birbirlerinden kapakçıklarla ayrılırlar. Kapakçıklara atriyoventriküler kapak denir. Sağ karıncık ile kulakçık arasındaki kapağa triküspid kapakçık, sol karıncık ile sol kulakçık arasındaki kapakçığa ise mitral kapakçık denir. Karıncıklardan atardamlara açılan kapakçıklara ise semilunar (yarımay) kapakçığı denir. Sağ karıncık ile akciğer atardamarı arasındaki kapakçığa pulmoner kapakçık, sol karıncık ile aort atardamarı arasındaki kapakçığa aort kapakçığı denir [7].



**Şekil 2.2.** Kalp Kapakçıkları [http://www.cengizbolcal.com.tr/trikuspid-kapak-hastaliklari/]

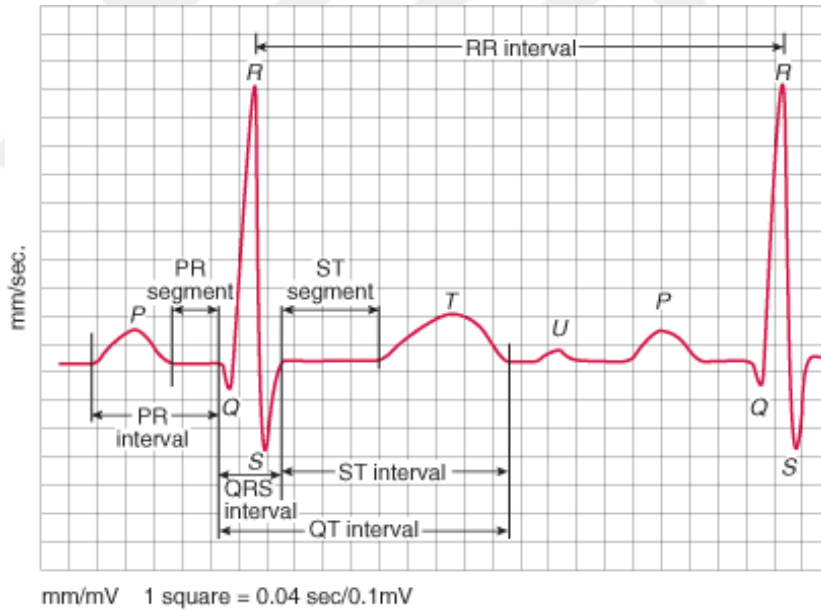
Kulakçıkların alt yüzeylerinde atriyoventriküler kapakçıklar bulunur. Üst taraflarında ise toplardamarlar yer alır. Sol kulakçıkta akciğer toplardamarları (pulmoner venler), sağ kulakçıkta ise alt (inferior) ve üst (superior) ana toplardamarları (vena cava) bulunur.

Kalp, dıştan içe doğru epikart, miyokart ve endokart adlı üç katmandan oluşur. Kalbin kas yapısına miyokart adı verilir [8]. Çizgili kastan oluşmuş olsa da kalbin kasılması isteğimiz dışında gerçekleşmektedir. Kalbin en dış tarafı perikart adı verilen zarla çevrilidir. Perikart, kalp ve büyük damar köklerini çepeçevre saran çift duvarlı kese şeklinde bir zardır [9].

Kalp, aorttan çıkan koroner atar damarlar ile beslenir. Kirli kan yine koroner toplar damarlar ile sağ kulakçığa taşınır. Kalpte bulunan sağ atriyum kan ile dolduğunda, kasılarak kanı triküspid kapakçıktan sağ ventriküle gönderir. Sağ ventriküle geçen kan, buranın da kasılmasıyla pulmoner kapakçıktan geçerek pulmoner arter ile akciğere gönderilir. Akciğerlerde kan, arteriol adı verilen kılcal damarlar sayesinde alveol denen hava keseciklerine ulaşır ve bu hava keseciklerinde temizlendikten sonra pulmoner venler adı verilen toplardamarla kalpte bulunan sol atriyuma iletilir. Sol atriyum kasılarak, kanı mitral kapakçık ile sol ventriküle gönderir. Aort kapakçığının açılması, mitral kapakçığın kapanması ile kan aort atardamarı ile vücuda gönderilir. Bu dolaşım döngüsü sayesinde kalp, kirli kanı akciğerlere göndererek temizletir, akciğerden dönen temiz kanı ise vücuda pompalar [10].

Kalp kasılma işlemleri için kendi iç sinir sistemini kullanır. Tüm bu kasılmalar sonucunda kalp kaslarında elektriksel işaretler oluşur ve bu işaretler elektrokardiyografi aracılığı ile ölçülebilir. Elektrokardiyografi aracılığıyla ölçülen, kalbin elektriksel aktivitesi sonucu oluşan grafiğe de elektrokardiyogram (EKG) denir [11]. EKG, kalbin çalışma evreleri ile ilgili veriler sunmaktadır ve birçok kalp hastalığının tanısında kullanılmaktadır.

Elektrokardiyogram, P, Q, R, S, T, U adı verilen dalgalardan ve bazal çizgi (baseline) adıyla bilinen çizgiden oluşur. P dalgası, atriyal depolarizasyonu simgeler. Pozitif veya negatif defleksiyondur. Sağ atriyumdaki sinüs nodundan başlatılır. Genliği 0,25 mV, genişliği de 0,12 saniyedir. P dalgasının başından, QRS kompleksinin başladığı zamana kadar geçen süreye PR dalgası denir. Uyarının atriyumlara yayılmasını ve atrioventriküler kavşaktan geçmesi için gereken süredir. Yetişkin bireylerde PR dalgası, 0,12 ile 0,20 saniyedir.



**Şekil 2.3.** EKG Dalgası [<https://www.acilci.net/p-dalgasi/>]

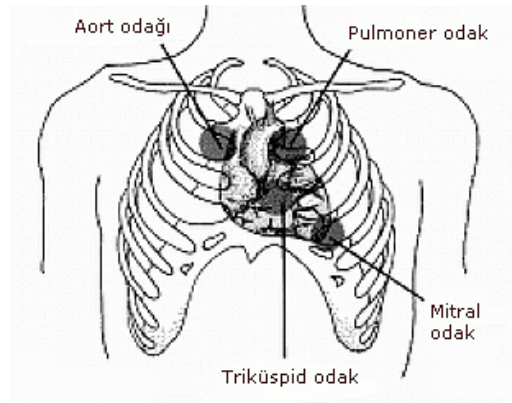
Uyarının ventriküller boyunca ilerlediğini gösteren dalgaya QRS kompleksi denir. QRS kompleksi, Q adı verilen bir negatif dalga, R adı verilen bir pozitif dalga ve S adı verilen ikinci bir negatif dalgadan oluşur. QRS kompleksi, ventriküler depolarizasyonu gösterir ve 0,1 saniye veya daha azdır. QRS kompleksinin bittiği nokta ile T dalgasının başlangıcı arasına, ST segmenti denir. Ventriküler repolarizasyonun başlamasına denk

gelir. Normal ST segmenti genellikle izoelektriktir. T dalgası ise ventriküler repolarizasyonun bir kısmını yansıtır. Asimetrik bir biçime sahiptir. Tepe noktası bitimine daha yakındır. QT aralığı, ventriküler repolarizasyon için geçen süreyi göstermektedir. QT aralığı için normal değerler kalp hızına bağlıdır. U dalgası, T dalgasından sonra görülen yuvarlak ve küçük bir defleksiyondur. Yönü T dalgasının yönü ile aynıdır [12].

## 2.2. Kalp Sesleri

Kalp sesleri, kanın kardiyovasküler sistemde dolaşımı ve kalp kapakçıklarının hareketleri sırasında meydana gelen seslerdir. Kalp seslerinin dinlenerek incelenmesi ve analiz edilmesine fonokardiyografi, fonokardiyografi sonucu oluşan ses dalgalarının grafiğine ise fonokardiyogram adı verilir [13].

Kalp sesleri, stetoskopun göğüse yerleştirilmesi ile dinlenir. Bu işleme oskültasyon denir. Seslerin analizi için göğüs dört bölgeye ayrılmış ve bu bölgelere oskültasyon bölgeleri adı verilmiştir. Bu bölgeler dinlenerek kalpteki sorunlu bölgenin nispeten tespiti yapılır. Oskültasyon bölgeleri kalp kapakçıklarının bulunduğu bölgeler değil, kalp kapakçıklarından geçen kanın akışı yönündeki bölgelerdir. Bu bölgeler, kendi baş harfleri ile gösterilir ve “Z” şeklinde bir sıralama oluşturur.

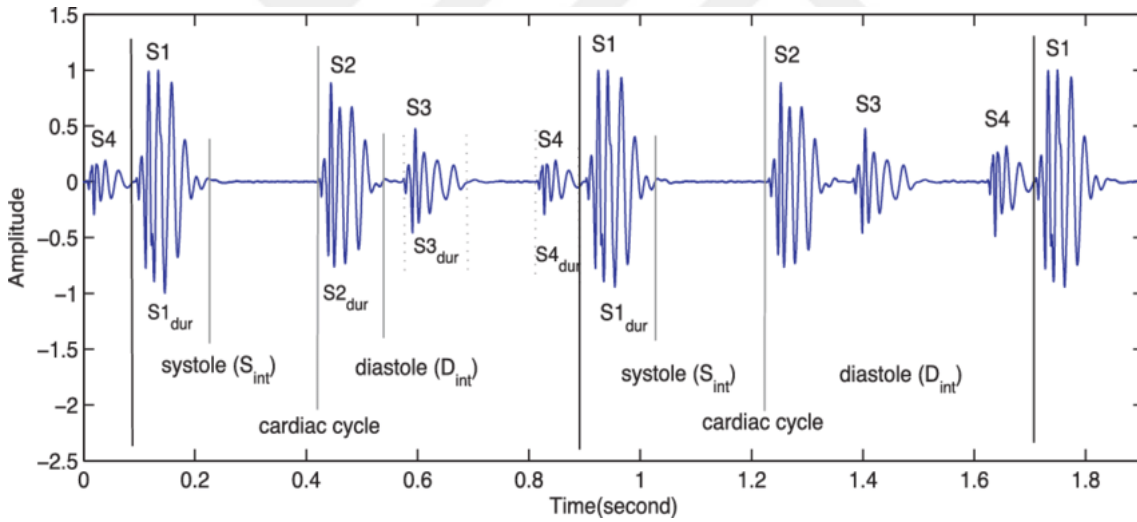


**Şekil 2.4.** Oskültasyon Bölgeleri [<https://ahmetalpman.com/kalp-sesleri/>]

Şekil 2.4'te aort odağı olarak gösterilen bölge, 2. interkostal aralık ile sternumun sağ kenarının kesiştiği bölgedir ve aortik odak olarak adlandırılır. Aort kapakçığı ile ilgili seslerin analizi yapılır. Pulmoner odak olarak gösterilen bölge, 2. interkostal aralık ile sternumun sol kenarının kesiştiği bölgedir ve pulmoner odak olarak adlandırılır.

Pulmoner kapakçıklarla ilgili ses analizleri yapılır. Triküspid odak olarak gösterilen bölge, sağ 4. kostanın sternumun sol kenarıyla kesiştiği bölgedir ve triküspid odak olarak adlandırılır. Mitral odak ile gösterilen bölge de sol midklaviküler hattın 5. interkostal aralığı kestiği bölgedir ve mitral odak olarak adlandırılır. Triküspid odak dinlenerek triküspid kapakçık ile ilgili, mitral odak dinlenerek mitral kapakçık ile ilgili ses analizleri yapılır [14]. Kalp seslerinin yoğunluğu, kalitesi ve frekansı önemlidir. Oskültasyon sırasında hekim kalp seslerinin bu özelliklerine dikkat eder.

Kalp sesleri, S1 ve S2 adında iki ses dalgasından oluşur. S1, birinci kalp sesi ve S2 ise ikinci kalp sesi olarak adlandırılır. Bunlara ek olarak gelen diğer klik seslerine ve tok seslere ek sesler denir. S3, S2'den sonra işitilen düşük perdeli tok bir sestir. S3 kalp sesi ergenliğe kadar dinlenebilir fakat daha sonraki yaşlarda konjestif kalp yetmezliği bulgusudur. S1 ve S2 seslerinin aralığı önemlidir. S1 ve S2 arasındaki kısa aralık sistol aralığıdır. S2 ve bir sonraki sesin S1 dalgası arası da diyastole denk gelir.



**Şekil 2.5.** Kalp Sesleri (S1, S2, S3 ve S4) [15]

S1 dalgası, atriyoventriküler kapakların kapanması ve kapak dokularının gerilmesi sonucu oluşur. Sistolün başlangıcını belirler. En iyi apekte duyulur. Mitral (M1) ve triküspid (T1) parçalarında oluşur. Bu iki parça arasındaki süre kısa olduğu için birinci ses genellikle tek olarak duyulur. S2 dalgası, aort ve pulmoner kapakların kapanması sonucu oluşur. Diyastol başlangıcını belirler. Ana parça olan aortik component (A2), en iyi aort odağında ve pulmoner component (P2) ise en iyi pulmoner odakta duyulur [12].



