

T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**DALGACIK DÖNÜŞÜMÜ VE DERİN ÖĞRENME TABANLI  
YÖNTEMLER KULLANARAK BEYİN TÜMÖRLERİNİN  
SINIFLANDIRILMASI**

**DOKTORA TEZİ**

**Zihni KAYA**

**Bilgisayar Mühendisliği Ana Bilim Dalı  
Bilgisayar Mühendisliği Programı**

**KASIM, 2023**



T.C.  
İSTANBUL AYDIN ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**DALGACIK DÖNÜŞÜMÜ VE DERİN ÖĞRENME TABANLI  
YÖNTEMLER KULLANARAK BEYİN TÜMÖRLERİNİN  
SINIFLANDIRILMASI**

**DOKTORA TEZİ**

**Zihni KAYA**  
**(Y1813.610002)**

**Bilgisayar Mühendisliği Ana Bilim Dalı**  
**Bilgisayar Mühendisliği Programı**

**Tez Danışmanı: Prof. Dr. Zafer ASLAN**

**KASIM, 2023**

## TEZ SINAV TUTANAĐI

## ONUR SÖZÜ

Doktora tezi olarak sunduđum “Dalgacık Dönüşümü ve Derin Öğrenme Tabanlı Yöntemler Kullanarak Beyin Tümörlerinin Sınıflandırılması” adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Kaynakça’da gösterilenlerden oluştuđunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. 9/11/2023)

Zihni KAYA

## ÖNSÖZ

Bu çalışma, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Doktora Tezi olarak hazırlanan “Dalgacık Dönüşümü ve Derin Öğrenme Tabanlı Yöntemler Kullanarak Beyin Tümörlerinin Sınıflandırılması” isimli tezi içermektedir. Çalışmamın her aşamasında bilgi ve deneyimleri ile yardımcı olan danışmanım Sayın Prof. Dr. Zafer Aslan’a teşekkür ederim. Çalışmam boyunca destekleri için Prof. Dr. Ali Güneş ve Prof. Dr. Ali Okatan’a da teşekkür ederim.

Kasım, 2023

Zihni KAYA

# DALGACIK DÖNÜŞÜMÜ VE DERİN ÖĞRENME TABANLI YÖNTEMLER KULLANARAK BEYİN TÜMÖRLERİNİN SINIFLANDIRILMASI

## ÖZET

Dünya Sağlık Örgütü (DSÖ) raporlarına göre beyin tümörleri nedeniyle meydana gelen ölümlerin sayısı her geçen gün artmaktadır. Kötü huylu beyin tümörleri çok hızlı bir biçimde büyürler ve yayılırlar. Bu nedenle hastalığın erken teşhisi hayati bir önem taşımaktadır. Beyin tümörlerinin tanı ve teşhisinde Manyetik Rezonans Görüntüleme (MRG) yöntemi yaygın olarak kullanılmaktadır. Uzmanlar tarafından MRG görüntüleri incelenerek tümörün türü tespit edilmektedir. Fakat, uzmanlar tarafından manuel olarak yapılan tümör sınıflandırması çok fazla zaman alabilir hem de yanlış kararlar verilmesine yol açabilir. Bu nedenle, beyin tümörlerini otomatik olarak sınıflandırma çalışmaları önemli bir araştırma konusu haline gelmiştir. Bu tez çalışmasında beyin MRG görüntülerinden tümör türlerini otomatik olarak sınıflandıran bir yöntem önerilmiştir. Ön işlem aşamasında; öznitelik çıkartmak, boyut azaltmak ve gürültü gibi istenmeyen verileri ortandan kaldırmak için beyin MRG görüntülerine Ayrık Dalgacık Dönüşümü (ADD) uygulanmıştır. Elde edilen dalgacık öznitelikleri önceden eğitilmiş Evrişimli Sinir Ağı (ESA) modellerinden Inception V3 ağının girdisi olarak kullanılmıştır. Deneysel çalışmalar; 1621 gliyom, 1645 menenjiyom, 1757 hipofiz bezi ve 2000 normal beyin MRG görüntüsünün bulunduğu veri seti üzerinde yapılmıştır. Önerilen modelin doğruluğunu ölçmek için 5 kat çapraz doğrulama yöntemi uygulanmış ve ortalama %99,58 doğruluk başarıımı elde edilmiştir. Bu tez çalışmasından elde edilen sonuçlar derin öğrenme tabanlı yöntemler kullanılarak beyin tümörlerinin başarılı bir şekilde sınıflandırılabileceğini göstermiştir.

**Anahtar kelimeler:** Beyin tümörü sınıflandırma, ayırık dalgacık dönüşümü, evrişimli sinir ağları, transfer öğrenme, Inception v3

# **CLASSIFICATION OF BRAIN TUMORS USING WAVELET TRANSFORM AND DEEP LEARNING-BASED METHODS**

## **ABSTRACT**

According to the World Health Organization (WHO) reports, the number of deaths due to brain tumors is increasing day by day. Malignant brain tumors grow and spread very quickly. Therefore, early diagnosis of the disease is of vital importance. Magnetic Resonance Imaging (MRI) method is widely used in the diagnosis and diagnosis of brain tumors. The type of tumor is determined by examining MRI images by specialists. However, manual tumor classification by experts can take too much time and lead to wrong decisions. Therefore, studies to automatically classify brain tumors have become an important research topic. In this thesis, a method that automatically classifies tumor types from brain MRI images is proposed. In the pre-processing stage; Discrete Wavelet Transform (DWT) has been applied to brain MRI images for feature extraction, size reduction, and elimination of unwanted data such as noise. Obtained wavelet features are used as input of Inception V3 network, which is one of the pre-trained Convolutional Neural Network (CNN) models. Experimental studies were performed on the dataset of 1621 glioma, 1645 meningioma, 1757 pituitary gland and 2000 normal brain MRI images. To measure the accuracy of the proposed model, the 5-fold cross-validation method was applied and an average accuracy of 99.58% was achieved. The results obtained from this thesis study showed that brain tumors can be successfully classified using deep learning-based methods.

**Keywords:** Brain tumor classification, discrete wavelet transform, convolutional neural networks, transfer learning, Inception v3



## İÇİNDEKİLER

ONUR SÖZÜ.....	iii
ÖNSÖZ.....	iv
ÖZET.....	v
ABSTRACT.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELER LİSTESİ.....	iv
<b>I. GİRİŞ .....</b>	<b>1</b>
<b>II. VERİ VE YÖNTEM.....</b>	<b>6</b>
A. Veri Seti.....	6
B. Ön İşlem.....	6
1. Histogram Germe.....	7
C. Dalgacık Dönüşümü .....	7
1. Sürekli Dalgacık Dönüşümü .....	10
2. Ayrık Dalgacık Dönüşümü .....	11
D. Ayrık Kosinüs Dönüşümü .....	14
E. Yapay Sinir Ağları.....	15
1. Biyolojik Sinir Hücresi .....	15
2. Yapay Sinir Hücresi .....	16
3. Basit Algılayıcı Modeli (Perseptron).....	19
4. Çok Katmanlı Algılayıcı.....	21
F. Sınıflandırma Ölçüm Metrikleri .....	27
1. Doğruluk Oranı .....	27
2. Karışıklık Matrisi.....	28
3. F1-Puanı.....	29
G. Transfer Öğrenme .....	30
<b>III. DERİN ÖĞRENME.....</b>	<b>33</b>
A. Evrişimli Sinir Ağları .....	35
1. Evrişim Katmanı .....	37
2. Havuzlama Katmanı.....	40
3. Tam Bağlantı Katmanı .....	41
B. Evrişimli Sinir Ağı Modelleri.....	42

1. LeNet.....	42
2. AlexNet.....	44
3. GoogLeNet.....	45
4. Inception V3.....	47
<b>IV. DENEYSSEL SONUÇLAR VE TARTIŞMA .....</b>	<b>49</b>
<b>V. SONUÇLAR VE ÖNERİLER .....</b>	<b>55</b>
<b>VI. KAYNAKÇA .....</b>	<b>57</b>
<b>EKLER.....</b>	<b>64</b>
<b>ÖZGEÇMİŞ.....</b>	<b>71</b>

## ÇİZELGELER LİSTESİ

Çizelge 1 Toplama fonksiyonları.....	17
Çizelge 2 Doğrulama setleri için erken durdurma eğitim adım sayıları.....	48
Çizelge 3 Doğrulama setleri için doğruluk oranları.....	48
Çizelge 4 İlgili çalışmaların doğruluk oranları.....	49
Çizelge 5 Doğrulama seti-1 için karışıklık matrisi.....	50
Çizelge 6 Doğrulama seti-2 için karışıklık matrisi.....	50
Çizelge 7 Doğrulama seti-3 için karışıklık matrisi.....	51
Çizelge 8 Doğrulama seti-4 için karışıklık matrisi.....	51
Çizelge 9 Doğrulama seti-5 için karışıklık matrisi.....	52
Çizelge 10 Sınıflara göre performans metriklerinin ortalama değerleri.....	52

## ŞEKİLLER LİSTESİ

Şekil 1: Farklı düzlemlere ait beyin MRG görüntüleri.....	2
Şekil 2: Normal ve tümörlü beyin MRG görüntüleri.....	6
Şekil 3: Veri ön işlem aşamaları.....	7
Şekil 4: Bir sinüs dalgası ve bir dalgacık.....	9
Şekil 5: Zaman ve frekans uzayındaki sinyaller.....	9
Şekil 6: S sinyalinin düşük (A) ve yüksek (D) frekans bileşenlerine ayrılması.....	12
Şekil 7: Dalgacık ayrıştırma ağacı.....	13
Şekil 8: Biyolojik sinir hücresi.....	16
Şekil 9: Yapay sinir hücresi.....	16
Şekil 10: Aktivasyon fonksiyonları .....	17
Şekil 11: Basit algılayıcı yapısı.....	19
Şekil 12: Çok katmanlı algılayıcı yapısı.....	21
Şekil 13: İkili sınıflandırma için karışıklık matrisi.....	28
Şekil 14: Transfer öğrenme diyagram gösterimi.....	31
Şekil 15: Farklı YZ disiplinleri arasındaki ilişkiyi gösteren Venn şeması.....	34
Şekil 16: ESA sinir düğümü.....	36
Şekil 17: ESA Mimarisi.....	36
Şekil 18: İki boyutlu evrişim örneği.....	38
Şekil 19: Maksimum ve ortalama havuzlama işlemi.....	40
Şekil 20: Düzleştirme işlemi.....	41
Şekil 21: Tam bağlantı katmanı gösterimi.....	41
Şekil 22: MNIST veri seti.....	42
Şekil 23: LeNet modeli.....	43
Şekil 24: AlexNet modeli.....	43
Şekil 25: GoogLeNet modeli blok şeması.....	44
Şekil 26: Basit inception modülü.....	45
Şekil 27: Inception modülü.....	46
Şekil 28: Inception v3 modeli.....	46

Şekil 29: Inception-A, Inception-B ve Inception-C modülleri.....	47
Şekil 30: Kategoriye göre performans metriklerinin ortalama değerleri.....	53

## KISALTMALAR LİSTESİ

- ADD** : Ayrık Dalgacık Dönüşümü  
**AFD** : Ayrık Fourier Dönüşümü  
**AKD** : Ayrık Kosinüs Dönüşümü  
**BT** : Bilgisayarlı Tomografi  
**BT** : Bilgisayarlı Tomografi  
**ÇKA** : Çok Katmanlı Algılayıcı  
**DD** : Dalgacık Dönüşümü  
**ESA** : Evrişimli Sinir Ağları  
**KZFD** : Kısa Zaman Fourier Dönüşümü  
**MR** : Manyetik Rezonans  
**MRG** : Manyetik Rezonans Görüntüleme  
**SDD** : Sürekli Dalgacık Dönüşümü  
**YSA** : Yapay Sinir Ağları  
**YZ** : Yapay Zeka

## I. GİRİŞ

İnsan sinir sisteminin merkezi organı olan beyin insan vücudundaki en büyük ve en karmaşık organlardan biridir. İnsan beyni, hayat fonksiyonlarını ve zihinsel süreçleri düzenleyen yaklaşık 1,2-1,4 kg ağırlığında bir organdır. Beyinde 100 milyardan fazla sinir hücresinin bulunduğu tahmin edilmektedir. Sinir hücrelerinin diğer sinir hücrelerine mesaj iletmesini sağlayan sinaps adı verilen trilyonlarca bağlantıya sahip olduğu düşünülmektedir (Kaplan, 2020: 13).

Beyin bölgesindeki hücrelerin anormal bir şekilde gelişmesiyle oluşan doku kütlelerine tümör denir (Anitha et al., 2017: 354). Beyin tümörü, beyinde bulunan hücrelerin farklılaşarak kontrolsüz bir biçimde çoğalmasıyla veya vücudun başka bir yerindeki kanser hücrelerinin beyne gelerek yayılmasıyla meydana gelen bir hastalıktır. Beyin tümörlerinin oluşum nedenleri tam olarak bilinmese de bazı kimyasal maddelere maruz kalma, genetik faktörler, radyasyona maruz kalma, bağışıklık sistemi sorunları ve bazı virüsler risk faktörleri arasında yer almaktadır. Beyin tümörü görülen bir kişide kusma, uyuşukluk, kişilik ve davranış bozuklukları, baş ağrısı ve el titremesi gibi belirtiler görülmektedir (Aslan, 2022: 399).

Beyin tümörleri iyi huylu (kansersiz) ve kötü huylu (kansersiz olmayan) olmak üzere iki ana gruba ayrılır. İyi huylu beyin tümörleri, beynin diğer bölgelerine yayılmazlar ve cerrahi müdahale ile çıkartılabilirler. Ama, beyindeki diğer dokuların normal çalışmasını engelleyebilirler. Kötü huylu beyin tümörleri ise diğer organlardan beyne yayıldığında ortaya çıkar. Bu tümörlerin sınırları tam olarak belli olmadığından cerrahi müdahale ile çıkarılmaları zordur (Kazdal, 2013: 18).

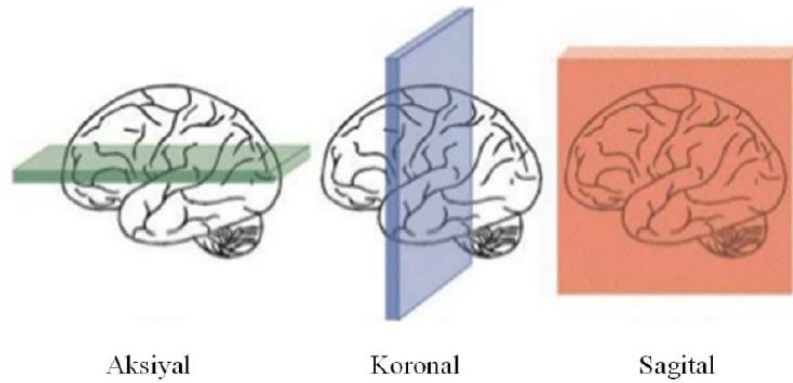
Yetişkinlerde en çok görülen kötü huylu beyin tümörlerinden olan gliomalar, glial hücrelerden kaynaklanmaktadır (Rehman et al., 2019: 599). Glioma ölümcül bir kanser türü olup bu hastalığa yakalananların yaşama şansı oldukça düşüktür. Dünya Sağlık Örgütü (World Health Organization) glioma tümörleri yüksek dereceli tümörler ve düşük dereceli tümörler olmak üzere iki gruba ayırmaktadır.

Radyoloji, bazı hastalıkların tanı ve tedavisi için görüntüleme yöntemlerinin kullanıldığı tıpta bir uzmanlık alanıdır. Tümörlü hücreler şekil, çekirdek ve boyut yapıları bakımından normal hücrelere göre farklı özelliklere sahiptirler. Ortaya çıkan semptomlara ve araştırılan bölgeye göre görüntüleme teknolojisi tercih edilmektedir. Beyin tümörlerinin tanısında, Manyetik Rezonans Görüntüleme (MRG) ve Bilgisayarlı Tomografi (BT) taramaları yaygın olarak kullanılmaktadır (Arı, 2019: 1).

BT, x-ışınlarını kullanarak vücudun kesitsel görüntüsünü oluşturmaya yarayan radyolojik bir teşhis yöntemidir. BT’de kontrast sayısının çokluğu dokuları birbirinden ayırdığından incelenen bölge daha iyi görüntülenir.

MRG, fiziğin temel ilkelerini kullanarak doku farklılıklarından kaynaklı görüntüler elde edilebilmesini sağlayan bir tıbbi görüntüleme yöntemidir (Yüksel, 2019: 1). Beynin yumuşak dokusuyla ilgili yüksek çözünürlüklü görüntünün yanı sıra aynı dokuyla ilgili birden çok şekilsel bilgi de vermektedir (Aslan, 2021: 1). Bu nedenle, MRG yöntemi beyin tümörlerinin tespitinde yaygın olarak kullanılmaktadır.

MRG yöntemiyle hastanın pozisyonu değiştirilmeden farklı düzlemlerde görüntüler alınabilmektedir (Oyar, 2008: 32). Aksiyel düzlemde beyin kesitinin üstten görünümü, sagittal düzlemde beyin kesitinin yandan görünüşü, koronal düzlemde ise beyin kesitinin arkadan görünüşü elde edilir. Farklı düzlemlerden elde edilen beyin MRG görüntüleri sayesinde tümörlerle ilgili hacimsel, şekilsel ve dokusal olarak daha doğru yorum yapılabilir (Duman, 2021: 11). Şekil 1’de farklı düzlemlere ait beyin MRG görüntüleri verilmiştir.



Şekil 1: Farklı düzlemlere ait beyin MRG görüntüleri

ABD’de her yıl %29,7’si kötü huylu olmak üzere yaklaşık 84,000 kişiye beyin tümörü teşhisi konulmaktadır. 2023 yılında ise 24,810 yeni kötü huylu beyin tümörü



vakası görüleceği ve 18,990 kişinin bu nedenle öleceği tahmin edilmektedir (Siegel et al., 2023: 21). Bu nedenle beyin tümörlerinin sınıflandırılması amacıyla yapılan çalışmalar büyük önem arz etmekte ve her geçen gün artmaktadır.

Beyaz madde, gri madde ve beyin omurilik sıvısı gibi farklı dokulardan oluşan beyin yapısından tümörün tespit edilmesi oldukça zor ve önemli bir görevdir. Şekil ve büyüklükleri farklı olan beyin tümörleri beynin farklı bölgelerinde bulunabilirler. Büyük miktarda MRG görüntülerinin uzmanlar tarafından gerekli zaman içerisinde incelenmesi zaman alıcı ve zor bir iştir. Bu nedenle, beyin tümörlerinin bilgisayar destekli karar sistemleri ile tespit edilmesine ihtiyaç duyulmaktadır. Yapay öğrenme yöntemleri MRG görüntülerinden beyin tümörlerinin tespit edilebilmesi için hızlı ve doğru çözümler sunmaktadır.

Ayrık Dalgacık Dönüşümü (ADD), bir görüntüden anlamlı bilgiler çıkartmak, gürültü gibi istenmeyen verileri ortadan kaldırmak ve boyut azaltmak için yaygın olarak kullanılmaktadır. ADD, herhangi bir sinyal veya görüntünün çok çözünürlüklü gösterimini sağlayan güçlü bir matematiksel araçtır. Sinyali veya görüntüyü farklı frekans bileşenlerine ayrıştırır ve ardından her bileşeni kendi ölçeğine uygun bir çözünürlükle inceler (Aggarwal and Agrawal, 2012).

Yapay Sinir Ağları (YSA), biyolojik sinir sisteminde bulunan sinir hücrelerinden esinlenerek geliştirilmiş basit proses elemanlarından oluşmaktadır. Sinir hücreleri, bir sinir hücresinden diğer sinir hücresine bilgileri ileten ağırlıklandırılmış bağlantılar ile bağlıdır. Sinir hücreleri arasındaki bilgi iletimi sinapslar yoluyla gerçekleşir. YSA, sinir hücrelerine gelen girdiyi kendi ağırlığı ile çarparak toplar. Sonrasında, yapay sinir hücresine gelen net girdi belirli bir aktivasyon fonksiyonundan geçirilerek çıktı üretilir. Üretilen çıktı dış dünyaya veya diğer sinir hücrelerine gönderilir.

Tıbbi görüntü sınıflandırmasında olduğu gibi veri seti boyutları küçük olduğunda transfer öğrenme yaklaşımı yaygın olarak kullanılmaktadır. Veri seti küçük olduğunda aşırı öğrenme (overfitting) gerçekleşebilir. Transfer öğrenme yaklaşımında daha önceden büyük veri setiyle eğitilmiş modelden elde edilen bilgiler yeni modelde kullanılır. Transfer öğrenmenin avantajlarından bir diğeri, düşük hesaplama gücüne ve kısa eğitim süresine sahip oluşudur. Model, önceden eğitilmiş ağ katmanlarından gelen

ağırlık değerlerini kullanır ve yalnızca tam bağlantılı son katman ağırlıkları eğitilir (Türkdamar vd., 2023: 46).