### 3. YAPAY ZEKA

Çeşitli görevleri ve işlemleri bilgisayar kontrollü bir makine veya robotun, zeki canlılara benzer bir şekilde yerine getirme kabiliyetine yapay zekâ denir. Yapay zekâ kavramı hızla gelişmekte ve kendi kendine öğrenebilen makineler veya robotlar artık insan hayatına entegre olabilmektedir. Hatta insandan bağımsız, kendi kendini eğitebilen bir yapay zekaya doğru yeni araştırmalar ve geliştirmeler yapılmaktadır.

Yapay zekâ teriminin doğuşu 1940'lı ve 1950'li yıllara dayanmaktadır. Bu dönemde yaşamış bazı bilim insanları, “yapay bir beyin yaratma” fikrini ortaya atmışlardır. Yine bu bilim adamları 1956'da bu alanda akademik bir disiplinin kurucuları olmuşlardır. Bu dönemde Alan Turing, düşünen bir makinenin yapılıp yapılmayacağını araştırmış ve Turing Testi'ni (kendi deyimiyle taklit oyunu – imitation game) yayınlamıştır [16]. Bu teste göre bir makine, bir insanla sohbet ediyormuş gibi davranmalıydı. Teste göre bir adet sorgulayıcı, bir adet bilgisayar ve bir adet gönüllü insan sohbet etmeli, sorgulayıcının sorduğu sorulara göre aldığı cevapların, bilgisayardan mı yoksa gönüllü insandan mı geldiğini bulması gerekmektedir. Sorgulayıcı, bilgisayar ve gönüllü insanın oluşturduğu test grubunun, birbirleriyle iletişimi tamamen yazı aracılığıyla olmalıydı. Bu cevaplar ya bir ekran da gösterilmeli ya da bir klavye sistemi ile yazılmalıydı. Eğer sorgulayıcı, sorduğu sorular neticesinde makine ve insanı ayırt edemiyorsa, makine testi geçmiş sayılırdı. Bu test ve yapılan araştırmalar sonucunda, yapay zekanın temelleri atılmış oldu.

Yine aynı yıllarda Prof. Dr. Cahit Arf'in de “Makineler Düşünebilir mi? ve Nasıl Düşünebilir?” adlı makalesinde, o zamanın şartlarına göre makinelerin oda kadar bile büyük olanları, çeşitlilik bakımından insan beyninden oldukça düşük olduğuna, insan beyninin kendini eğittiği fakat makinelerin yapıldığı gibi kaldığına dikkat çekmiştir. Bununla beraber, kendi kendini eğiten makineleri de tasarlamının mümkün olduğuna vurgu yapmıştır [17].



Şekil 3.1. Yapay Zekâ ve Alt Dalları

### 3.1. Makine Öğrenmesi

Makine öğrenmesi, yapay zekânın bir alt kümesi olup, bir makine veya bilgisayara büyük veri setlerine dayalı karar verme yetisi kazandırmaktır [18]. Shai Shalev-Shwartz ve Shai Ben-David yazdıkları kitapta, makine öğrenimini verilerdeki anlamlı kalıpların otomatik olarak algılanması şeklinde tanımlamaktadırlar [19]. Nils J. Nilsson ve arkadaşları tarafından yazılan kitapta da makine öğrenmesinin genellikle yapay zekâ ile ilişkili görevleri gerçekleştiren sistemlerdeki değişiklikleri ifade ettiğini ve bu ilişkili görevlerin tanıma, teşhis, planlama, robot kontrolü ve tahmin gibi görevler olduğunu belirtmişlerdir [20]. Michael Negnevitsky'nin kaleme aldığı kitapta, makine öğreniminin genel olarak bilgisayarların deneyimlerden, örneklerden ve benzetmelerden öğrenmesini sağlayan mekanizmaları içerdiğinden bahsetmektedir [21]. Makine öğrenmesi genellikle, veri madenciliği, istatistik, örüntü tanıma ve olasılık işlemlerinde kullanılır. Makine öğrenmesi yaklaşımı, hazırlanan algoritmalarla çok fazla veri setinin analiz edilmesine olanak sağlar. Bu büyük veri setlerinin analiz edildiği algoritmalar, makinelerin öğrenimini ve daha kararlı çıkarım yapmasını sağlamış olur.



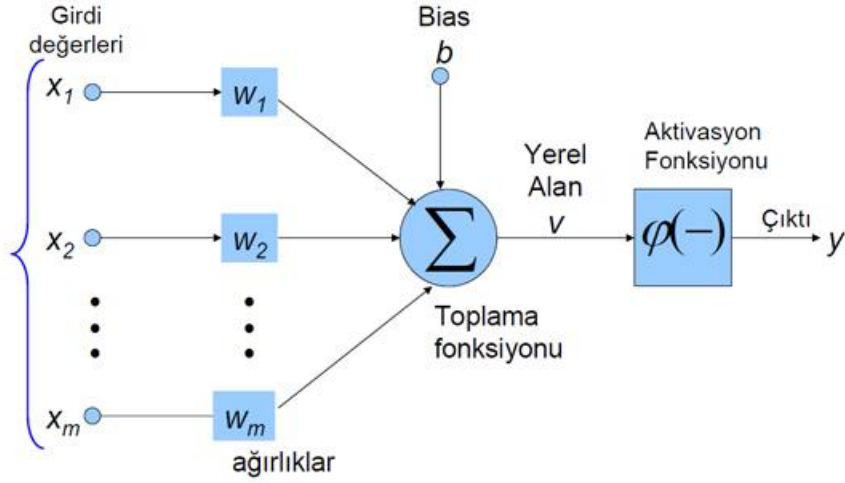
Makine öğrenmesi ve yapay zekâ arasındaki fark ise, yapay zekanın insan gibi düşünen ve davranan bir teknoloji, makine öğrenmesinin ise veri setleri ile daha kararlı çıkarım yapmasını sağlayan bir teknoloji olmasıdır [22].

### 3.2. Yapay Sinir Ağları

Beyinde bulunan sinirlerin çalışmalarını taklit eden algoritmalara, yapay sinir ağları adı verilmektedir [23]. Bu ağlar, birbirlerine sinaps benzeri yapılarla bağlı yapay nöronlardan oluşmaktadır. Her bir nöron arasındaki bağlantı, o nörondan diğer nörona bilgi taşır. Bilgiyi alan nöron gerekli işlemi yaparak, bu bilgiyi bir sonraki nörona aktarır.

Yapay sinir ağları öğrenmeye meyillidirler. Kendi performanslarını arttırmak için deneyimleri kullanabilirler. Yeteri kadar örnek ile eğitilirse, henüz karşılaşmadıkları örnekleri genelleyebilirler. İnsan konuşmasındaki kelimeleri belirleyebilir, havalimanındaki patlayıcıları tespit edebilirler. Yapay sinir ağları, insanların gözünden kaçan olayları yakalayabilirler. Örneğin, Chase Manhattan Bank, çalınan kredi kartlarının kullanımıyla ilgili bir dizi bilgiyi incelemek için bir sinir ağı kullandı ve en şüpheli satışların 40 ila 80 dolar arasında değişen kadın ayakkabıları için olduğunu keşfetti [21].

Yapay sinir ağları, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere en basit haliyle üç katmandan oluşur. Veri setleri giriş katmanından yapay sinir ağlarına girilir. Bu veriler, gizli katmanda bulunan nöronlara giderken belli başlı işlemlere tabi tutulur. Gizli katmandan geçen veriler çıkış katmanına gelerek sonuca ulaştırılır [23]. Tek katmanlı yapay sinir ağlarına “perceptron”, çok katmanlı yapay sinir ağlarına ise “multi layer perceptron” adı verilir [24].



**Şekil 3.2.** Yapay Sinir Hücresi [https://www.psikolojik.gen.tr/images/yapay-sinir-aglari.jpg]

Yapay sinir hücresi Denklem 3.1’i kullanır. Girdileri ( $x$ ), ağırlıklar ( $w$ ) ile çarpıp, bias ( $b$ –sapma değeri) ile toplayarak toplam net çıkışını üretir. Net çıkışını daha sonra aktivasyon fonksiyonuna girdi olarak verip,  $y$  çıkışı elde edilir [25]. Ağırlıklar 0 ile 1 arasında belirlenir.

$$y = f(\sum_{i=0}^n (w_i * x_i) + b) \quad (3.1)$$

Yapay sinir hücrelerinin görevi girişleri Denklem 3.1’e göre işleme sokup diğer bir yapay sinir hücresine aktarmaktır.

Yapay sinir ağları tasarlanırken ağırlıklar ilk olarak rastgele üretilir. Daha sonra girdiler ile çarpılarak yapay sinir hücresinden çıkar. Her bir ağırlık, sinir hücresine giren girdinin önemini belirler. Bu değere eklenen bias, sonucu artırır veya azaltır. Buradan da çıkan sonuç aktivasyon fonksiyonuna verilir ve çıktı elde edilir. Bu aşama, daha öncede anlatıldığı gibi bir yapay sinir hücresinin yaptığı işlemler bütünüdür. Daha sonra elde edilen çıktı ile olması gereken hedef çıktı arasındaki farka bakılır. Bu fark ne kadar büyükse sistem o kadar kötü sonuç vermektedir. Bu fark maliyet (cost) fonksiyonu ve hata değerini verir. Maliyet fonksiyonu aşağıdaki denklem ile bulunur.

$$E_{total} = \sum \frac{1}{2} (hedeflenen - çıkış)^2 \quad (3.2)$$

Burada çıkan maliyet değerinin minimum olması istenir. Yani bu fonksiyonun yerel minimumunun yakınsanması gerekmektedir. Bu yakınsamanın da doğru iteratif şekilde

yapılması gerekir. Maliyet fonksiyonunun bulunmasına kadar gelen işlemler bütünü, 1 epoch olarak adlandırılır. Bu epoch sonrasında maliyet değeri büyük çıkarsa, hata geriye yayılmalıdır. Bu işlem için geri yayılım (backpropagation) kullanılır. Bu işlemin amacı, önceki adımlarda bulunan ağırlıkların güncellenerek, çıktının hedef çıktıya yaklaşması yani maliyet fonksiyonunu teorik değer olan sıfıra indirmektir. Bu işlemler yapılırken geçen süreye ise öğrenme denir. Yapay sinir ağları, bu işlemlerden geçerek kendini eğitir.

Geri yayılım işlemi sırasında, azalma (gradient descent) algoritması kullanılır. Bu algoritma yerel minimuma yakınsarken türev kullanır. Bu algoritmanın yakınsama yönü gradient doğrultusuna ters yönde hareket ettirilir. Burada, bu kurala geçmeden önce hatanın ağırlığa bağlı olduğunu bulmamız gerekir. Şekil 3.2’de gösterilen bir yapay sinir hücresinde, geri yayılım uygularsak, çıktı olarak aldığımız  $y$  değerine göre hesaplanan hatanın aktivasyon fonksiyonuna bağlı olduğu, aktivasyon fonksiyonunun da net çıktıya bağlı olduğu, net çıktısının da giriş ile ağırlığın çarpımına bağlı olduğu ortaya çıkmaktadır. Giriş değerimiz sabit olduğundan net çıktısı ağırlığa bağlıdır. Burada matematikte uygulanan zincir kuralı devreye girmektedir. Burada birbirine bağımlı üç adet çözülmesi gereken türev vardır. Bunlar Denklem 3.3’te gösterilmiştir.

$$\frac{\partial Hata}{\partial w} = \frac{\partial Hata}{\partial aktivasyon} * \frac{\partial aktivasyon}{\partial net} * \frac{\partial net}{\partial w} \quad (3.3)$$

Denklem 3.3’te de görüldüğü gibi üç adet türev çözülüp, çarpılarak ağırlığın çıkış hatasına etkisi bulunur. Daha sonra bulunan bu değer, ağırlığın yeni değere güncellenmesinde kullanılmaktadır.

$$w_{yeni} = w - \left( \text{öğrenme katsayısı} * \frac{\partial hata}{\partial w} \right) \quad (3.4)$$

Denklem 3.4’te yeni ağırlık değerinin denklemi verilmiştir. Ağırlığa göre hata değerinin türevi bulunarak öğrenme katsayısı (learning rate) ile çarpılır. Bulunan bu değer, ilk ağırlıktan çıkarılarak yeni ağırlık değeri bulunmuş olur. Yapay sinir ağlarının eğitimi boyunca buraya kadar anlatılan işlemler her epochda tekrarlanarak, tüm ağırlıkların son değerleri bulunur ve sistem kararlı bir hale getirilmiş olur.