Evrişimli Sinir Ağları (ESA) bilgisayarla görü alanında yaygın olarak kullanılan en popüler derin öğrenme mimarisidir. ESA'nın geleneksel ağlara kıyasla en büyük avantajı, önemli özellikleri herhangi bir insan müdahalesi olmadan otomatik olarak algılamasıdır (Alzubaidi et al., 2021). ESA'nın ilk katmanlarında kenar bilgisi gibi daha basit özellikler çıkarılırken daha derin katmanlara gidildikçe daha karmaşık özellikler çıkarılmaktadır (Kızrak ve Bolat, 2018: 266).

Literatürde, beyin MRG görüntüleri üzerinden beyin tümörlerinin sınıflandırılması amacıyla ESA modelleri yaygın olarak kullanılmaktadır. Özkaraca ve arkadaşları basit ESA, önceden eğitilmiş VGG16Net ve DenseNet modelleri ve yoğun katmanlara sahip ESA modellerini kullanarak beyin tümörlerini sınıflandırmışlardır (Özkaraca et al., 2023). Yaptıkları deneysel çalışmalarda transfer öğrenme yönteminin doğruluk başarımını arttırmadığını gözlemlemişlerdir. Transfer öğrenme yöntemini kullanmadan geliştirdikleri yoğun katmanlara sahip ESA modeli ile %94-97 arasında doğruluk başarımları elde etmişlerdir.

Gomez Guzman ve arkadaşları önceden eğitilmiş InceptionResNetv2, Inceptionv3, Xception, Resnet-50, efficientnetB0, Mobilenetv2 modeli ve birde geliştirdikleri generik ESA modeli ile beyin tümörlerini sınıflandırmışlardır (Gomez Guzman et al., 2023). Deneysel çalışmalar sonucunda önceden eğitilmiş Inception V3 modelinin %97,12 ile en yüksek doğruluk başarımını elde ettiğini görmüşlerdir.

Ali ve arkadaşları beyin MRG görüntülerini sınıflandırmak için hibrit bir yöntem önermişlerdir (Ali et al., 2022). Öznitelik çıkarımı için GoogleNet, ShuffleNet ve NasNet-Mobile modellerininin evrişim katmanlarını kullanmışlardır. Çıkarılan öznitelikleri sınıflandırmak için SVM, K-En Yakın Komşu (KNN) ve Lineer Diskriminant Analizi (LDA) sınıflandırıcılarını kullanmışlardır. En yüksek doğruluk başarımını %98,40 ile Shuffle-Net modelinden çıkardıkları öznitelikleri SVM ile sınıflandıran modelden elde etmişlerdir.

Deepak ve Ameer Inception V1 olarak da bilinen önceden eğitilmiş GoogLeNet modelini kullanarak beyin tümörlerini sınıflandırmışlardır (Deepak and Ameer, 2019). Önerdikleri yöntemi üç sınıflı (gliyom, menenjiyom, hipofiz bezi) figshare veri setini

kullanarak test etmişlerdir. Model doğruluğunu tahmin etmek için 5 kat çapraz doğrulama tekniği uygulamışlar ve %98 ortalama doğruluk başarımları elde etmişlerdir.

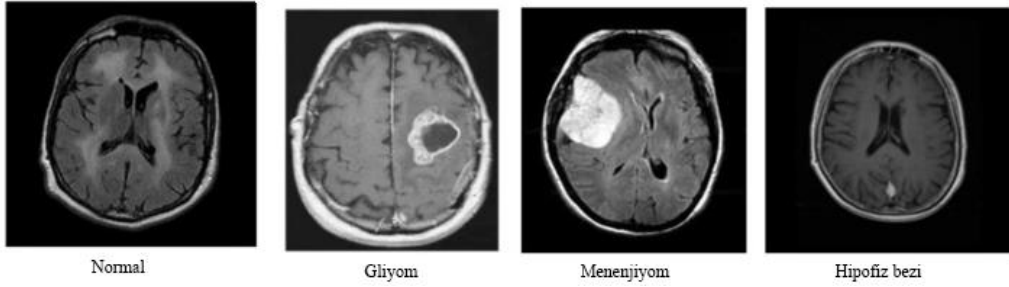
Soumik ve Hossain önceden eğitilmiş Inception V3 modelini kullanarak beyin MRG görüntülerini sınıflandırmışlardır (Soumik and Hossain, 2020). Veri seti olarak figshare veri tabanında bulunan 3 sınıflı 3063 adet beyin MRG görüntüsünü kullanmışlardır. Inception V3 modelininin evrişim katmanlarını kullanarak öznetelik çıkarımı yapmışlar ve softmax sınıflandırıcısını kullanarak beyin MRG görüntülerini sınıflandırmışlardır. Yapılan çalışmada ortalama doğruluk ve fl puanını sırasıyla %99,45 ve %99,35 olarak hesaplamışlardır.

Swati ve arkadaşları figshare veri setini kullanarak önceden eğitilmiş VGG-19 modeli ile beyin MRG görüntülerini sınıflandırmışlardır (Swati et al., 2019). Öznetelik çıkarımı için VGG-19 modelininin evrişim katmanlarını, sınıflandırma için ise softmax sınıflandırıcısını kullanmışlardır. Model doğruluğunu kestirmek için 5 kat çapraz doğrulama yöntemini uygulamışlardır. Deneysel çalışmalarda %94,82 ortalama doğruluk başarımları elde etmişlerdir.

## II. VERİ VE YÖNTEM

### A. Veri Seti

Tez kapsamında kullanılan veri seti Kaggle web sitesinden alınmıştır. Veri setindeki beyin MRG görüntüleri figshare, SARTAJ ve Br35H veri tabanlarının birleşiminden oluşmaktadır (Nickparvar, 2023). Veri setinde toplam 7023 beyin MRG görüntüsü bulunmaktadır. Görüntüler JPG formatında ve 512x512x3 boyutlarındadır. Veri setinde 2000 tane normal beyin MRG görüntüsü bulunmaktadır. Tümörlü beyin MRG görüntülerinin ise 1621 tanesi gliyom, 1645 tanesi menenjiyom ve 1757 tanesi hipofiz bezi türüne aittir. Şekil 2’de normal ve tümörlü beyin MRG görüntüleri yer almaktadır.



Şekil 2. Normal ve tümörlü beyin MRG görüntüleri

### B. Ön İşlem

Bilgisayarlı tomografi aygıtları ile oluşturulan beyin MR görüntülerinde çeşitli gürültüler olabilir. Oluşabilecek gürültüler yanlış yoğunluk düzeylerine neden olur ve sistemin başarımını olumsuz etkiler. Bu nedenle, beyin MR görüntülerinin kalitesini arttırmak için görüntülere histogram eşitleme yöntemi uygulanmıştır. Histogram eşitleme işleminden sonra görüntülere 1 seviyeli ADD ayrıştırması uygulanmış ve görüntü boyutları 256x256x3 boyutlarına düşürülmüştür. Sonrasında, görüntü yoğunluk değerleri Z-skor standardizasyon yöntemi ile ortalama değeri 0 standart sapması 1 olacak şekilde Eşitlik 1’de verildiği gibi standartlaştırılmıştır.

$$z = \frac{x - \mu}{s} \quad (1)$$

Eşitlik 1’de  $x$  eğitim setindeki her bir örneği,  $\mu$  ortalamasını  $s$  ise örneklerin standart sapmasını göstermektedir. Son olarak, Inception v3, 224x24x3 boyutlarındaki girdi katmanına sahip görüntüler için tasarlandığından görüntüler 224x24x3 boyutlarında olacak şekilde yeniden boyutlandırılmıştır. Şekil 3’de ön işlem aşamasında yapılan işlemlerin blok diyagramı verilmiştir.



Şekil 3: Veri ön işlem aşamaları

## 1. Histogram Germe

Histogram germe, bir görüntünün histogramını genişletme yöntemidir. Bu sayede görüntüler daha iyi bir kontrasta sahip olur ve analiz edilmesi kolaylaşır.

$$x_{yeni} = (A - B) \frac{x_{eski} - a}{b - a} + B \quad (2)$$

Denklem 2’de verilen  $a$  değişkeni görüntü içindeki en küçük piksel değerini,  $b$  değişkeni ise en büyük piksel değerini göstermektedir.  $A=255$  ve  $B=0$  seçildiğinde histogram eğrisi 0–255 gri-ton değerleri arasına gerilir.

## C. Dalgacık Dönüşümü

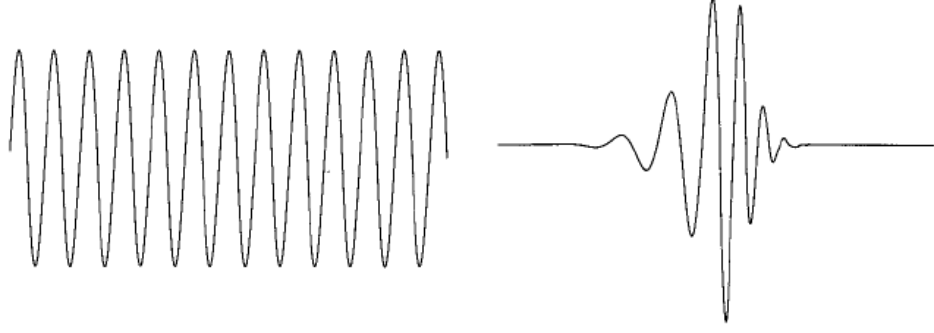
Bir sinyal, bağımsız değişkeni genellikle zaman olan ve zaman içerisinde belirli bir kurala göre veya rasgele değerler alan bir değerler dizisidir. Zamanın her anında bir değere sahip olan sinyallere analog sinyaller denir. Analog sinyaller  $x(t)$  notasyonu kullanılarak gösterilir. Bu notasyonda;  $x$  sinyali,  $t$  bağımsız değişken olan zamanı, yumuşak parantezler ise analog sinyal olduğunu gösterir. Herhangi bir zamanda, sonlu değerlerden en fazla birini alan değerler dizisine sayısal sinyal nedir. Sayısal sinyaller  $x[n]$ ,  $n=...$ ,  $-2, -1, 0, 1, 2, ...$ ;  $n \in \mathbb{Z}$  notasyonu ile gösterilir. Burada;  $x$  sinyali,  $n$  bağımsız değişken olan zamanı, köşeli parantezler ise sayısal sinyal olduğunu gösterir (Gökhan, 2022).

Sinyaller, genellikle zamanın bir fonksiyonu olarak ifade edilmektedir. Zaman uzayındaki bir sinyal matematiksel dönüşümler kullanılarak işlenebilir. Böyle bir dönüşüm sinyal işleme olarak bilinir. Sinyallerin zaman uzayındaki gösteriminde eksenlerden biri genliği diğeri ise zamanı gösterir. Bazı uygulamalarda istenen bilgi frekans bilgisinde saklı olabilir. Bu durumda zaman uzayından frekans uzayına dönüşüm yapmak gerekmektedir. Frekans uzayında eksenlerden biri frekansı temsil ederken diğeri eksen genlik bilgisini içerir.

1950 yılının sonlarından beri kullanılan Fourier dönüşümü, dönüşüm tabanlı görüntü işlemenin temel dayanağı olmuştur. Fourier dönüşümü ile sinyaller zaman uzayından frekans uzayına taşınır. Fakat, dönüşüm sonrasında sinyallerdeki zaman bilgisi kaybolur. Bundan dolayı kısa zaman Fourier dönüşümünün kullanılması gerekmektedir. Bu yöntem, sabit aralıklarla yerleştirilen zaman pencereleri içerisinde uygulanmaktadır. Böylece, bir frekansın mevcut olduğu zaman aralığı hesaplanabilir. Ancak, bir frekansın hangi zaman noktasında meydana geldiği bulunamaz (Bilgin, 2008:28). Fourier dönüşümü, zamana göre frekansı değişmeyen sinyallerin analizinde kullanılan oldukça iyi bir yöntemdir. Ama, durağan olmayan sinyallerin analizinde yeterli değildir.

Görüntüler, genellikle nesnelere oluşturmak için bir araya gelen aynı doku veya yeğlilik seviyelerinin bitişik bölgelerinden oluşur. Eğer bir görüntüdeki nesnelere boyutu küçük ya da kontrastı düşük ise onu yüksek çözünürlükte incelemek isteriz. Görüntüde ki nesnelere büyük boyutlu ve yüksek kontrastlı iseler yüzeysel bir bakış yeterlidir. Hem küçük hem de büyük nesnelere ya da küçük ve yüksek kontrastlı nesnelere bir arada ise onları farklı çözünürlüklerde incelemek gerekebilir. Çok çözünürlük teorisi, görüntülerin birden çok çözünürlükte temsili ve analizi ile ilgilidir. Bu sayede bir çözünürlükte algılanamayan özellikleri bir diğeriinde algılamak kolay olabilir (Gonzales ve Woods, 2014:430).

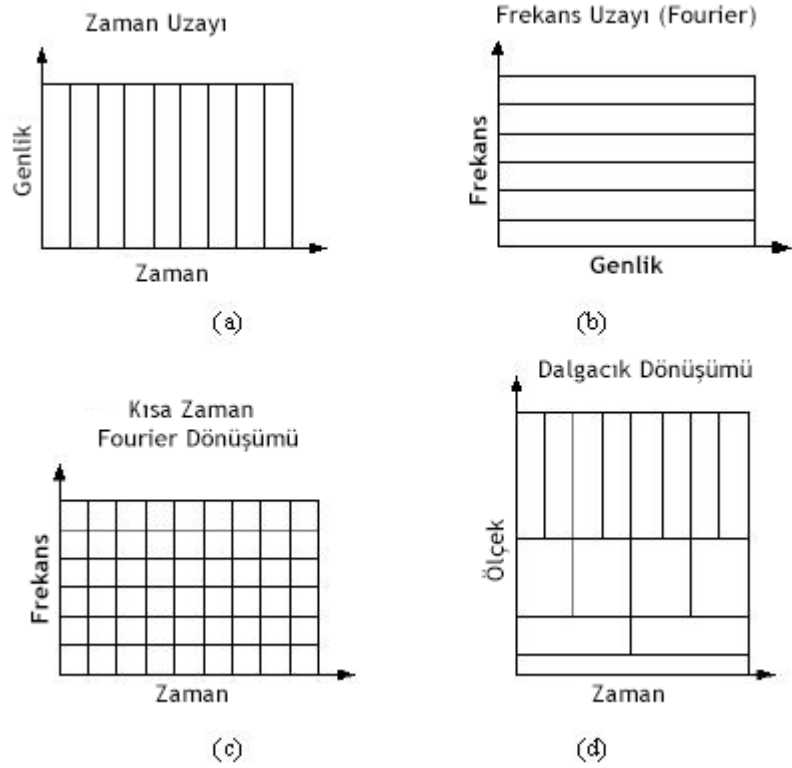
Dalgacık dönüşümleri, dalgacıklar olarak isimlendirilen değişken frekans ve sınırlı süreli küçük dalgalara dayanmaktadır.



Şekil 4: Bir sinüs dalgası ve bir dalgacık (Burrus et al., 1998:15)

Şekil 4'te bir sinüs dalgası ve bir dalgacık arasındaki fark görülmektedir. Bir sinüs dalgası –sonsuzdan +sonsuzya uzanır. Yani, zamanda lokalize değildir. Bir dalgacık ise zamanda lokalizedir. Bu, dalgacık dönüşümünün frekans bilgisine ek olarak zaman bilgisini de elde etmesini sağlar (Siddiqi and Sevindir, 2012:234).

Dalgacık dönüşümü, düşük frekans bilgisinin önem olduğu durumlarda büyük zaman aralıklarını, yüksek frekans bilgisinin önemli olduğu durumlarda küçük zaman aralıklarının kullanıldığı bir yöntemdir. Yani, dalgacık dönüşümü değişik boyutlarda pencerelerin kullanımına izin vermektedir.



Şekil 5: Zaman ve frekans uzayındaki sinyaller

Şekil 5a'da zaman uzayındaki bir sinyalin zamana göre nasıl değiştiğinin grafiği gösterilmektedir. Zaman uzayındaki bu gösterim bir sinyalin analizi için çok fazla anlam taşımaz. Bu nedenle, sinyallerin frekans uzayında analiz edilmesine ihtiyaç duyulmaktadır. Şekil 5b'de bir sinyalin Fourier dönüşümü kullanılarak oluşturulan frekans-genlik grafiği gösterilmektedir. Ancak, dönüşüm işleminde zaman bilgisi kaybolur.

Şekil 5c'de gösterilen, eşit aralıklı pencerelerin kullanıldığı KZFD yönteminde bir sinyalin hangi zaman aralığında meydana geldiği bulunabilir. Ancak, hangi zaman noktasında meydana geldiği tespit edilemez. Bu sorunların çözümü için Şekil 5d'de grafiği gösterilen dalgacık dönüşümü önerilmektedir. Bu yöntem, düşük frekans bilgisinin önem olduğu durumlarda büyük zaman aralıklarının, yüksek frekans bilgisinin önemli olduğu durumlarda ise küçük zaman aralıklarının kullanımına izin vermektedir (Bilgin, 2008:28).

Dalgacık dönüşümü, sürekli dalgacık dönüşümü ve ayrık dalgacık dönüşümü olarak iki durumda incelenebilir.

## 1. Sürekli Dalgacık Dönüşümü

Sürekli dalgacık dönüşümü, tüm zaman aralığı boyunca ana dalgacık fonksiyonu  $\Psi$ 'nin ölçeklenmiş ve ötelenmiş haliyle sinyalin çarpımından meydana gelir (Meral, 2013:37).

$$SDD(s, \tau) = \int_{-\infty}^{+\infty} f(t) \Psi_{s,\tau}^*(t) dt \quad (3)$$

Denklem 3'de verilen sürekli dalgacık dönüşümü  $f(t)$  sinyali ile  $\Psi_{s,t}(t)$ 'nin çarpımı olarak ifade edilir. Burada, \* kompleks eşleniği,  $s$  ölçek parametresini,  $\tau$  öteleme parametresini,  $f(t)$  dönüşümü yapılacak fonksiyonu,  $\Psi_{s,t}(t)$  ise ana dalgacık fonksiyonunu temsil etmektedir. Denklem 3'de,  $f(t)$  sinyali, sürekli dalgacık dönüşümü ile iki parametrelili  $SDD(s, \tau)$  fonksiyonuna dönüştürülür. Dalgacık fonksiyonları bir ana dalgacıktan ( $\Psi$ ), ölçek ve öteleme faktörlerinin kullanılmasıyla elde edilir. Matematiksel ifadesi Denklem 4'te verilmiştir (Arı vd., 2008:24).

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t-\tau}{s}\right) \quad (4)$$



Denklem 4'deki  $\frac{1}{\sqrt{s}}$  faktörü farklı ölçeklere sahip enerji normalizasyon faktörü olarak açıklanır. Denklem 4'ün Denklem 3'de yerine konulmasıyla aşağıdaki ifade elde edilir.

$$SDD(s, \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t) \Psi^* \left( \frac{t - \tau}{s} \right) dt \quad (5)$$

$\Psi_{s,r}(t)$  fonksiyonu Denklem 6'da verilen koşulu yerine getirmek zorundadır. Yani,  $\Psi_{s,r}(t)$  fonksiyonunun meydana getirdiği alan sıfır olmalıdır.

$$\int_{-\infty}^{+\infty} \Psi_{s,r}(t) dt = 0 \quad (6)$$

Sürekli dalgacık dönüşümünde her zaman birimine göre sinyalin frekans bileşenleri değişmektedir.  $\tau$  zaman aralığının konumuna göre değerlendirilir ve bu aralık bütün sinyal boyunca kaydırılır. Böylece, frekans spektrumlarının zaman bağlı bilgileri elde edilmiş olur. Ölçek parametresi olan  $s$ 'nin ifadesi Denklem 7'de verilmiştir.

$$s = \frac{1}{f} \quad (7)$$

Denklem 7'ye göre frekans artışına karşın ölçek parametresi  $s$  küçülmektedir. Ölçekleme bir dalgacığın sıkıştırılması ya da genişletilmesi anlamına gelmektedir. Genişletme ölçekleme faktörü kullanılarak tanımlanır. Daha küçük bir ölçek faktörü daha sıkılaştırılmış bir dalgacık anlamındadır (Arı vd., 2008:25). Ölçekleme ile düşük ölçeklerde yüksek frekans bilgisi, yüksek ölçeklerde ise düşük frekans bilgileri elde edilir.