### 3.2.1. Aktivasyon fonksiyonları

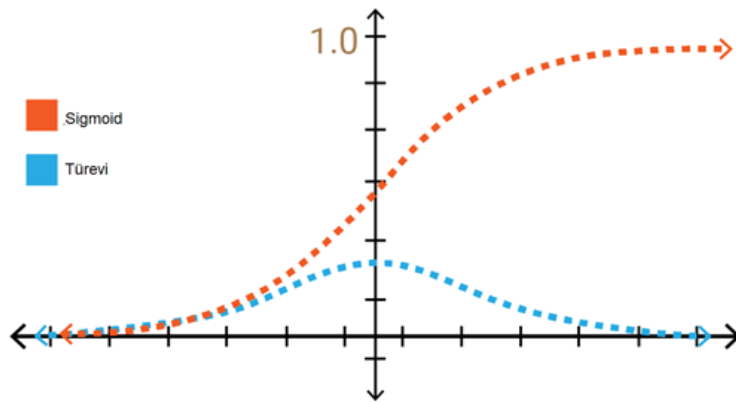
Yapay sinir ağı, yapay sinir hücrelerinin çıktısı aktive edilmediği zaman gerçek dünya üzerindeki lineer olmayan sonuçlara karşı yetersiz kalmaktadır. Eğer bu çıkışlara aktivasyon fonksiyonu uygulanmazsa, yapay sinir ağı lineer regresyon gibi davranacaktır. Yapay sinir ağlarının, lineer olmayan durumları da öğrenmesi istenmektedir. Bu nedenle yapay sinir hücrelerinin çıktılarına aktivasyon fonksiyonu uygulanmaktadır. Denklem 3.3'te de görüldüğü gibi geri yayılım işlemi sırasında fonksiyonların türevleri alındığından, aktivasyon fonksiyonunun türevinin alınabilir olması gerekmektedir.

#### 3.2.1.1. Linear (doğrusal) fonksiyon

Linear fonksiyon, türevi sabit olan bir fonksiyondur. Bu sebepten lineer fonksiyonlar, yapay sinir ağlarında kullanılmazlar. Aktivasyon fonksiyonu olarak, türevi sabit olan bir lineer fonksiyon kullanılırsa, yapay sinir ağı eğitimi gerçekleştirilmiyor demektir. Ayrıca, lineer fonksiyonun çıkışı doğrusaldır. Bu fonksiyonlar kullanıldığında, giriş ile çıkış arasında hep doğrusallık olacaktır. Böylelikle aradaki katmanlar, işlevini yitirecektir.

#### 3.2.1.2. Sigmoid fonksiyonu

Linear olmayan ve aynı zamanda da türevlenebilir bir fonksiyondur. Bunlara ilaveten, fonksiyon 0 ile +1 arasında çıktı üretir. Bu sebeplerle, yapay sinir ağlarında çok sık kullanılan bir aktivasyon fonksiyonudur.

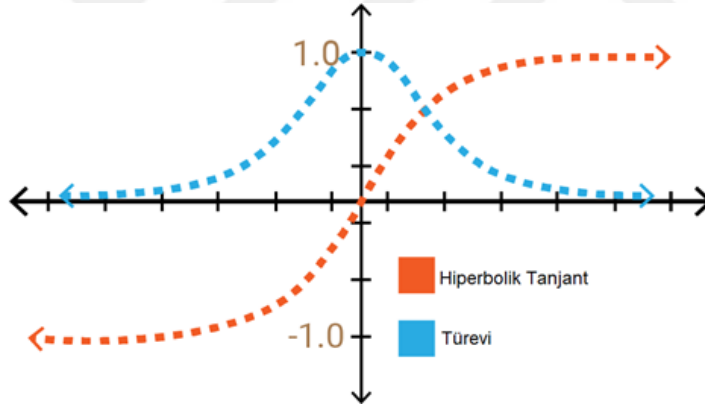


Şekil 3.3. Sigmoid Fonksiyonu [26]

Çok sık kullanılmasına karşın sigmoid fonksiyonunun, Şekil 3.3'te de görüleceği gibi uç noktalarında x değerlerine y değerlerinin tepkisi azdır ve bu bölgelerin türevleri sıfıra yakınsar. Bu nedenle, bu bölgelerde eğitim minimum düzeydedir. Hatta çıktı sıfır olursa, eğitim gerçekleşmemektedir. Bu nedenle, yapay sinir ağından istenilen performans alınamaz.

### 3.2.1.3. Tanh (hiperbolik tanjant) fonksiyonu

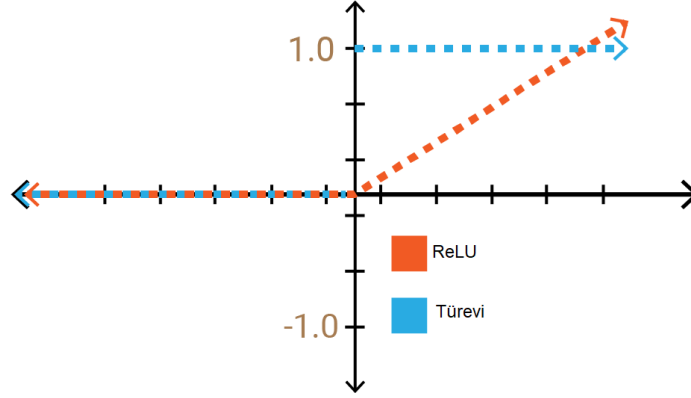
Sigmoid fonksiyonuna çok benzemektedir fakat fonksiyonun çıktı aralığı -1 ile +1 arasındadır. Sigmoid fonksiyonundan farkı, türevinin dik olması ve bu sebeple daha fazla değer alabilmesidir. Böylelikle daha hızlı öğrenme işlemi gerçekleşebilir. Fakat bu aktivasyon fonksiyonunda da uçlardaki türev sıfıra yakınsar. Sigmoid fonksiyonundaki gibi eğitim bu bölgelerde minimum düzeyde olacak, yapay sinir ağından istenilen performans alınamayacaktır.



Şekil 3.4. Hiperbolik Tanjant Fonksiyonu [26]

### 3.2.1.4. ReLU (rectified linear unit) fonksiyonu

Pozitif tarafta lineer fonksiyon ile aynı özellikleri gösteren ReLU fonksiyonunun çıktıları pozitif değerlerdir. Negatif tarafta çıktısı ise sıfırdır. Fonksiyonun çıktı aralığı 0 ile  $+\infty$  arasındadır. Bu nedenle, doğrultulmuş lineer fonksiyon olarak adlandırılır.

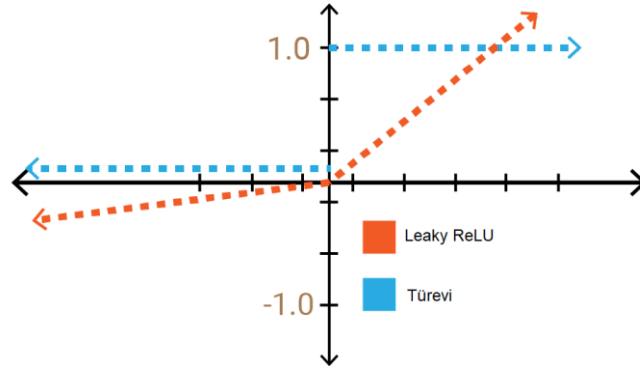


Şekil 3.5. ReLU Fonksiyonu [26]

ReLU fonksiyonu, doğrusal olmayan fonksiyon grubundadır. ReLU fonksiyonunun negatif tarafta sıfır değeri üretmesi, yapay sinir ağının eğitimindeki iş yükünü hafifletmektedir. Sigmoid ve tanjant fonksiyonlarına göre hesaplama yükü azdır. Bu yüzden, çok katmanlı yapay sinir ağlarında daha çok tercih edilirler. Ancak ReLU fonksiyonunun bu özelliği, negatif tarafta türevinin sıfır olmasına yol açmaktadır. Bu fonksiyonun negatif kısmında eğitim gerçekleşmez. Bu sebeple, Sızıntı ReLU (Leaky ReLU) fonksiyonu kullanılır.

### 3.2.1.5. Sızıntı ReLU (leaky ReLU) fonksiyonu

Bu fonksiyonda, sızıntı değeri negatif bölgede 0,01 olarak verilir. Böylelikle, bu fonksiyonun tanım aralığı  $-\infty$  ile  $+\infty$  arasındadır. Bu sızıntı nedeniyle fonksiyonun negatif tarafı, sıfıra değil  $-\infty$ 'a gider. Bu bölgede sıfırken gerçekleşmeyen eğitim, sızıntı sayesinde negatif bölgede de devam etmiş olur.



Şekil 3.6. Sızıntı ReLU Fonksiyonu [26]

### **3.2.1.6. Softmax fonksiyonu**

Sigmoid fonksiyonu ile benzer yapıdadır. Sınıflayıcı olarak kullanıldığında yüksek performans göstermektedir. Bu yüzden genellikle, ikiden fazla sınıflandırma kullanılan yapay sinir ağlarının çıkış katmanlarında kullanılır.

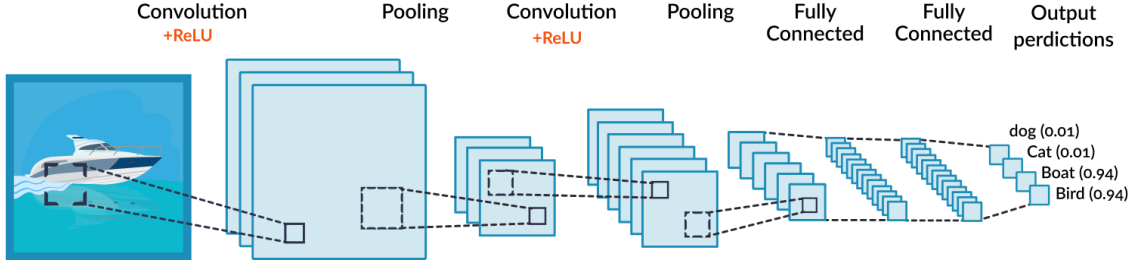
## **3.3. Derin Öğrenme**

Derin öğrenme, beyinde bulunan sinir ağlarının yapısı ve fonksiyonu gibi çalışan yapay sinir ağları algoritmaları kullanan bir makine öğrenmesi alt kümesidir [27]. Makine öğrenmesinin alt kümesi olması sebebiyle, derin öğrenme de büyük veri setlerinin analizini yaparak, kararlı çıkarımlar yapabilmektedir. Derin öğrenmenin makine öğrenmesinden farkı, makine öğrenmesi tek katmanda işlem yaparken, derin öğrenme birçok katmanda aynı anda işlem yapabilmektedir. Makine öğrenmesi insan zekasına ihtiyaç duyarken, derin öğrenme insan zekasına ihtiyaç duymamaktadır. Örneğin, sınıflandırma yapılacak iki nesnenin özelliklerini makine öğrenmesine parametre olarak girmek gerekirken, derin öğrenmeye verilen iki nesnenin farkını gösteren parametreleri derin öğrenme kendisi çıkarabilmektedir.

### **3.3.1. Başlıca derin öğrenme mimarileri**

#### **3.3.1.1. Konvolüsyonel sinir ağları (Convolutional neural network)**

Bu derin öğrenme mimarisi, sıklıkla görüntü işleme ve analizi için kullanılmaktadır. Çok katmanlı yapay sinir ağı modelidir. Bu model, filtreler ile işlem yapan bir modeldir. Kullanılan filtreler, görüntü içerisinde sınıflandırılacak olan nesnenin özneliğini çıkarmaktadır. Filtreler farklı değerler ve boyutlarda ayarlanarak, baskın olmayan özneliklerinde bulunmasını sağlar. Konvolüsyonel sinir ağı içerisinde çok sayıda katman bulunur. Bu katmanlar, konvolüsyonel katman, havuzlama (pooling) katmanı, aktivasyon katmanı, sınıflandırıcı katman gibi katmanlardır ve bu katmanlardan çok sayıda bulunur. Her katman, kendi işlemini yerine getirerek sonuçlar sınıflandırıcı katmandan çıktı olarak verilir [28].



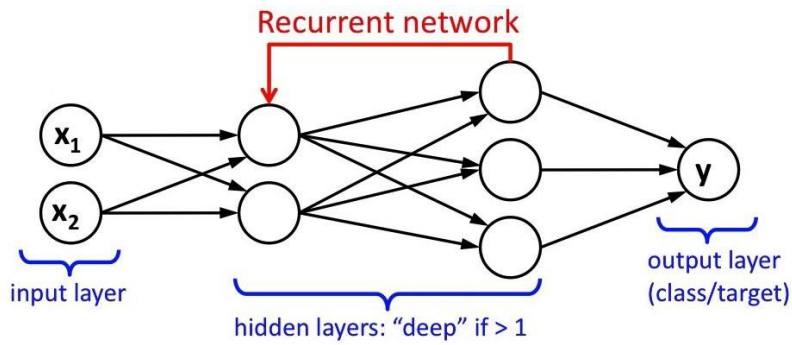
**Şekil 3.7.** Konvolüsyonel Sinir Ağı [<https://missinglink.ai/wp-content/uploads/2019/07/A-Convolutional-Neural-Network.png>]

Konvolüsyonel sinir ağı, en çok kullanılan yapay sinir ağı mimarisidir. Tespit ve sınıflandırma için kullanılır.

### 3.3.1.2. Tekrarlayan sinir ağı (Recurrent neural network)

Tekrarlayan sinir ağları, bir katmana giren girdinin o katmanda işlemini yaparak çıkışı alır. Daha sonra bu çıkışı kaydeder ve girişi bu çıktı ile geri besler. Çıkışları kaydetmesi nedeniyle, dahili bir hafızasından söz edilebilir. Konvolüsyonel sinir ağlarında çıkış, sadece o an işlenen girişe bağlı iken, tekrarlayan sinir ağlarında çıkış, geçmişteki girişlere de bağlıdır.