## 2. Ayrık Dalgacık Dönüşümü

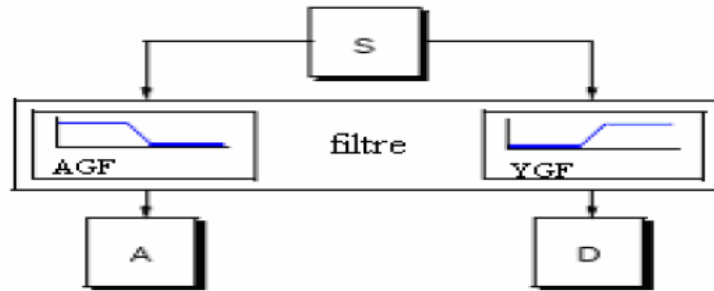
Sürekli dalgacık dönüşümünde tüm ölçek ve öteleme değerlerinde hesaplama yapılmaktadır. Bu ise çok fazla dalgacık katsayısının oluşmasına ve işlem yüküne neden olmaktadır. Daha etkin bir yöntem, dalgacık katsayılarını belirli ölçekler ve zaman dilimlerinde seçerek ayrık dalgacık dönüşümünü kullanmaktır (Küçük ve Ağralıoğlu, 2006:72).

$$\Psi_{s,r}(t) = \frac{1}{\sqrt{s_0^m}} \Psi\left(\frac{t-nr_0s_0^m}{s_0^m}\right) \quad (8)$$

ADD analizi, ikinin kuvveti olarak seçilen ölçekleme ve öteleme parametreleri kullanılarak yapılmaktadır. Denklem 8’de verilen dalgacık fonksiyonunda m, n tam sayı değerler olup, sırasıyla, ölçekleme ve zamanda öteleme parametreleridir.  $s_0 > 1$  olmak koşuluyla  $s_0$  sabit genişleme adımıdır. Genellikle 2 olarak seçilir.  $r_0$  zaman eksenindeki sabit öteleme değeridir. Genellikle 1 olarak alınır. Bu ayrık değerler kullanılarak ayrık dalgacık dönüşümü denklem 9’daki gibi ifade edilebilir.

$$ADD(s, \tau) = \int_{-\infty}^{+\infty} f(t) \frac{1}{2^{s/2}} \Psi\left(\frac{t-\tau 2^s}{2^s}\right) dt \quad (9)$$

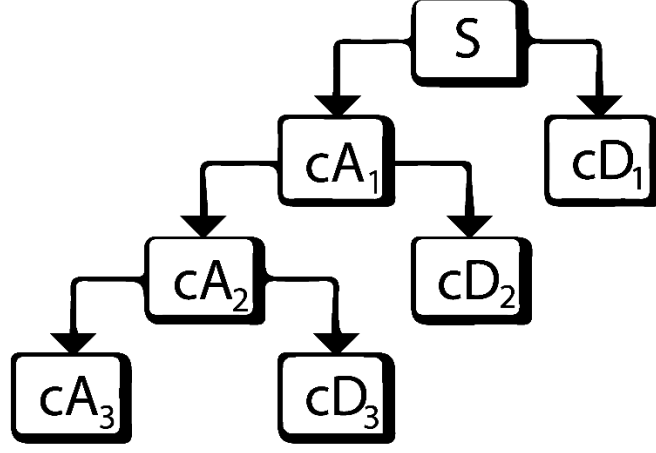
Denklem 9’da s ölçekleme (frekans) parametresi  $\tau$  ise öteleme (zaman) parametresidir. Bu formülü kullanarak ADD’yi gerçekleştirmek oldukça zordur. Bunun yerine kullanılan etkin bir yöntem 1989’da Mallat tarafından filtreler kullanılarak gerçekleştirilmiştir (Mallat, 1989:674). Mallat algoritması olarak bilinen bu yöntemde iki kanal alt bant geçiren bir donanım süzgeci kullanılmaktadır. Şekil 6’da bir sinyale uygulanan filtre ve sonuçları görülmektedir. Temel fikir olarak bir sinyal alçak frekans ve yüksek frekans olmak üzere iki ana bileşene ayrıştırılmaktadır. Bu bileşenlerden ilki düşük frekanslı bileşenler yani alçak geçiren filtre çıkışlarıdır. Diğeri ise yüksek frekanslı bileşenler yani yüksek geçiren filtre çıkışlarıdır.



Şekil 6: S sinyalinin düşük (A) ve yüksek (D) frekans bileşenlerine ayrılması

Filtreleme sonucunda oluşan sinyallerden A sinyaline yaklaşım D sinyaline ise detay adı verilir. Yaklaşımlar yüksek ölçekle elde edilirler ve sinyalin orijinalini verirler. Detaylar ise düşük ölçekle elde edilirler ve sinyalin ayrıntısını temsil ederler. Örneğin, insan sesinden yüksek frekans bileşenleri ortadan kaldırırsa konuşmanın

içeriği anlaşılabilir. Ama, düşük frekanslar ortadan kaldırılırsa anlaşılmayan sesler duyulur.



Şekil 7: Dalgacık ayrıştırma ağacı

Şekil 7’de bir sinyalin ADD kullanılarak alt bileşenlerine ayrıştırılması işleminin blok diyagramı gösterilmektedir. Bu diyagramda c katsayıları ifade etmektedir (Arı vd., 2008:30). Alt bileşenlere ayırma işlemi sinyal üzerinde istenildiği kadar gerçekleştirilebilir. Sinyal, öncelikle yaklaşım ve detay bileşenlerine ayrıştırılır. Sonrasında, elde edilen yaklaşım bileşeni tekrar yaklaşım ve detay bileşenlerine ayrıştırılır. Bu işlem, ardışık olarak istenilen sonuç elde edilene kadar devam ettirilir.

#### D. Ayrık Kosinüs Dönüşümü

Ayrık kosinüs dönüşümü (AKD) 1974 yılında Ahmed ve arkadaşları tarafından tanıtıldı (Ahmed et al., 1974:90). AKD, bir sinyalin kosinüs fonksiyonları olarak frekans düzlemine aktarılma işlemidir. AKD, görüntü sıkıştırma kapasitesi, giriş verilerinden bağımsız olması ve hızlı hesaplama yöntemleri nedeniyle görüntü işlemede yaygın olarak kullanılmaktadır.

AKD, Fourier dönüşümüne benzerdir. Fourier dönüşümünde bulunan karmaşık işlemi ortadan kaldıran ve gerçek sayı düzleminde işlem yapan bir dönüşüm türüdür (Aydoğdu, 2021:54). Bir görüntüyü farklı öneme sahip parçalara ayırmaya yardımcı olur. Bir sinyali veya görüntüyü uzamsal alandan frekans alanına taşır.

8x8 boyutundaki bir görüntü parçası bire bir dönüşüm kullanılarak frekans alanına taşınır. Bu dönüşüm işlemi sonucunda toplam enerjinin büyük kısmı az miktarda dönüşüm katsayısında toplanır. Böylece, görüntünün daha az veriyle ifade edilmesi sağlanır. Dönüşüm katsayıları, başta görüntü sıkıştırma olmak üzere birçok alanda kullanılır (Emek, 2006:46).

Bir boyutlu AKD katsayıları Denklem 10'daki gibi hesaplanmaktadır.

$$F(u) = \sqrt{\frac{2}{N}} \alpha(u) \sum_{x=0}^{N-1} f(x) * \cos\left(\frac{\pi u(2x+1)}{2N}\right), u = 0, 1, \dots, N-1 \quad (10)$$

Denklem 10'da  $f(x)$ , N adet özniteliğe sahip bir boyutlu giriş vektörüdür.  $f(u)$  ise N adet özniteliğe sahip bir boyutlu AKD katsayıları vektörüdür.  $\alpha(u)$  fonksiyonu eşitlik (11)'deki gibi hesaplanmaktadır.

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & \text{aksi takdirde} \end{cases} \quad (11)$$

Dönüşüm sonucunda elde edilen AKD katsayıları alçak ve yüksek frekans bileşenlerinden meydana gelmektedir. AKD katsayı vektörünün birinci katsayısı DC bileşen olarak isimlendirilmektedir ve sinyalin ortalama bilgisini tutmaktadır. Diğer katsayılar ise AC bileşen olarak isimlendirilmektedir. AKD katsayı vektörünün son elemanları, sinyal hakkında detaylı bilgi veren yüksek frekanslı bileşenlerdir (Aydoğdu, 2021:54).

N piksellik bir vektör N tane kosinüs fonksiyonun ağırlıklı toplamıyla gösterilebildiği gibi N x M piksellik bir matris de N x M kosinüs fonksiyonunun ağırlıklı toplamıyla gösterilebilir. İki boyutlu bir görüntü için AKD katsayıları Denklem 12'deki gibi hesaplanmaktadır.

$$F(u,v) = \left(\frac{4}{N.M}\right)^{\frac{1}{2}} c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) * \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2M}\right) \quad (12)$$

Burada, N satır sayısını, M sütun sayısını, f(x,y) ise matrisin ilgili hücre değerini ifade etmektedir. c(u) ve c(v) fonksiyonları Denklem 13'deki gibi değerler almaktadır.

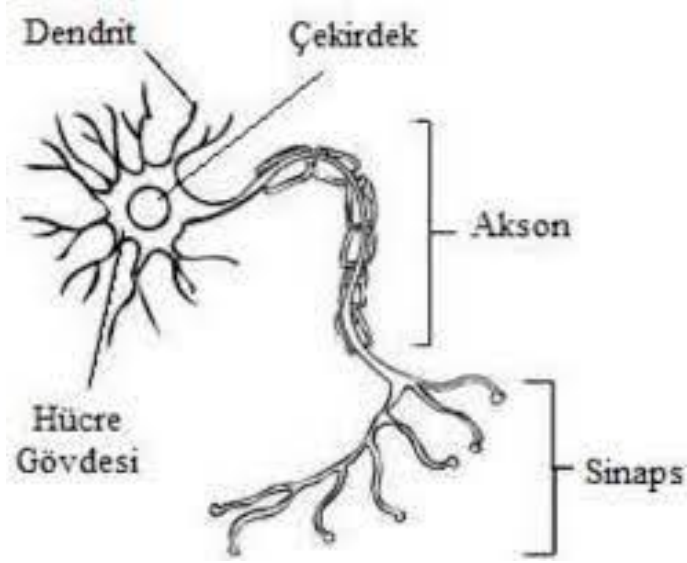
$$c(f) = \begin{cases} \frac{1}{\sqrt{2}} & f = 0 \\ 1 & \text{aksi takdirde} \end{cases} \quad f = u \text{ veya } v \text{ için} \quad (13)$$

## E. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninin öğrenme sürecinden esinlenerek geliştirilmiş bilgisayar programlarıdır. İnsanlarda öğrenme, sinir hücreleri arasındaki sinaptik bağlantıların kurulması ile olur (Ersoy vd., 2012:192). Benzer şekilde, YSA'da da yapay sinir hücreleri arasındaki bağlantı ağırlıklarının ayarlanması ile öğrenme gerçekleşir. YSA, kendisine verilen örnekleri kullanarak bilgiler toplamakta daha sonra hiç karşılaşmadığı örnekler verildiğinde bunlar hakkında karar verebilmektedir.