RNN'ler önceden hesaplanmış olan veriler hakkında bilgi sahibi olan ve bunu hafızada tutan bir yapıdır. Teoride, RNN'ler uzun vadeli bilgi tutabilmek için tasarlanmalarına karşın, uygulamada sadece birkaç adım önceki bilgiye kadar ulaşabilmektedir.



**Şekil 3.8.** Tekrarlayan Sinir Ağı [<https://images.deepai.org/glossary-terms/5142bdf22de24169bda948b961bfce99/download.jpeg>]

Tekrarlayan sinir ağları, sıklıkla dil çevirileri için kullanılsa da bir dizi işlemden sonra gelebilecek işlemi tahmin etmek için de kullanılır. Cümle tamamlama algoritmaları buna bir örnektir.



Gerçek dünyayla ilişkilendirdiğimizde RNN'lerin sıradan sinir ağlarına göre daha anlamlı olduğu gözükmemektedir. İnsanlar herhangi bir yazıyı okurken veya birini dinlerken düşüncelerini sıfırdan başlatmaz, devamlı olarak önceki bilgilerini anlamlandırarak devam eder. Örneğin; 5 kelimelik bir cümleyi RNN için yapılandırırsak, sinir ağı her kelime için bir katman oluşturur ve 5 katmanlı bir yinelemeli sinir ağı oluşmuş olur. Ağ her katmandaki bilgi birikimini sürdürür [29].

Tahmin edilmesi gereken çıktı ile o çıktı ile ilintili giriş arasındaki farkın açılması, tekrarlayan sinir ağları için dezavantajdır. Tekrarlayan sinir ağları, bu şekilde uzun vadeli bağlamları öğrenmekte güçlük çekerler. Bu dezavantajı ortadan kaldırmak için Uzun Kısa Vadeli Hafıza olarak adlandırılan farklı bir tekrarlayan sinir ağı kullanılır. Bu sinir ağı, uzun vadeli bağlamları öğrenebilir, istediği kaydedilen bilgiyi silebilir, yenisini kaydedebilir. Böylelikle giriş ve çıkış arasındaki bağlam farkı çok fazla olsa bile, yapay sinir ağı geçmiş bilgileri unutarak kendi içerisinde tekrardan öğrenebilir.

### **3.3.2. Konvolüsyonel sinir ağında kullanılan katmanlar**

Bu sinir ağındaki ilk katman, giriş katmanıdır. Görüntüler yapay sinir ağına bu katmandan verilir. Giriş olarak verilecek görüntü verileri ham şekildedir. Giriş olarak verilen görüntülerin çözünürlüğü ve boyutu, modelin başarımı için önemli bir etkidir.

Bu sinir ağında kullanılan bir diğer katman konvolüsyon katmanıdır. Bu katmanda, görüntü üzerindeki özellikleri çıkarmak için kullanılan filtreler bulunur. Bu filtrelerin boyutları ve değerleri değiştirilerek, özelliklere göre ayarlanabilir. Filtreler bu aşamada görüntüye konvolüsyon işlemi uygulayarak bir çıkış verisi oluşturur. Özellik haritası bu aşamada oluşturulur. Eğitim esnasında kullanılan filtrelerin katsayıları her adımda yinelenir. Böylelikle ağ üzerinde önemli bölgelerin yeri tespit edilmiş olur. 7x8 boyutlarında renkli bir görüntü olduğunu düşünürsek, giriş matrisimiz 7x8x3 olacaktır. Konvolüsyon işlemi için 3x3'lük bir filtre ile giriş görüntüsü üzerinde tüm kareler taranacak şekilde kaydırma işlemi yapılır. Filtre katsayıları her bir renk kanalındaki değerlerle çarpılıp bunların toplamı hesaplandığında aktivasyon haritası oluşur [30].

Bir diğer katman ise havuzlama katmanıdır. Bu katman, kendisine giriş olarak verilen katmandan gelen özellikleri kaybetmeden boyutunu küçültmek ve sinir ağının yükünü hafifletmek için uygulanır. Aktivasyon katmanında aktivasyon fonksiyonu bulunur. En

sık kullanılan aktivasyon fonksiyonu ReLU'dur [30]. Daha sonra tam bağı katman, dropout katmanı ve sınıflayıcı katman bulunur. Tam bağı katmanda, nesne özellikleri ile sınıflar eşlenir.

Yapay sinir ağlarında aşırı öğrenme sonucu ezber meydana gelir. Bu ezberin önüne geçmek ve tam bağı katmanı düzeltmek için dropout katmanı kullanılır [30]. Bu katman ezber yapmış düğümleri ortadan kaldırır. Sınıflandırma katmanında, sınıflandırma sayısı kadar çıkış üretilir. Yumuşatma (softmax) katmanı da sınıflandırma katmanı olarak kullanılır ve olasılıksal çıkış üretir.



## 4. KALP SESLERİNİN DERİN ÖĞRENME İLE SINIFLANDIRILMASI

Kalp sesleri sınıflandırılırken, açık kaynak kodlu olarak C ve CUDA ile Joseph Redmon tarafından yazılan Darknet kütüphanesini, yine açık kaynak olarak Google tarafından yazılan Tensorflow kütüphanesine, Python programlama dilinde uyarlayan ve Trieu Trinh'in yazdığı Darkflow isimli kütüphane kullanılmıştır [31][32]. Bu kütüphane kullanılırken Yolov3 isimli algoritmanın ağırlıkları ve modeli kullanılmıştır [33].

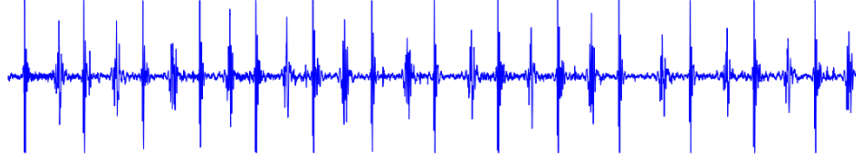
İkinci olarak ise Alexey Bochkovsky ve arkadaşlarının geliştirmiş olduğu Yolov4 isimli algoritma kullanılmıştır [34]. Üçüncü olarak kullanılan kaynak ise Glenn R. Jocher'ın kurduğu Ultralytics adlı şirketin açık kaynak olarak geliştirmiş olduğu Yolov5 isimli derin öğrenme algoritmasıdır [35].

İlk önce, kalp seslerinden oluşan veri setleri hazırlanmış, sağlıklı ve hasta kalp sesi olmak üzere iki sınıfa ayrılarak etiketlenmiştir. Etiketlenen bu veriler derin öğrenme algoritmalarında eğitilmiş ve daha sonra ise eğitim sonuçları test edilmiştir.

### 4.1. Veri Setinin Hazırlanması

Hasta ve sağlıklı kalp sesi olarak açık kaynak veri seti kullanılmıştır [36]. Bu veri setinde, farklı uzunluklarda, ses formatında 480 adet kalp sesi verisi bulunmaktadır. Bu verilerin 351 tanesi sağlıklı kalp sesi, 129 tanesi ise hasta kalp sesidir. 350 tane sağlıklı kalp sesinin, 120 tanesi, 129 tane hasta kalp sesinin ise 29 tanesi gürültü içermektedir.

Veri setini oluşturan kalp sesleri, WAV uzantılıdır. Fakat kullanılan yapay sinir ağlarını, görüntü girdisi ile eğitmek gerekmektedir. Bu nedenle her bir kalp sesinin, WAV formatından görüntü formatı olan PNG formatına çevrilmesi gerekmektedir. Bunun için Python programlama dili ile kod yazılmıştır. Bu kod, WAV dosyalarını okuyarak, PNG uzantılı olacak şekilde görüntü üzerine çizdirmektedir. Görüntü üzerine çizdirilen kalp sesi verileri, veri setinin bulunduğu klasöre kaydedilmiştir. Görüntü üzerine çizdirilmiş kalp sesi verisi örneği, Şekil 4.1'de gösterilmiştir.



**Şekil 4.1.** Python ile Çizdirilen Bir Kalp Sesi Verisi

Veri setindeki dosya isimlerinin eğitimde kullanılırken sıralı olması, etiketleme ve dosya takibinde büyük kolaylık sağlamaktadır.

Çok sayıda verinin bulunduğu algoritmalarda, isimlerin elle değiştirilmesi zaman almaktadır. Bu nedenle, veri setindeki dosyalar isimlendirilirken “Advanced Renamer” isimli hazır program kullanılmıştır [37]. Bu programda, dosya ismine tarih, saat gibi nelerin ekleneceği, hangi sayıdan başlayacağı, kaç basamaklı olacağı gibi ayarlamalar yapılabilmektedir.

480 adet kalp sesi verisi, bu şekilde yeniden isimlendirilmiştir. “001.png” isimli dosyadan “351.png” isimli dosyaya kadar sağlıklı kalp sesi, “352.png” isimli dosyadan “480.png” isimli dosyaya kadar ise hasta kalp sesleri veri setine eklenmiştir.

#### **4.2. Veri Setinin Etiketlenmesi**

Veri setleri, eğitim aşamasından önce etiketlenmelidir. Bu etiketler, konvolüsyonel sinir ağında kullanılmaktadır. Yapay sinir ağı, her bir veriyi eğitim için kullandığında etiketiyle birlikte öğrenir. Bu etiket, o verinin yapay sinir ağındaki sınıfını oluşturur.

Bu veri setinde, iki adet etiket kullanılacaktır. Bunlardan ilki sağlıklı, ikincisi hastadır. Bu etiketlemeyi Yolov3 algoritmasında kullanılan kütüphanenin okuyabilmesi için, “xml” uzantılı “annotation” dosyaları oluşturulması gerekmektedir. Bu dosyaların içerisinde, verinin bilgisayar içerisinde hangi klasörde olduğu, verinin dosya ismi ve tam uzantılı dosya yolu bulunmaktadır. Bunların haricinde etiketlenen verinin, çözünürlüğü, renk derinliği ve etiketlenen kısmın piksel koordinatları yer alır.

İkinci ve üçüncü olarak kullanılan Yolov4 ve Yolov5 algoritmaları için verileri, “txt” uzantısı ile etiketlemek gerekmektedir. Bu “txt” dosyasının içerisinde, görüntü içerisinde bulunan etiketlenen verinin sınıfının numarası ve koordinatları yer

almaktadır. Eğitim yapılırken “txt” uzantılı “annotation” dosyaları, veri seti ile birlikte eğitime verilir.

Çok sayıda veri bulunan veri setlerinde, işlem yükünü hafifletmek için hazır programlar kullanılır. Burada da veri setini etiketlemek için Python programlama dili ile Tzuta Lin tarafından yazılmış “LabelImg” adlı kod kullanılmaktadır [38]. Arayüzün sol tarafında bulunan butonlardan gerekli ayarlamalar yapılabilmektedir. Buton açıklamaları aşağıda verilmiştir. İkinci ve üçüncü olarak kullanılan açık kaynak kodlu Yolov4 ve Yolov5 derin öğrenme algoritmaları için “txt” uzantılı dosyalar gerekli olduğundan, “LabelImg” içerisinde “PascalVOC” tipi yerine “YOLO” tipi seçilmiştir.

Open Dir: Veri setinin bulunduğu klasörü açmak için kullanılır.

Change Save Dir: Kaydedilecek olan “xml” uzantılı “annotation” dosyalarının klasörünü belirlemek için kullanılır.

Next Image: Etiketlenecek bir sonraki resmi seçmek için kullanılır.

Prev Image: Bir önceki resme geçmek için kullanılır.

Save: “Annotation” dosyasını kaydetmek için kullanılır.

Create Rect Box: Etiketleme yapılacak alanı seçmek için kullanılır.

### 4.3. Konfigürasyon Dosyalarının Ayarlanması

Veri setinde bulunan etiket yani sınıf sayısına göre, kod klasörü içinde bulunan “cfg” uzantılı konfigürasyon dosyası ayarlanmalıdır. Konfigürasyon dosyası içinde bulunan [Region] katmanındaki (en son katman) sınıf sayısı, veri setindeki sınıf sayısı ile değiştirilmelidir.

```
[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11, 16.62,10.52
bias_match=1
classes=2
coords=4
num=5
softmax=1
jitter=.2
rescore=1
```

Şekil 4.2. Değiştirilmesi Gereken Sınıf Sayısı Satırı

Daha sonra bu sınıf sayısına göre, bir önceki katmanda bulunan filtre de değiştirilmelidir. Aşağıdaki denkleme göre filtre sayısı değiştirilir.

$$filters = num * (classes + 5) \quad (4.1)$$

Şekil 4.3'te gösterilen filtre sayısı, iki sınıflı bir veri için düzenlenmiştir.

```
[convolutional]
size=1
stride=1
pad=1
filters=35
activation=linear
```

**Şekil 4.3.** Değiştirilmesi Gereken Filtre Sayısı Satırı

Yolov4 algoritması için de “cfg” uzantılı konfigürasyon dosyası içinde aynı şekilde [Region] katmanındaki sınıf sayısı veri setindeki sınıf sayısı ile değiştirilmelidir. Yolov5 algoritmasında ise konfigürasyon dosyası ayarlarında, “train” ve “validation” verilerinin dosya yolları ile sınıf sayıları verilmiştir. En son satırda ise sınıfların etiketleri yer almaktadır.