### 1. Biyolojik Sinir Hücresi

Biyolojik sinir ağları, insan beyinde bulunan, birbirleri ile bağlantılı milyarlarca sinir hücresinden (nöron) meydana gelmektedir. Beyinde, yaklaşık 100 milyar sinir hücresinin olduğu ve her bir sinir hücresinin 50.000 – 250.000 arasında başka sinir hücresi ile bağlantılı olduğu tahmin edilmektedir (Öztemel, 2006:45). Şekil 8'de biyolojik sinir hücresi yapısı gösterilmiştir.

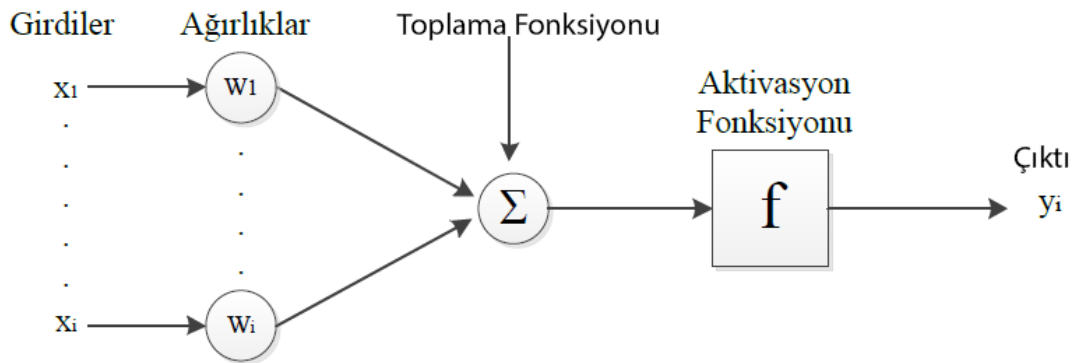


Şekil 8: Biyolojik sinir hücresi

Şekil 8’de gösterildiği gibi, temel bir biyolojik sinir hücresi; hücre gövdesi, akson, dendrit ve sinapstan meydana gelmektedir. Dendritler, diğer sinir hücrelerinden gelen sinyalleri çekirdeğe iletirler. Çekirdek, dendritlerden gelen sinyalleri bir araya getirir ve aksona iletir. Akson, bir araya getirilen bu sinyalleri işleyerek sinapslara iletir. Sinapslar ise elde edilen yeni sinyalleri diğer sinir hücrelerine gönderir (Orhan, 2022).

## 2. Yapay Sinir Hücresi

Yapay sinir ağları (YSA), biyolojik sinir hücrelerinin çalışma şekline esinlenerek geliştirilmiştir. Biyolojik sinir ağları, sinir hücrelerinden meydana gelmektedir. Benzer şekilde yapay sinir ağları da yapay sinir hücrelerinden (proses elemanları) oluşmaktadır. Şekil 9’da yapay sinir hücresi yapısı gösterilmiştir.



Şekil 9: Yapay sinir hücresi (Bülbul, 2022:28)

Şekil 9’da gösterildiği gibi bir yapay sinir hücresinin girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere 5 temel elamanı vardır.

Girdiler, yapay sinir hücresine dış dünyadan veya başka hücrelerden gelen bilgilerdir. Girdiler, ağırlık öğrenmesi için kullanılan örneklerdir.

Ağırlıklar, bir yapay sinir hücresine gelen bilginin hücre üzerindeki etkisini gösterir. Ağırlık değerleri negatif, pozitif veya sıfır olabilir. Ağırlık değerinin sıfır olması girdinin hücre üzerinde bir etkisinin olmadığı anlamına gelir.

Toplama fonksiyonu, yapay sinir hücresine gelen net girdiyi hesaplamak için kullanılır. Toplama fonksiyonu olarak Çizelge 1’de verilen formüllerden biri kullanılabilir. Ama, yaygın olarak ağırlıklı toplamı bularak hesaplanır. Bu yöntemde, hücreye gelen girdi değeri kendi ağırlığı ile çarpılarak toplanır.

$$NET = \sum_{i=1}^n G_i A_i \quad (14)$$

Denklem 14’te G girdileri, A ağırlıkları n ise bir yapay sinir hücresine gelen toplam girdi sayısını göstermektedir. Yapay sinir ağlarında toplama fonksiyonu olarak değişik formüller kullanılabilir. Çizelge 1’de farklı toplama fonksiyonları yer almaktadır. Hangi formülün daha iyi olduğunu belirlemenin bir yöntemi bulunmamaktadır. Tasarımcının, deneme yanılma yöntemiyle uygun toplama fonksiyonunu seçmesi gerekmektedir (Öztemel, 2006:49).

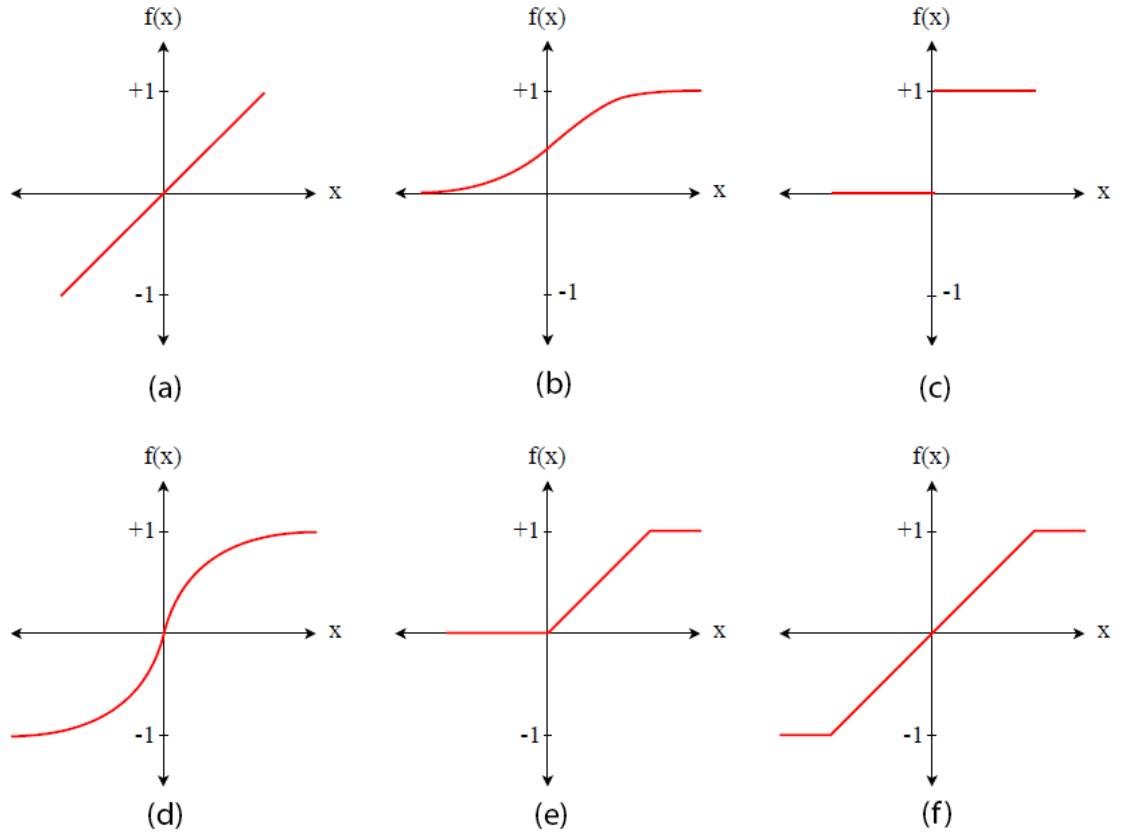
Çizelge 1: Toplama fonksiyonları

Toplama Fonksiyonu:	Yığılma Fonksiyonu:
$NET = \sum_{i=1}^n G_i A_i$	$NET = NET_{eski} \sum_{i=1}^n G_i A_i$
Çoğunluk Fonksiyonu:	Çarpım Fonksiyonu:
$NET = \sum_{i=1}^n sgn(G_i A_i)$	$NET = \prod_{i=1}^n (G_i A_i)$
Minimum Fonksiyonu:	Maksimum Fonksiyonu:
$NET = \min (G_i A_i)$	$NET = \max (G_i A_i)$

Aktivasyon fonksiyonu, yapay sinir hücresine gelen net girdiyi işleyerek bir çıktı üretir. YSA modelinin performansı seçilen aktivasyon fonksiyonuna doğrudan bağlıdır. Aktivasyon fonksiyonu olarak; doğrusal, sigmoid, basamak, hiperbolik tanjant, eşik değer veya rampa fonksiyonlarından biri kullanılabilir. Özellikle, çok katmanlı algılayıcılarda yaygın olarak sigmoid fonksiyonu kullanılmaktadır. [0,1] arasında bir çıktı üreten doğrusal olmayan bir fonksiyondur. Hesaplanması denklem 15’de verilmiştir.

$$F(NET) = \frac{1}{1 + e^{-NET}} \quad (15)$$

Denklem 15’de e doğal logaritmanın tabanı (Euler sayısı) NET ise toplama fonksiyonu tarafından bulunan değerdir. Şekil 10’da yaygın olarak kullanılan aktivasyon fonksiyonu grafikleri verilmiştir.



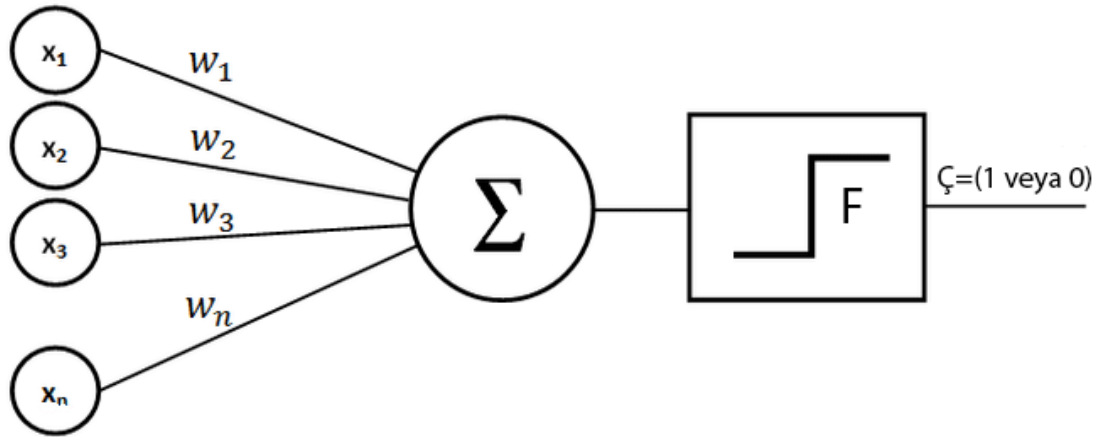
Şekil 10: Aktivasyon fonksiyonları: (a) doğrusal fonksiyon, (b) sigmoid fonksiyonu, (c) basamak fonksiyonu, (d) hiperbolik tanjant fonksiyonu, (e) eşik değeri fonksiyonu, (f) rampa fonksiyonu



Yapay sinir hücresinin çıktısı aktivasyon fonksiyonu tarafından üretilen değerdir. Üretilen çıktı; kendisine, diğer hücelere veya dış dünyaya verilir. Bir hücrenin birden fazla girdisi olmasına rağmen yalnızca bir çıktısı olabilir (Öztemel, 2006:51).

### 3. Basit Algılayıcı Modeli (Perseptron)

Frank Rosenblatt tarafından 1958 yılında tanıtılan perseptron örüntü sınıflandırma amacıyla geliştirilmiştir (Rosenblatt, 1958). Perseptron modeli tek bir yapay sinir hücresinden (proses elemanı) oluşmaktadır. Ağ, birden fazla girdi alarak bir veya sıfırdan oluşan tek bir çıktı üretmektedir. Şekil 11’de basit algılayıcı yapısı gösterilmiştir.



Şekil 11: Basit algılayıcı yapısı

Şekil 11’de gösterildiği gibi bir perseptron; girdi, ağırlıklar, toplama fonksiyonu ve aktivasyon fonksiyonundan oluşmaktadır. Girdi birimleri ağırlıkları ile çarpılarak toplama fonksiyonuna verilir. Aktivasyon fonksiyonu olarak bir veya sıfır değerlerinden birini üreten eşik değer fonksiyonu kullanılır. Örneğin, perseptron girdilerinin ağırlıklı toplamı 0.55 ve eşik değeri 0 ise ağırlıklı toplam 0’dan büyük olduğu için çıktı 1 olacaktır.

Basit algılayıcı modeli eğitilebilir ilk yapay sinir ağı modelidir. Ağa girdi değerleri ve bu değerlere karşılık gelen ağırlıklar verilerek öğrenme kuralına göre çıktı değeri hesaplanır. Hesaplanan çıktı değeri olması gereken değere ulaşana kadar ağırlıklar ve eşik değerleri güncellenir (Öztemel, 2006:62)

Basit algılayıcı modelin eğitilmesi için kullanılan algoritma aşağıda verilmiştir.

**Adım 1:** Veri setindeki her bir örneğe ait öznitelikler ile ona karşılık gelen etiketler ağı gösterilir. Öznitelikler herhangi bir değerde olabilir. Etiketler ise kullanılan aktivasyon fonksiyonu nedeniyle 1 veya 0 değerlerinden birini alabilir.

**Adım 2:** Yapay sinir hücresine gelen net girdi hesaplanır. Formülü Denklem 16'da verilmiştir.

$$NET = \sum_{i=1}^n w_i x_i \quad (16)$$

**Adım 3:** Yapay sinir hücresinin çıktısı hesaplanır. Net girdi eşik değerinden büyükse 1 küçükse veya eşitse 0 değerini alır.

$$\zeta = \begin{cases} 1 & \text{Eğer } NET > \phi \\ 0 & \text{Eğer } NET \leq \phi \end{cases} \quad (17)$$

Yapay sinir hücresinin ürettiği çıktı ile hedeflenen çıktı aynı olursa ağırlıklarda bir değişiklik yapılmaz. Eğer üretilen çıktı ile hedeflenen çıktı farklı ise iki durum ortaya çıkar:

- a) Yapay sinir ağının ürettiği çıktı 1 hedeflenen çıktı 0 olabilir. Bu durumda ağırlık değerleri azaltılır. Yeni ağırlık değerleri denklem 18'de verildiği şekilde hesaplanmaktadır.

$$W_n = W_o - \lambda X \quad (18)$$

Denklem 18'de  $\lambda$  öğrenme katsayısını,  $X$  girdileri,  $W_o$  eski ağırlık değerlerini,  $W_n$  ise yeni ağırlık değerlerini göstermektedir. Öğrenme katsayısı, yanlış tahmin edilen çıktıdan sonra modelin ne kadar değiştirileceğini belirleyen 0 ile 1 arasında sabit bir değerdir.

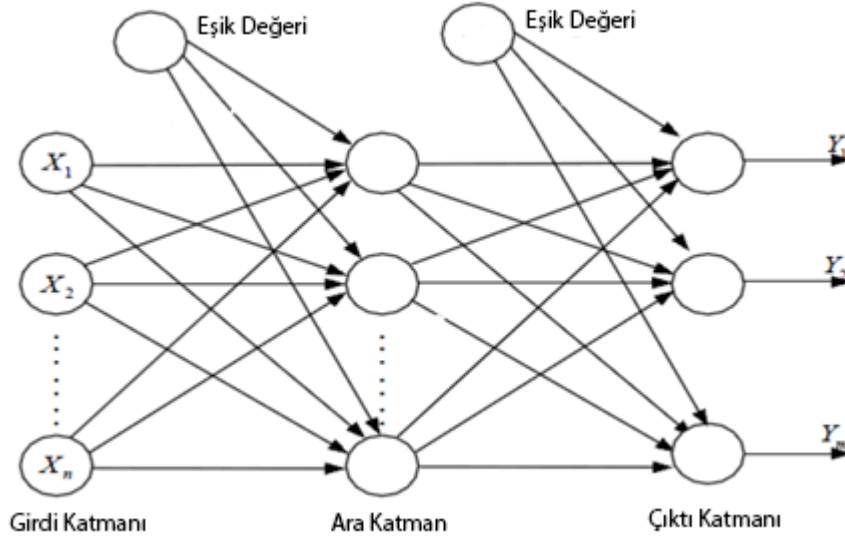
- b) Yapay sinir ağının ürettiği çıktı 0 hedeflenen çıktı 1 olabilir. Bu durumda ağırlık değerleri artırılır. Yeni ağırlık değerleri Denklem 19'da verildiği şekilde hesaplanmaktadır.

$$W_n = W_o + \lambda X \quad (19)$$

**Adım 4:** Eğitim setindeki tüm örnekler için ağı ürettiği çıktı ile hedeflenen çıktı aynı olana kadar yukarıdaki ilk üç adım tekrarlanır.

#### 4. Çok Katmanlı Algılayıcı

Tek katmanlı algılayıcılarla girdinin sadece doğrusal işlevleri öğrenebilir. Dışlayıcı VEYA (XOR) gibi doğrusal olmayan problemler gerçekleştirilemez (Alpaydın, 2013:207). Tek katmanlı algılayıcıların dışlayıcı VEYA problemini çözmemesi yapay sinir ağı alanında yapılan çalışmaların durmasına neden olmuştur. Bu sorun çok katmanlı algılayıcıların (ÇKA) kullanılmasıyla çözülmüştür. Bunun için girdi katmanını doğrudan çıktı katmanına bağlamak yerine bunların arasına ara katman konularak bir çözüm getirilmiştir (Elmas, 2010:46). Şekil 12’de ÇKA ağlarının yapısı verilmiştir.



Şekil 12: Çok katmanlı algılayıcı yapısı (Kargı, 2013:66)

Şekil 12’de görüldüğü gibi ÇKA ileriye doğru bağlantılı girdi katmanı, ara katmanlar (gizli katmanlar) ve çıktı katmanından oluşmaktadır. Bilgiler girdi katmanından ağa verilir, ara katmanlardaki proses elemanlarında işlenerek çıktı katmanına iletilir.

Girdi katmanı eğitim setindeki her bir örneğin özniteliklerinin giriş olarak verildiği katmandır. Bu katmandaki sinir hücrelerinde herhangi bir işlem yapılmaz gelen bilgi olduğu gibi bir sonraki katmana aktarılır. Girdi katmandaki her sinir hücresinin tek girdisi ve tek çıktısı olmaktadır. Sinir hücresinin çıktısı bir sonraki ara katmandaki bütün sinir hücrelerine iletilmektedir.

Ara katmanlar girdi katmanından gelen bilgileri işleyerek çıktı katmanına iletir. Gizli katmanlar olarak da isimlendirilen ara katmanlar bir ÇKA ağında birden fazla sayıda olabilir. Ayrıca, her ara katmanda birden fazla sayıda sinir hücresi bulunabilir. Bu katmanlardaki her bir sinir hücresi bir sonraki katmanda bulunan tüm sinir hücrelerine bağlıdır. Ara katmanlar bir anlamda ağın zeka içeren kısmını oluşturur. Bu katmandaki sinir hücresi sayısı fazla olursa ağ ezberlemeye yönelir. Uygulamalarda az sayıda sinir hücresi (2 veya 3) ile başlayıp sonra bu sayıyı arttırmakla (örneğin 5, 10, 15) başarımlarını değerlendirmesi yapılması önerilmektedir (Nabiyev, 2012:585).

Çıktı katmanı ara katmandan gelen bilgilerin işlenerek etiket bilgisinin üretildiği katmandır. Çıktı katmanında sinir hücresi sayısı uygulamanın ihtiyacına göre istenildiği kadar olabilir. Bu katmandaki sinir hücreleri kendinden önce gelen ara katmandaki bütün sinir hücrelerine bağlıdır.

ÇKA ağları gözetimli öğrenme stratejisini kullanmaktadır. ÇKA ağlarına eğitim sırasında eğitim setinin öznitelikleri girdi olarak verilmekte ve bu özniteliklere karşılık üretilen çıktıların etiket değerlerine yaklaşması sağlanmaktadır. Bu ağlarda öğrenme kuralı olarak en küçük kareler yöntemine dayanan Genelleştirilmiş Delta Kuralı kullanılmaktadır. Bir ÇKA ağının eğitilmesi ileriye doğru hesaplama ve geriye doğru hesaplama aşamalarından oluşmaktadır.