Yolov3 algoritması için “normal” ve “sick” adındaki etiketler, kod klasörünün içinde bulunan “labels.txt” dosyasına eklenir. Etiketler, text dosyasının en başına yazılır. Etiketlerin sırası, veri setinde gruplanan sıraya göre yapılmalıdır. Burada kullanılan veri setinde, ilk grup sağlıklı, ikinci grup hasta kalp sesleri olduğu için text dosyasının ilk satırına normal, ikinci satırına ise sick etiketi yazılmıştır.

Yolov4 algoritması için ise “obj.names” ve “obj.data” isimli iki dosya oluşturulur. İlk dosya olan “obj.names” isimli dosyanın içerisinde, her bir satırda etiketlerin isimleri yer almaktadır. Diğer dosya olan “obj.data” dosyasının içeriğinde ise sınıf sayıları, “train” ve “validation” dosyalarının yolları ile etiket isimlerinin olduğu dosyanın yolu ve yedeklerin alınacağı dosyanın yolu bulunmaktadır.

Konfigürasyon dosyasının içinde, yapay sinir ağının katman tanımlamaları yapılmaktadır. Kalp seslerinin sınıflandırılması için kullanılan Yolov3 algoritmasında 9 adet konvolüsyon katmanı, 6 adet havuzlama (maxpool) katmanı kullanılmıştır. Konvolüsyonel katmanların her birinde aktivasyon fonksiyonu olarak, “leaky” kullanılmıştır. Bu katmanların haricinde, son katman olarak sınıflandırıcı katman ve bu

katmanda softmax aktivasyon fonksiyonu kullanılmıştır. Konvolüsyonel katmanlarda 3x3, maxpool katmanlarında 2x2 filtre kullanılmıştır.

Yolov4 algoritmasında ise 161 katman bulunmaktadır. Yolov4 Backbone, Neck ve Head olarak 3 ana bölümden oluşmaktadır. Bu bölümlerden ilki olan Backbone bölümünde, girdi olarak verilen veri setindeki özelliklerin çıkarılması için konvolüsyonel sinir ağlarının kabiliyetlerini geliştiren CSPNet kullanılmıştır [39]. CSPNet, daha az hesaplama ile daha zengin gradyan kombinasyonu elde edilmesi için geliştirilmiştir. Buradaki konsept, gradyan akışını bölerek farklı ağ yollarında akışını sağlamaktır. İkinci bölüm olan Neck bölümünde, farklı boyutta ve oranlardaki nesnelerin özelliklerinin çıkarılması için PANet ile görüntülerin özellik haritalarını tek seferde hesaplayan bir havuzlama (pooling) katmanı olan SPP kullanılmıştır [40][41]. Head kısmında ise Yolov3 algoritmasının katmanları kullanılmıştır.

Yolov5 derin öğrenme algoritmasında ise 407 katman bulunmaktadır. Bu katmanlar Backbone, Neck ve Head olarak 3 ana bölümden oluşmaktadır. Bu bölümlerden ilki olan Backbone bölümünde, Yolov4'te kullanılan CSPNet kullanılmıştır. İkinci bölüm olan Neck bölümünde, farklı boyutta ve oranlardaki nesnelerin özelliklerinin çıkarılması için PANet kullanılmıştır. PANet, nesnelerin segmentasyonu için kullanılmaktadır. Head kısmı ise Yolov3 ve Yolov4 ile aynıdır.

#### **4.4. Veri Setinin Eğitilmesi**

Veri seti eğitilirken, düzenlenen konfigürasyon dosyaları algoritmalara girdi olarak verilir. Konfigürasyon dosyaları ile yapay sinir ağları modellenmiş olur. Yolov3 algoritması eğitilirken, ilk ağırlık olarak Yolov3 algoritmasının “yolov3-tiny.weights” isimli ağırlığı kullanılmıştır. Yolov4 algoritmasını eğitirken ilk ağırlık olarak “yolov4.weights” isimli ağırlık kullanılmıştır. Yolov5 algoritması eğitilirken ise “yolov5x.pt” isimli ağırlık ilk ağırlık olarak kullanılmıştır.

Yapay sinir ağlarına bu ağırlıklar tanımlandıktan sonra, eğitim için gerekli olan veri seti ve veri setlerinin etiket dosyası olan “annotation” dosyaları, yapay sinir ağına girdi olarak verilir. Eğitim yapılırken, geri yayılım sırasında kullanılacak olan öğrenme oranı da koda verilir. Öğrenme oranı büyük olursa hedefe ulaşmak zorlaşmaktadır. Çok küçük olduğunda ise yakınsama uzun sürecektir. Bu nedenle test eğitimleri yapılmalı ve

optimum deęer belirlenmelidir. Yapılan testler sonucu Yolov3 algoritması için optimum deęerin 0,00001 olduęu belirlenmiştir. Yolov4 ve Yolov5 algoritmalarında ise bu deęer 0.001 olarak belirlenmiştir. Yolov3 algoritmasının eęitimleri ve testleri sırasında NVIDIA markasının GTX 1050TI modeli ekran kartı, Yolov4 ve Yolov5 algoritmalarının eęitimi ve testleri sırasında NVIDIA markasının GTX 2080TI modeli kullanılmıştır. Kullanılan kütüphanelere de bu ekran kartının %90 oranında kullanılacağı bilgisi girilmiştir. Bu deęer bilgisayarda bulunan kartın ve işlemcinin donanımsal özelliklerine göre belirlenmiştir. Son olarak, epoch sayısı ve kaç adımda bir checkpoint dosyası kaydedeceği belirlenmiştir. Yine epoch sayısı çok küçük olduğunda loss deęerinin sıfıra yakınsaması sıfırdan uzak olmakta ve modelin başarımlı oranı düşük olmaktadır. Çok büyük olduğunda ise sıfıra yakınsama belli bir epoch sayısından sonra azalmaktadır ve başarımlı oranındaki deęişiklik çok küçük oranlarda olmaktadır. Bu durumda eęitim sonlandırılmalıdır. Yapılan eęitim testlerinde optimum epoch sayısının 1000 olacağı görüşmüştür. Epoch sayısı olarak 1000 ve yine 1000 adımda da checkpoint dosyası kaydetmesi bilgisi algoritmalara verilmiştir. Model eęitilirken bütün veriler aynı anda eęitime katılmaz. Belli sayıda parçalara bölünerek verilerin tamamı tekrar tekrar eęitime katılırlar. Bu parçaların içerdiği veri sayısına “batch”, tüm parçaların eęitilmesi için gerekli tur sayısına ise “epoch” denir. Veri seti yığınları ekran kartında eęitileceğinden, 2'nin katları olarak seçilmelidir. Yapılan eęitim testlerinde 8, 32, 64 gibi batch deęerleri test edilmiş, optimum deęer olarak Yolov3 ve Yolov4 için 16 seçilmiştir. Yolov5 için ise donanımsal olarak hafızada problemler yaşandığı için 8 deęeri seçilmiştir.

Eęitim sırasında kaydedilen checkpoint dosyaları, geri yayılım işlemi sonucu elde edilen yeni aęırlık dosyalarını içermektedir. Maliyet fonksiyonu ve hata deęerinin en düşük olduğu andaki aęırlık deęerleri kullanılır.



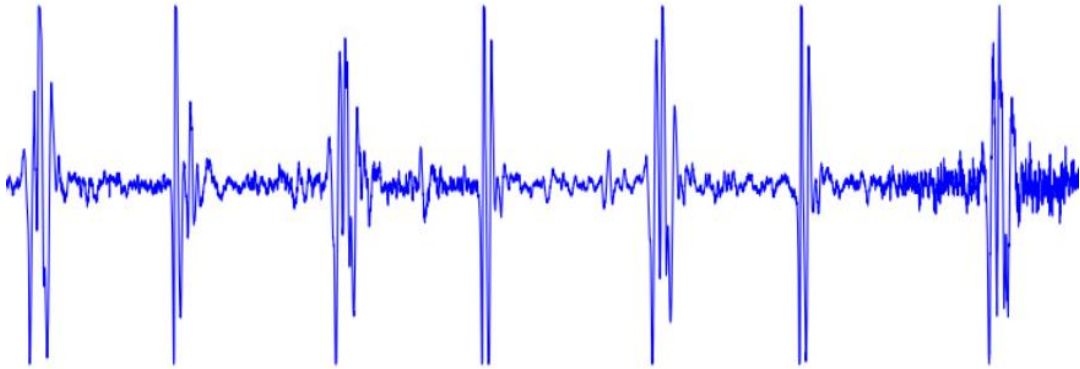
#### 4.5. Eğitim Sonucunun Gerçekleştirilmesi

Yolov3 algoritması için Python programlama dili ile yazılan kod ile eğitim sonucunda çıkan yeni ağırlık değerleri ile Yolov3 algoritması test edilmiştir. Test sırasında, eğitilen yapay sinir ağı görüntü üzerinden sınıflandırma yaptığından, bilgisayara takılan bir USB kamera kullanılmıştır.

Yazılan kod içerisindeki ayar kısmında, model olarak oluşturduğumuz konfigürasyon dosyası verilmiştir. Ağırlık kısmında, eğitimden çıkan en son ve kararlı checkpoint dosyası olan “8308” numaralı dosya kullanılmıştır. Eşik değeri olarak “0.1” seçilmiştir. Ekran kartı kullanım oranı ise “%90” olarak seçilmiştir.

Daha sonra kod içerisinde, USB kamera ile çözünürlük ayarları yapılmıştır. Kameradan alınan frame'ler, Darkflow kütüphanesi ile Yolov3 algoritmasına girdi olarak verilmiştir. Yapay sinir ağından geçen girdilerin tahmin çıktıları, “dictionary list” şeklinde geri alınmıştır. Bu çıktının içerisinde, tahmin edilen nesnenin, üst sol piksel koordinatları, alt sağ piksel koordinatları, etiketi ve tahmin (benzerlik) oranı vardır. Bu parametreler ayrıştırılarak açılan pencerede, kamera görüntüsü üzerine çizdirilir.

Şekil 4.4'te, Yolov3 algoritması testi için yapay sinir ağına verilen sağlıklı bir kalp sesinin görselleştirilmiş hali verilmiştir.



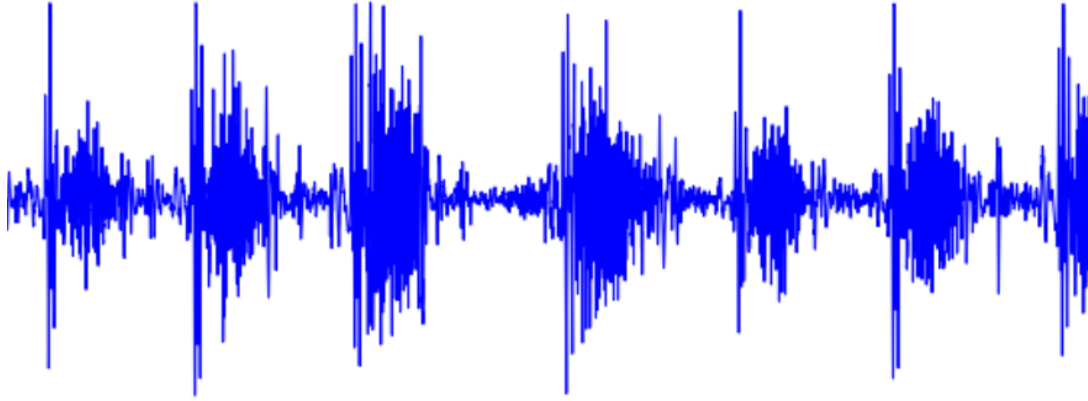
Şekil 4.4. Sağlıklı Kalp Sesi

Şekil 4.5'te ise, Yolov3 algoritması için eğitilmiş modelden alınan çıktının kamera görüntüsü üzerindeki hali görülmektedir.



**Şekil 4.5.** Yolov3 Algoritması Çıktısının Kamera Görüntüsü Üzerinde Gösterilmesi

Şekil 4.6’da Yolov3 algoritması testi için yapay sinir ağına verilen hasta bir kalp sesinin görselleştirilmiş hali verilmiştir.



**Şekil 4.6.** Hasta Kalp Sesi

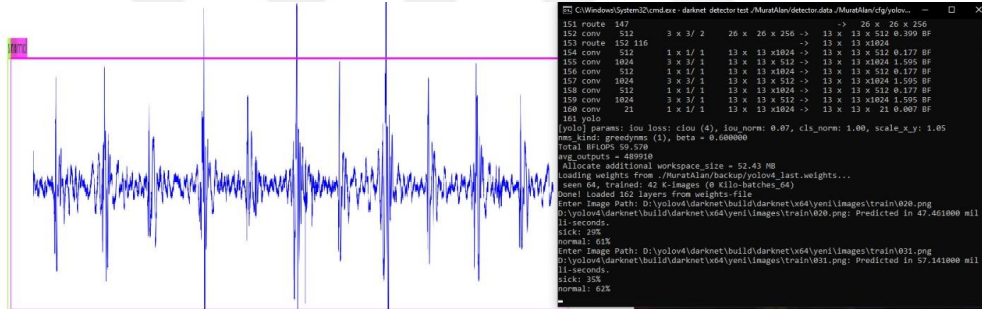
Şekil 4.7’de ise, Yolov3 algoritması için eğitilmiş modelden alınan çıktının kamera görüntüsü üzerindeki hali görülmektedir.



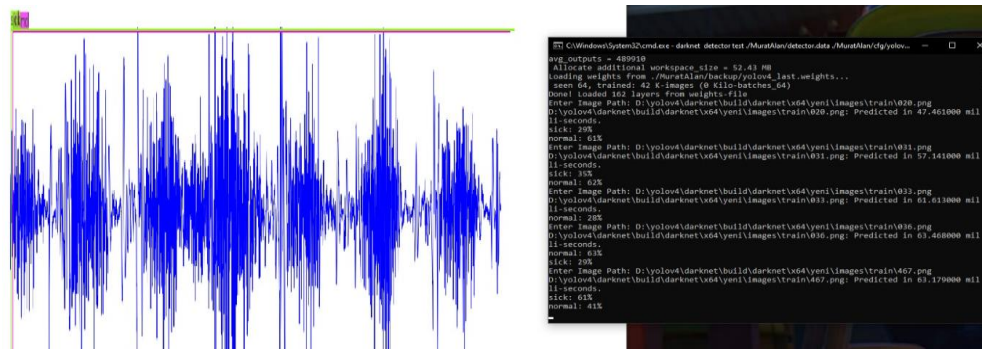
Şekil 4.7. Yolov3 Algoritması Çıktısının Kamera Görüntüsü Üzerinde Gösterilmesi

Yolov4 algoritması da Yolov3 algoritmasında kullanılan yöntemlerle test edilmiştir.

Şekil 4.8’de ve Şekil 4.9’da Yolov4 algoritmasının test sonuçları verilmiştir.

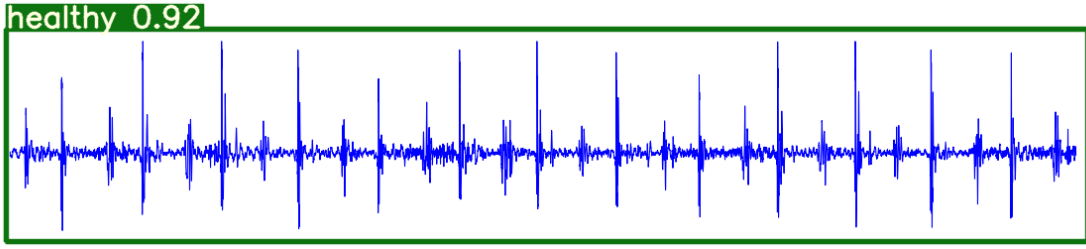


Şekil 4.8. Yolov4 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi

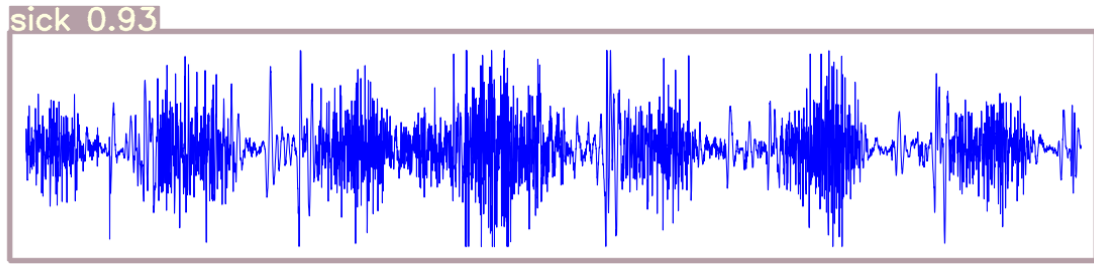


Şekil 4.9. Yolov4 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi

Yolov5 algoritmasının testi için de aynı yöntemler kullanılmıştır. Şekil 4.10’da ve Şekil 4.11’de Yolov5 algoritmasının test sonuçları verilmiştir.



Şekil 4.10. Yolov5 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi



Şekil 4.11. Yolov5 Algoritması Çıktısının Görüntü Üzerinde Gösterilmesi

#### 4.5. Derin Öğrenme Analizi Sonuçları

Yapılan eğitimler sonucunda, eğitilen algoritmaların performans ölçümleri için sadece tahmin oranı yetmemektedir. Performans ölçümü yaparken IoU (Intersection over Union) değeri, mAP (Mean Average Precision) ve AP (Average Precision) değeri, precision (kesinlik) değeri, recall (geri çağırma) değeri, TP (True Positive), FP (False Positive), TN (True Negative), FN (False Negative) gibi değerlere de bakılmalıdır.

IoU değeri, kesin referans verisinde bulunan nesnenin dikdörtgeni ile modelin tahmin ettiği nesnenin dikdörtgeninin arasındaki benzerliği ifade etmektedir. Bu değer, Denklem 4.1'de de verildiği gibi kesin referans olarak ölçüme katılan veride bulunan nesnenin içinde bulunduğu dikdörtgen ile modelin tahmin ettiği nesnenin içinde bulunduğu dikdörtgenin kesiştiği alanın bileşim alanına oranıdır.

$$IoU = \frac{\text{Dikdörtgenlerin Kesişim Alanı}}{\text{Dikdörtgenlerin Bileşim Alanı}} \quad (4.1)$$

AP ve mAP değerleri, genel manada ortalama kesinlik değerleridir. AP değeri, veri setinde bulunan her bir sınıf için ortalama kesinlik değeri iken, mAP değeri ise tüm sınıfların ortalama kesinlik değeridir. Yani mAP, AP değerlerinin ortalamasıdır.

Kesinlik ve geri çağırma grafiğinin altında kalan alan AP değerini vermektedir. AP değeri Denklem 4.2’de verilen eşitlik ile bulunur.

$$AP = \int_0^1 p(r)dr \quad (4.2)$$

Kesinlik ve geri çağırma değerleri ise tahminlerle ilgili oranları vermektedir. Kesinlik değeri, modelin doğru olarak tahmin ettiği değerlerin kaçının gerçekten doğru olduğunu veren orandır. Geri çağırma oranı ise modelin doğru olarak tahmin etmesi gereken oranın ne kadarını doğru olarak tahmin ettiğini verir. Kesinlik değerinin eşitliği Denklem 4.3’te, geri çağırma değerinin eşitliği ise Denklem 4.4’te verilmiştir.

$$Kesinlik (Precision) = \frac{TP (True Positive)}{TP (True Positive) + FP (False Positive)} \quad (4.3)$$

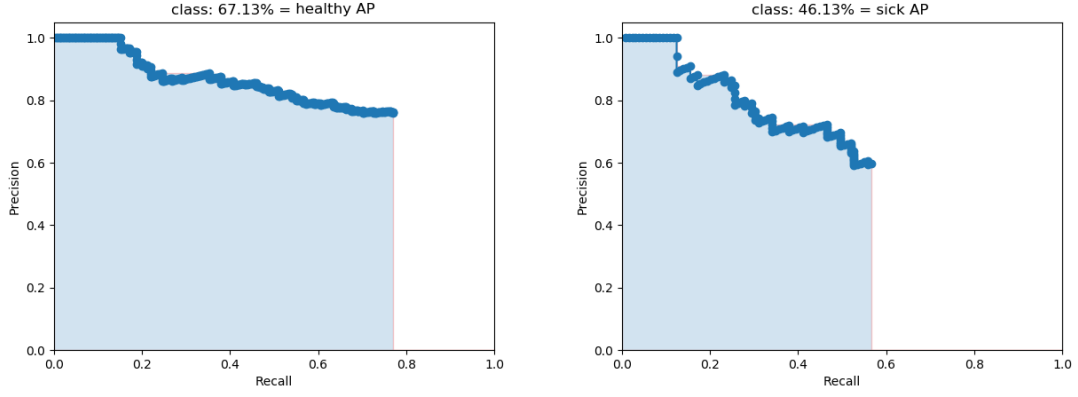
$$Geri Çağırma (Recall) = \frac{TP (True Positive)}{TP (True Positive) + FN (False Negative)} \quad (4.4)$$

Kesinlik ve geri çağırma değerleri hesaplanırken TP, FP, TN ve FN değerleri kullanılır. Bu değerlerin açıklaması aşağıda verilmiştir.

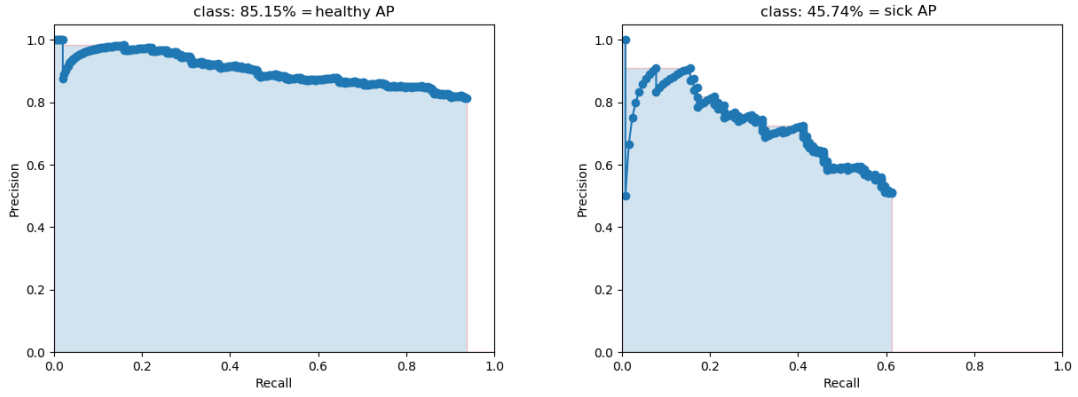
- TP: Eğitilen modelin doğru olarak etiketlenen veriyi doğru olarak tahmin etme sayısıdır.
- FP: Eğitilen modelin doğru olarak etiketlenen veriyi yanlış olarak tahmin etme sayısıdır.
- TN: Eğitilen modelin yanlış olarak etiketlenen veriyi yanlış olarak tahmin etme sayısıdır.
- FN: Eğitilen modelin yanlış olarak etiketlenen veriyi doğru tahmin etme sayısıdır.

Bu çalışmada yapılan eğitim ve testlerin, Yolov3, Yolov4 ve Yolov5 algoritmaları için sonuçları aşağıda paylaşılmıştır. Eğitilen algoritmaların ve modellerin performans ölçümleri için eğitime katılan veriler kesin referans (ground truth) olarak ve modellerin testleri sonucu çıkan veriler ise tespit sonuçları (detection result) olarak ölçüme katılmıştır. Performans ölçümü yapılırken, Joao Cartucho’nun Github üzerinden açık kaynak olarak paylaşmış olduğu “mAP (mean Average Precision)” isimli kod kullanılmıştır [42].

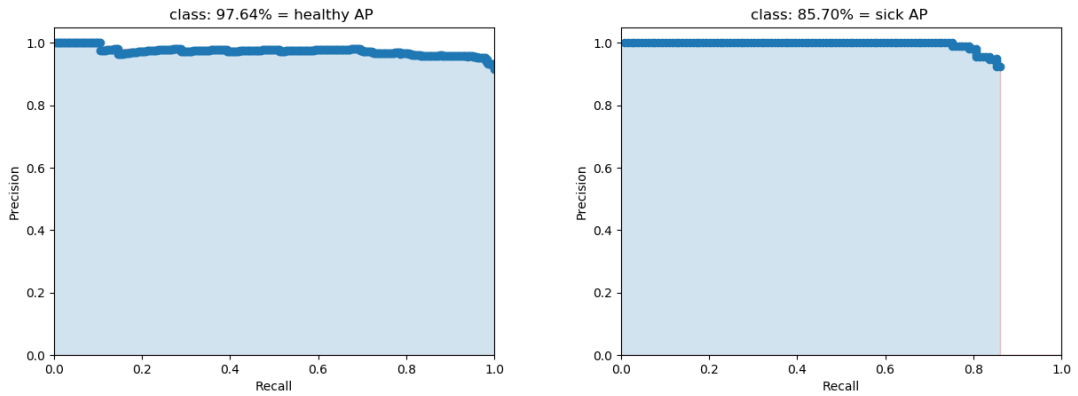
Yolov3, Yolov4 ve Yolov5 algoritmalarının IoU değeri minimum %50 iken performans testleri sonucunda sırasıyla, Şekil 4.12'deki, Şekil 4.13'teki ve Şekil 4.14'teki veriler elde edilmiştir.



Şekil 4.12. Yolov3 Algoritması AP<sub>50</sub> Değerleri



Şekil 4.13. Yolov4 Algoritması AP<sub>50</sub> Değerleri



Şekil 4.14. Yolov5 Algoritması AP<sub>50</sub> Değerleri

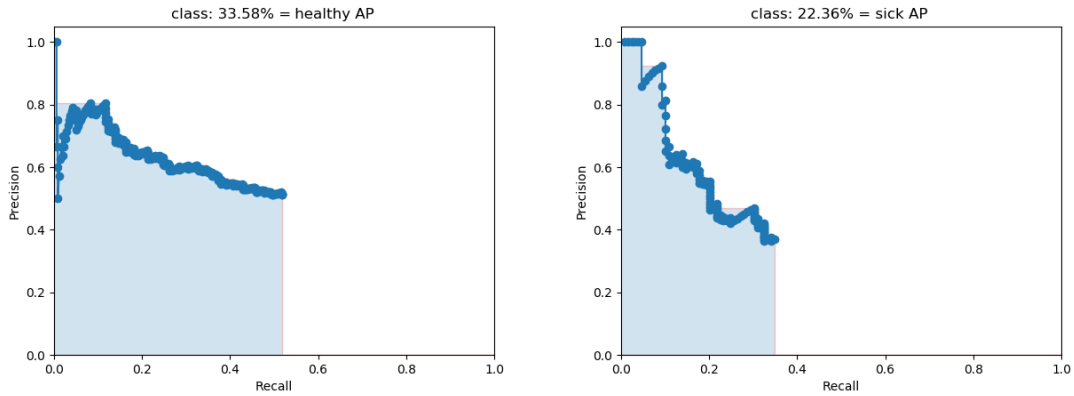
Elde edilen verilere göre IoU değeri minimum %50 iken hesaplanan AP değerleri ve mAP değerleri Tablo 4.1’de verilmiştir.

Algoritma	Healthy AP <sub>50</sub>	Sick AP <sub>50</sub>	mAP <sub>50</sub>
Yolov3	67,13%	46,13%	56,63%
Yolov4	85,15%	45,74%	65,45%
Yolov5	97,64%	85,70%	91,67%

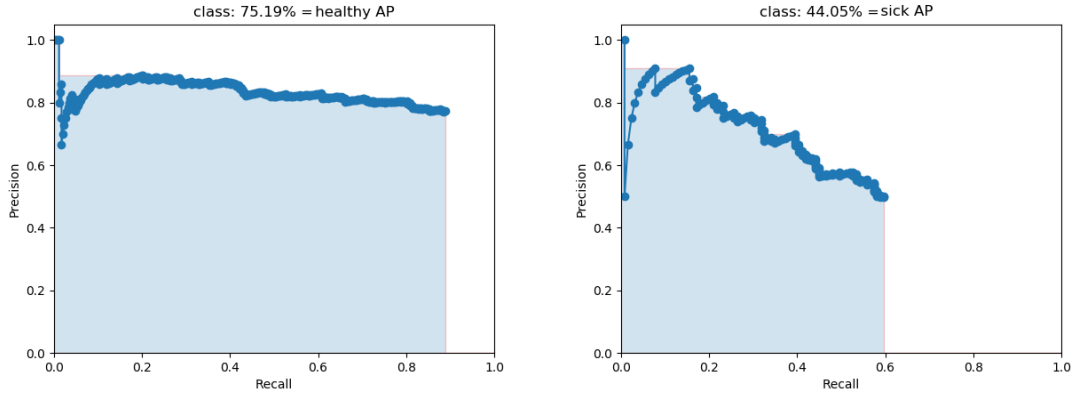
**Tablo 4.1.** Algoritmaların AP ve mAP Değerleri (IoU=0.50)

Tablo 4.1’de de görüldüğü üzere IoU değeri %50 iken ortalama kesinlik değeri bakımından %56,63 mAP değeri ile en düşük mAP değerine sahip algoritma Yolov3 olarak belirlenmiştir. %65,45 mAP değeri ile Yolov4 ikinci, %91,67 mAP değeri ile Yolov5 en yüksek ortalama kesinlik değerine sahip algoritma olmuştur.

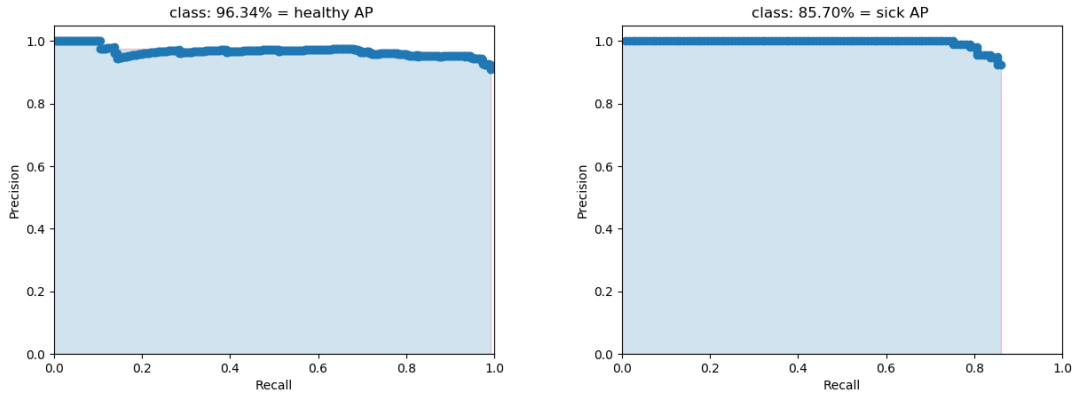
Yolov3, Yolov4 ve Yolov5 algoritmalarının IoU değeri minimum %75 iken performans testleri sonucunda sırasıyla, Şekil 4.15’teki, Şekil 4.16’daki ve Şekil 4.17’deki veriler elde edilmiştir.



**Şekil 4.15.** Yolov3 Algoritması AP<sub>75</sub> Değerleri



**Şekil 4.16.** Yolov4 Algoritması AP<sub>75</sub> Değerleri



**Şekil 4.17.** Yolov5 Algoritması AP<sub>75</sub> Değerleri

Elde edilen verilere göre IoU değeri minimum %75 iken hesaplanan AP değerleri ve mAP değerleri Tablo 4.2’de verilmiştir.

Algoritma	Healthy AP <sub>75</sub>	Sick AP <sub>75</sub>	mAP <sub>75</sub>
Yolov3	33,58%	22,36%	27,97%
Yolov4	75,19%	44,05%	59,62%
Yolov5	96,34%	85,70%	91,02%

**Tablo 4.2.** Algoritmaların AP ve mAP Değerleri (IoU=0.75)



Tablo 4.2’de de görüldüğü üzere IoU değeri %75 iken ortalama kesinlik değeri bakımından %27,97 mAP değeri ile en düşük mAP değerine sahip algoritma Yolov3 olarak belirlenmiştir. %59,62 mAP değeri ile Yolov4 ikinci, %91,02 mAP değeri ile Yolov5 en yüksek ortalama kesinlik değerine sahip algoritma olmuştur.

Elde edilen verilere göre IoU değeri minimum %50 ve %75 iken hesaplanan tüm AP değerleri ve mAP değerleri Tablo 4.3’de verilmiştir.

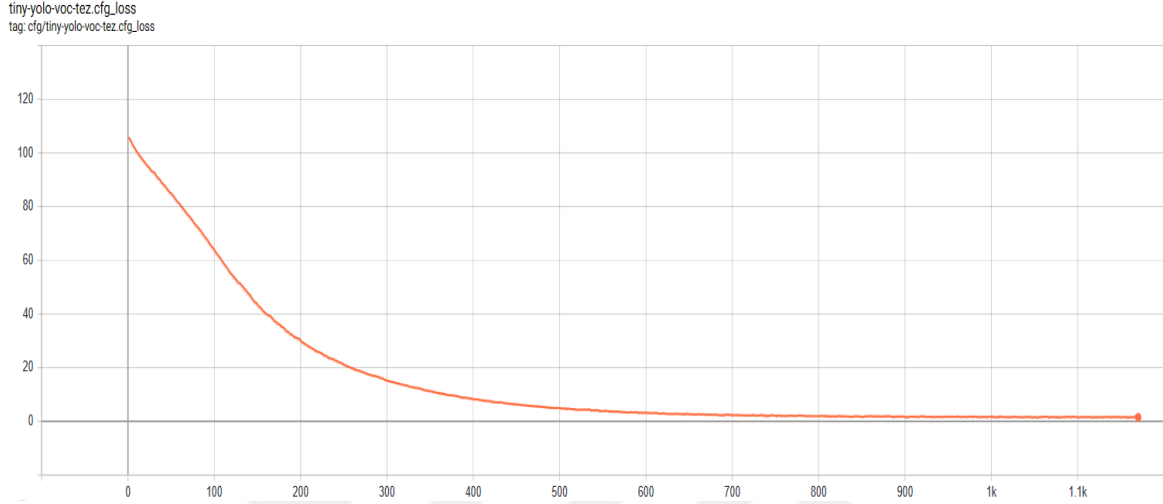
Algoritma	Healthy AP <sub>50</sub>	Sick AP <sub>50</sub>	Healthy AP <sub>75</sub>	Sick AP <sub>75</sub>	mAP <sub>50</sub>	mAP <sub>75</sub>
Yolov3	67,13%	46,13%	33,58%	22,36%	56,63%	27,97%
Yolov4	85,15%	45,74%	75,19%	44,05%	65,45%	59,62%
Yolov5	97,64%	85,70%	96,34%	85,70%	91,67%	91,02%

**Tablo 4.3.** Algoritmaların AP ve mAP Değerleri



## 5. BULGULAR VE TARTIŞMA

Var olan veri seti ile yapılan eğitimler sonucunda, Yolov3 algoritmasının hata değerinin 0'a yaklaşım grafiği Şekil 5.1'de verilmiştir.

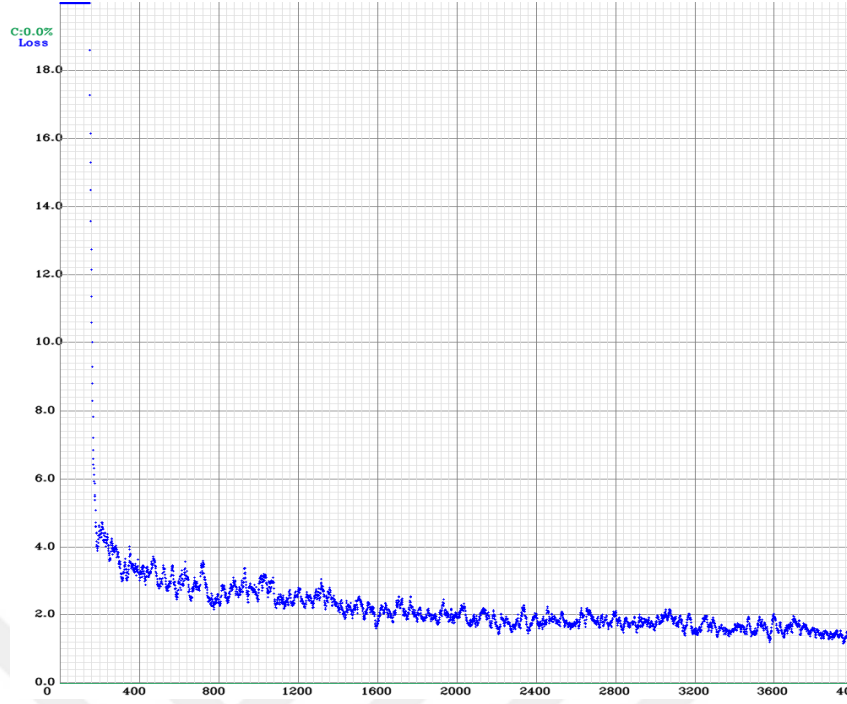


Şekil 5.1. Yolov3 Algoritmasında Hata Değerinin Eğitimle Sıfıra Yaklaşması

Hata değerinin 0'a yaklaşması, yapay sinir ağının eldeki veri seti ile iyi eğitildiğinin bir göstergesidir. Bu eğitim sonucunda, güncellenen ağırlık verileri elde edilmiştir. Bu ağırlık verileri ile Yolov3 algoritması test edilmiştir.

Eğitilen Yolov3 algoritması sağlıklı kalp sesi ile hasta kalp sesini ayırt etmeyi başarmıştır. Ortalama tahmin oranı ise IoU değeri minimum %50 iken %56,63, IoU değeri minimum %75 iken %27,97 bulunmuştur. Bu oran, eğitilen data set ile test edilen veriler arasındaki benzerliği belirtmektedir. Veri setinde bulunan örnek sayısı arttırıldıkça, tahmin oranı da artmaktadır. Kalp seslerinin yapay zekâ ile sınıflandırılması konusunda yapılan bir çalışmada yapay sinir ağının başarı oranının %95'in üzerinde olduğu sonucuna ulaşılmıştır [43]. Fakat yapılan çalışma, kalp seslerini sinyal olarak ele almaktadır. Burada eğitilen yapay sinir ağı ise kalp seslerini resim olarak ele almakta, sadece sinyalden değil resimden de çıkarım yapabilmektedir.

Var olan veri seti ile yapılan eğitimler sonucunda, Yolov4 algoritmasının hata değerinin 0'a yaklaşım grafiği Şekil 5.2'de verilmiştir.



**Şekil 5.2.** Yolov4 Algoritmasında Hata Değerinin Eğitimle Sıfıra Yaklaşması

Yolov4 algoritmasının eğitilmesi sonucu yapılan testlerde, Yolov4 algoritması da sağlıklı kalp sesi ile hasta kalp sesini ayırt etmeyi başarmıştır. Bu algoritmanın eğitimi sonucu ortalama tahmin oranı ise IoU değeri minimum %50 iken %65,45, IoU değeri minimum %75 iken %59,62 bulunmuştur. Yolov4'ün katman sayısının Yolov3'ten fazla olması başarı oranını arttırmıştır. Fakat yine de başarılı sayılabilecek bir test sonucu elde edilememiştir.

Yolov5 algoritmasının eğitilmesi sonucu yapılan testlerde de Yolov5 algoritması, sağlıklı kalp sesi ile hasta kalp sesini ayırt etmeyi başarmıştır. Bu algoritmanın eğitimi sonucu ortalama tahmin oranı ise IoU değeri minimum %50 iken %91,67, IoU değeri minimum %75 iken %91,02 bulunmuştur. Yolov5'in katman sayısının da Yolov4'ten fazla olması tahmin oranını arttıran en büyük etmenlerden biridir.

Literatürde, sinyal üzerinden konvolüsyonel sinir ağı kullanılarak sınıflandırma yapılabilen birçok çalışma vardır. Bu çalışmada ise veri setinde bulunan kalp sesleri eğitim ve testler sırasında görüntüye çevrilmiş ve hasta ile sağlıklı kalp sesi arasındaki farkın belirgin şekilde ortaya çıktığı görülmüştür. Veri setinde bulunan ses dalgalarının görüntüye çevrilmesinin nedeni, hem görüntü üzerinde öznelik bakımından belirgin farklılıkların olması, hem de hekimlerin daha hızlı erken tanı koyabilmesini sağlamaktır.

Bu çalışma ışığında ileride yapılacak olan çalışmalarda, mobil cihazlara (akıllı telefon, tablet vb.) yapılacak olan uygulama sayesinde, elektronik stetoskoplar ile dinledikleri kalp seslerinin tahmin sonucunu alacaklar, hem de görüntüsü ellerinde bulunan kalp seslerini tekrar dinlemek zorunda kalmayarak, mobil cihaz üzerinde bulunan kamera ile ses grafisinin analizini yapabileceklerdir. Bu sayede, hekimler erken tanı koymada yapay zekadan yararlanmış olacaklardır.





## 6. SONUÇ

Algoritmaların eğitilmesi ve test edilmesi ile elde edilen bulgular sonucunda, algoritmaların hasta ve sağlıklı kalp seslerini ayırt ettiği görülmüştür. Kalp seslerinin ayırt edilebilmesi haricinde, tahmin oranının algoritmaların katman sayıları ile orantılı olduğu sonucuna ulaşılmıştır. En düşük ortalama tahmin oranına sahip algoritma Yolov3 algoritması olmuştur. Yolov3 algoritmasını ikinci sırada Yolov4 algoritması takip etmektedir. Bu çalışma ile en başarılı algoritmanın Yolov5 olduğu sonucuna ulaşılmıştır. Yapay sinir ağının tahmin oranı, daha büyük veri setleri ile de artırılabilir. Veri seti içerisindeki örnek verilerin artırılması, tahmin oranını da arttırmaktadır. Bununla birlikte veri setinin büyüklüğü, eğitim için harcanan zamanı ve modelin büyüklüğünü de arttırmaktadır. Tüm bu etmenlerin yanı sıra veri setinin çeşitliliği de tahmin oranını artmasını sağlayacaktır.

Veri seti toplamak zaman açısından maliyetli olduğundan bu çalışmada açık kaynak veri seti kullanılmıştır. İleride yapılacak veri setini genişletme ve çeşitlendirme çalışmalarında hastane ortamından verilerin toplanması amaçlanmaktadır. Hastane ortamından veri setlerinin toplanması sırasında elektronik stetoskoplar veya ekokardiyografi cihazlarından alınan ses verilerinin kullanılması hedeflenmektedir. Böylelikle veri setinde bulunan ses verilerinin gürültüsüz olarak kayıt edilmesi öngörülmektedir. Hastane ortamından elde edilen veri setleri ile çeşitlilik artacak ve veri setini genişletme çalışmaları amaçlanan noktaya ulaşacaktır.

Bu çalışmanın ışığında, gelecek çalışmalarda veri seti içeriği sayı ve çeşitlilik bakımından artırılarak tahmin oranının daha da yükseltilmesi ve algoritma olarak Yolov5 algoritmasının kullanımı hedeflenmektedir. Ayrıca, hekimlerin erken tanı koymasını kolaylaştırmak için bu çalışmanın mobil cihazlar için uygulama olarak geliştirilmesi planlanmaktadır. Mobil uygulama platformu olarak, akıllı telefonlar ve tabletler düşünülmekte, böylelikle hekimlerin bu tarz cihazlara ulaşabilmesinin kolaylaştırılması hedeflenmektedir. Bu sayede hekimler, stetoskoplar ile dinledikleri kalp seslerinin tahmin sonucunu alacaklar, hem de görüntüsü ellerinde bulunan kalp seslerini tekrar dinlemek zorunda kalmayarak, mobil cihazlar üzerinde bulunan kamera ile ses grafisinin analizini yapabileceklerdir. Bu sayede, hekimler erken tanı koymada yapay zekadan yararlanmış olacaklardır.





## KAYNAKÇA

- [1] Şeker, A., Diri, B., Balık H.H. (2017) Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme, Gazi Mühendislik Bilimleri Dergisi, 47-64.
- [2] Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I., Sricharan, K. (2017) Recognizing Abnormal Heart Sounds Using Deep Learning, International Joint Conference on Artificial Intelligence, 19-25 August, Melbourne.
- [3] Türkiye İstatistik Kurumu, <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=27620>
- [4] Demir, F., Şengür. A., Çavaş, M. (2018) Derin Öznitelikler ve Destek Vektör Makineler ile Kalp Seslerinin Sınıflandırılması, International Conference on Artificial Intelligence and Data Processing, 28-30 Eylül, Malatya.
- [5] Zhang, Y., Ayyar, S., Chen, L.H., Li, E.J., Segmental Convolutional Neural Networks for Detection of Cardiac Abnormality With Noisy Heart Sound Recordings, 2016.
- [6] Alam, S., Banerjee, R., Bandyopadhyay, S., Murmur Detection Using Parallel Recurrent & Convolutional Neural Networks, 2018.
- [7] Sağlık Hizmetleri Dolaşım Sistemi, T.C. Milli Eğitim Bakanlığı, Ankara, 2015.
- [8] Garip, İ. C. (2014) Biyoaktif Peptit Amfifiller ile Çizgili Kas ve Kalp Kası Oluşumu İçin Mikroçevresel Sinyalleri Programlama. Yüksek Lisans Tezi, İhsan Doğramacı Bilkent Üniversitesi Mühendislik ve Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [9] Eroğlu, S. (2016) Avrupa Kardiyoloji Derneği 2015 Perikart Hastalıkları Tanı ve Yönetimi Kılavuzu ne yenilikler getirdi?, Türk Kardiyol Dern Ars 2016;44(1):1-3. doi:10.5543/tkda.2016.50067
- [10] Çelebi, G. (2017) Sayısal Steteskop ile Elde Edilen Kalp Ses (Fonokardiyogram) Sinyallerinin Bölütlenmesi ve Sınıflandırılması. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye.

- [11] Diker, A. (2019) Uç Öğrenme Makineleri Kullanarak EKG İşaretlerinin Sınıflandırma Başarımlarının İyileştirilmesi. Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elâzığ, Türkiye.
- [12] Kaplanoğlu, E., Ülkir, O., Akgün, G. (2017) Biyomekatronik Sistemlere Giriş, 1.Basım, Papatya Yayıncılık Eğitim, İstanbul.
- [13] Civelek, Y. (2005) Kardiyak Pasif Akusitk Konumlandırma İçin Yeni Bir Donanım Tasarımı. Yüksek Lisans Tezi, Boğaziçi Üniversitesi Biyomedikal Mühendisliği Enstitüsü, İstanbul, Türkiye.
- [14] Kotan, S. (2014) İkinci Kalp Sesi Çiftleşmesinin Anlık Frekans İzlenmesi Yöntemiyle Kestiriminde Farklı Zaman-Frekans Dönüşümü Yöntemlerinin Karşılaştırılması. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye.
- [15] Varghees, V. N., Ramachandran, K. I. (2014) A novel heart sound activity detection framework for automated heart sound analysis, Biomedical Signal Processing and Control, 174 – 188. doi: 10.1016/j.bspc.2014.05.002
- [16] Turing, A. M. (1950) Computing Machinery and Intelligence. Mind 49: 433-460.
- [17] Arf, C. (1959) Makine Düşünebilir mi? ve Nasıl Düşünebilir?, Atatürk Üniversitesi – Üniversite Çalışmalarını Muhite Yayma ve Halk Eğitimi Yayınları Konferanslar Serisi No:1, 91-103.
- [18] Adalı, E. (2017, Ocak-Mart). İnsanlaşan Makinalar - Yapay Zekâ. İstanbul Teknik Üniversitesi Vakfi Yayımları, 13(75). Erişim adresi: <http://www.itu.edu.tr/docs/default-source/haber-s1%C4%B1der---ekler/sayi75.pdf?sfvrsn=2>
- [19] Shalev-Shwartz, S., Ben-David, S. (2014) Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, New York.
- [20] Nilsson, N. J. (1998) Introduction to Machine Learning, Stanford University, San Francisco, California.
- [21] Negnevitsky, M. (2005) Artificial Intelligence: A Guide to Intelligent Systems, Pearson Education Limited, Essex.

- [22] Yapay Zeka ve Makine Öğrenimi Arasındaki Fark Nedir?, <https://www.exastax.com.tr/makine-ogrenimi/yapay-zeka-ve-makine-ogrenimi-arasindaki-fark-nedir/>
- [23] Kukreja, H., Bharath, N., Siddesh, C. S., Kuldeep, S., An Introduction to Artificial Neural Network, International Journal Of Advance Research And Innovative Ideas In Education, 2016.
- [24] <https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/>
- [25] Güvenç, U., Biroğul, S., Sönmez, Y., Yapay Sinir Ağları Eğitim Seti, The Proceedings of 7th International Educational Technology Conference, 2007.
- [26] Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması, <https://medium.com/@ayyucekizrak/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9Ft%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd>
- [27] Artificial Intelligence vs. Machine Learning vs. Deep Learning: What's the Difference. (2017, 10 Nisan). Erişim adresi: <https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f>
- [28] Doğan, F., Türkoğlu, İ., (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. DÜMF Mühendislik Dergisi, 10:2, 409-445.
- [29] Pervan, N. (2019) Derin Öğrenme Yaklaşımları Kullanarak Türkçe Metinlerden Anlamsal Çıkarım Yapma. Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Türkiye.
- [30] Tan, Z. (2019) Derin Öğrenme Yardımıyla Araç Sınıflandırma. Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elâzığ, Türkiye.
- [31] Darknet, <https://github.com/pjreddie/darknet>
- [32] Darkflow, <https://github.com/thtrieu/darkflow>

- [33] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., You Only Look Once: Unified, Real Time Object Detection, 2015.
- [34] Bochkovskiy, A., Wang, C.-Y., Hong-Yuan, M., YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020.
- [35] Ultralytics Yolo v5, <https://github.com/ultralytics/yolov5>, doi: 10.5281/zenodo.3983579.
- [36] Classifying Heart Sounds Challenge, <http://www.peterjbentley.com/heartchallenge/>
- [37] Advance Renamer Programı, <https://www.advancedrenamer.com/>
- [38] Image Label Programı, <https://github.com/tzutalin/labelImg>
- [39] Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., CSPNet: A New Backbone That Can Enhance Learning Capability of CNN, 2019.
- [40] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., Path Aggregation Network for Instance Segmentation, 2018.
- [41] He, K., Zhang, X., Ren, S., Sun, J., Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, 2015.
- [42] mAP (mean Average Precision), <https://github.com/Cartucho/mAP>
- [43] Say, Ö. (2002) Kalp Seslerinin Analizi ve Yapay Sinir Ağları ile Sınıflandırılması. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye.

## ÖZGEÇMİŞ

Murat Alan, 1994 yılında İstanbul'da doğdu. Orta öğrenimini 2012 yılında Habire Yahşi Anadolu Lisesi'nde, lisans öğrenimini ise 2016 yılında Marmara Üniversitesi Mekatronik Mühendisliği bölümünde tamamladı. Yüksek lisans eğitimine ise yine Marmara Üniversitesi Mekatronik Mühendisliği'nde devam etmektedir. Bununla birlikte Anadolu Üniversitesi'nde Çalışma Ekonomisi ve Endüstri İlişkileri bölümünde açıköğretim lisans eğitimi görmektedir. Aynı zamanda İstanbul Üniversitesi Sivil Hava Ulaştırma İşletmeciliği önlisans mezunudur. Halen PIXSelect Teknoloji A.Ş. firmasında "Project Supervisor" pozisyonunda çalışmaktadır.