**İleriye Doğru Hesaplama:** Bu aşamada, yapay sinir ağına gösterilen öznitelikler için çıktılar hesaplanır ve etiket değerleri kullanılarak yapılan hata bulunur. Eğitim setindeki öznitelikler herhangi bir bilgi işleme yapılmadan doğrudan girdi katmanındaki sinir hücrelerine verilir. Ara katmandaki sinir hücreleri girdi katmanındaki sinir hücrelerinin değerlerini ağırlıkları ile çarparak kullanılan ağırlıklı toplam fonksiyonuna göre NET girdi olarak alır (Öğücü, 2006:43). Ara katmandaki her bir sinir hücresinin NET girdisinin hesaplanması Denklem 20’de verilmiştir.

$$NET_{ap} = \sum_i w_{gi,ap} x_i \quad (20)$$

Denklem 20’de  $NET_{ap}$  ara katmandaki sinir hücresinin net girdisini,  $w_{gi,ap}$  girdi katmanındaki  $i$ . sinir hücresini ara katmandaki  $p$ . sinir hücresine bağlayan bağlantının ağırlık değerini,  $x_i$  ise girdi katmanındaki  $i$ . sinir hücresinin çıktısını göstermektedir.

Ara katmandaki sinir hücrelerinin çıktısı net girdinin aktivasyon fonksiyonundan geçirilmesiyle bulunur. Aktivasyon fonksiyonu olarak genellikle sigmoid, ReLU ve hiperbolik tanjant fonksiyonları kullanılmaktadır. Geriye doğru hesaplama aşamasında burada kullanılan aktivasyon fonksiyonunun türevi alınacağından türevlenebilir bir fonksiyon seçmek gerekmektedir. Sigmoid fonksiyonu kullanılması durumunda sinir hücresinin çıktısı Denklem 21’de verildi gibi hesaplanmaktadır.

$$F_{ap} = \frac{1}{1 + e^{-(NET_{ap} + \beta_{ap})}} \quad (21)$$

Denklem 21’de  $NET_{ap}$  ara katmandaki p. sinir hücresinin girdisini göstermektedir.  $\beta_{ap}$  ara katmandaki p. sinir hücresine bağlanan eşit değer elemanın ağırlığıdır. Eşik değer ünitesinin çıktısı 1’e eşittir.  $F_{ap}$  ise ara katmandaki p. sinir hücresinin sigmoid fonksiyonundan geçirildikten sonraki çıktısıdır.

Ara katman ile çıktı katmanı arasında da benzer işlemler gerçekleştirilir. Çıktı katmanındaki sinir hücrelerinin NET girdisi Denklem 22’deki gibi hesaplanmaktadır.

$$NET_{cq} = \sum_i w_{ap,cq} F_{ap} \quad (22)$$

Burada  $NET_{cq}$  çıktı katmanındaki q. sinir hücresinin girdisi göstermektedir.  $w_{ap,cq}$  ara katmandaki p. sinir hücresini çıktı katmanındaki q. sinir hücresine bağlayan bağlantının ağırlık değerini göstermektedir.  $F_{ap}$  ise ara katmandaki p. sinir hücresinin çıktısını göstermektedir. Çıktı katmanındaki sinir hücrelerinin çıktısı aktivasyon fonksiyonundan geçirildikten sonra bulunur.

$$F_{cq} = \frac{1}{1 + e^{-(NET_{cq} + \beta_{cp})}} \quad (23)$$

Denklem 23’te  $NET_{cq}$  çıktı katmanındaki q. sinir hücresinin NET girdisini,  $\beta_{cp}$  çıktı katmandaki p. sinir hücresine bağlanan eşit değer elemanın ağırlık değerini,  $F_{cq}$  ise çıktı katmanındaki q. sinir hücresinin sigmoid fonksiyonundan geçirildikten sonraki değerini göstermektedir. Çıktı katmanındaki sinir hücrelerinin çıktılarının bulunmasıyla ileriye doğru hesaplama aşaması tamamlanmaktadır (Öztemel, 2006:78).

**Geriye Doğru Hesaplama:** Bu aşamada ağı verilen örnekler için ağıın ürettiği çıktı ile hedeflenen çıktı arasındaki fark hesaplanır. Amaç, hata olarak kabul edilen bu farkın azaltılmasıdır. Bu amaçla, bulunan hata geriye doğru yayılarak ağırlık değerlerine dağıtılır ve bir sonraki iterasyonda hatanın düşmesi sağlanır. Hata geriye doğru yayıldığından önce çıktı katmanı ile ara katman arasındaki ağırlık değerleri sonrasında ise ara katman ile girdi katmanı arasındaki ağırlık değerleri değiştirilir.

Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi: Çıktı katmanındaki  $q$ . sinir hücresi için yapılan karesel hata Denklem 24'te verildiği gibi hesaplanmaktadır.

$$\varepsilon_q^2 = (T_q - F_{cq})^2 \quad (24)$$

Burada,  $T_q$  çıktı katmanındaki  $q$ . sinir hücresinin hedeflenen çıktısını  $F_{cq}$  ise çıktı katmanındaki  $q$ . sinir hücresinin ürettiği çıktıyı göstermektedir. Hatanın negatif değerler almasını önlemek için her iki tarafın karesi alınmıştır.

Delta öğrenme kuralına göre hatanın minimum olması için ağırlıklar hatanın negatif gradyanı yönünde değiştirilmelidir. Ara katman ile çıktı katmanı arasındaki ağırlıkların değişimi Denklem 25'te verildiği şekilde hesaplanmaktadır.

$$\nabla w_{ap,cq} = -\eta \frac{\partial \varepsilon_q^2}{\partial w_{ap,cq}} \quad (25)$$

Denklem 25'te  $\eta$  öğrenme oranını,  $w_{ap,cq}$  ise ara katmandaki  $p$ . sinir hücresini çıktı katmanındaki  $q$ . sinir hücresine bağlayan bağlantının ağırlık değerini göstermektedir. Gradyan karesel hatanın ağırlıklara göre kısmi türevi alınarak bulunur (Yavuz, 2021:37). Denklem 26'da verildiği gibi zincir kuralı uygulanarak hesaplanır.

$$\frac{\partial \varepsilon_q^2}{\partial w_{ap,cq}} = \frac{\partial \varepsilon_q^2}{\partial F_{cq}} \cdot \frac{\partial F_{cq}}{\partial NET_{cq}} \cdot \frac{\partial NET_{cq}}{\partial w_{ap,cq}} \quad (26)$$

$$\frac{\partial \varepsilon_q^2}{\partial F_{cq}} = -2(T_q - F_{cq}) \quad (27)$$

$$\frac{\partial F_{cq}}{\partial NET_{cq}} = F_{cq}(1 - F_{cq}) \quad (28)$$

$$\frac{\partial NET_{cq}}{\partial w_{ap,cq}} = F_{ap} \quad (29)$$

Hesaplanan terimler Denklem 30'da yerine yazılırsa hata fonksiyonunun  $w_{ap,cq}$  ağırlığına göre kısmi türevi bulunmuş olur.

$$\frac{\partial \varepsilon_q^2}{\partial w_{ap,cq}} = -2(T_q - F_{cq}) \cdot F_{cq} (1 - F_{cq}) \cdot F_{ap} \quad (30)$$

Böylece, ara katman ile çıktı katmanı arasında herhangi bir iterasyonda ağırlıkların değişim miktarı bulunmuş olur.

$$\begin{aligned} \nabla w_{ap,cp} &= -\eta \frac{\partial \varepsilon_q^2}{\partial w_{ap,cq}} \\ &= -\eta (T_q - F_{cq}) \cdot F_{cq} (1 - F_{cq}) \cdot F_{ap} \end{aligned} \quad (31)$$

$$w_{ap,cp}(T) = w_{ap,cp}(T - 1) + \nabla w_{ap,cp} \quad (32)$$

Girdi katmanı ile ara katman arasındaki ağırlıkların değişimi: Ara katman ve çıktı katmanı arasındaki ağırlıklar değiştirilirken yalnızca çıktı katmanındaki bir sinir hücresinin hatası dikkate alınmaktadır. Bu hatalar hesaplanırken girdi katmanı ile ara katman arasındaki ağırlıklar da göz önünde tutulmaktadır. Çünkü, en son ara katmana gelen bilgiler girdi katmanı veya önceki ara katmandan gelmektedir. Bu nedenle girdi katmanı ile ara katman arasındaki ağırlıklar değiştirilirken çıktı katmanındaki sinir hücrelerinin tümünün hatası dikkate alınmaktadır. (Öztemel, 2006:80).

$$\varepsilon^2 = \sum_{q=1}^n \varepsilon_q^2 \quad (33)$$

Girdi katmanı ile ara katman arasındaki ağırlıkların değişimi Denklem 34'de verildiği şekilde hesaplanmaktadır.

$$\nabla w_{gh,ap} = -\eta \sum_{q=1}^n \frac{\partial \varepsilon_q^2}{\partial w_{gh,ap}} \quad (34)$$

Denklem 34'de  $\eta$  öğrenme oranını,  $w_{gh,ap}$  ise girdi katmanındaki h. sinir hücresini ara katmandaki p. sinir hücresine bağlayan bağlantının ağırlık değerini

göstermektedir. Denklem 35’de verildiği gibi gradyanı hesaplamak için zincir kuralını uygulayalım.

$$\frac{\partial \varepsilon_q^2}{\partial w_{gh,ap}} = \sum_{q=1}^n \frac{\partial \varepsilon_q^2}{\partial F_{cq}} \cdot \frac{\partial F_{cq}}{\partial NET_{cq}} \cdot \frac{\partial NET_{cq}}{\partial F_{ap}} \cdot \frac{\partial F_{ap}}{\partial NET_{ap}} \cdot \frac{\partial NET_{ap}}{\partial w_{gh,ap}} \quad (35)$$

$$\frac{\partial \varepsilon_q^2}{\partial F_{cq}} = -2(T_q - F_{cq}) \quad (36)$$

$$\frac{\partial F_{cq}}{\partial NET_{cq}} = F_{cq}(1 - F_{cq}) \quad (37)$$

$$\frac{\partial NET_{cq}}{\partial F_{ap}} = w_{ap,cq} \quad (38)$$

$$\frac{\partial F_{ap}}{\partial NET_{ap}} = F_{ap}(1 - F_{ap}) \quad (39)$$

$$\frac{\partial NET_{ap}}{\partial w_{gh,ap}} = x_h \quad (40)$$

Hesaplanan terimler Denklem 41’de yerine yazılırsa hata fonksiyonunun  $w_{gh,ap}$  ağırlığına göre kısmi türevi bulunmuş olur.

$$\frac{\partial \varepsilon_q^2}{\partial w_{gh,ap}} = \sum_{q=1}^n -2(T_q - F_{cq}) \cdot F_{cq}(1 - F_{cq}) \cdot w_{ap,cq} \cdot F_{ap}(-F_{ap}) \cdot x_h \quad (41)$$

Böylece, girdi katmanı ile ara katman arasında herhangi bir iterasyonda ağırlıkların değişim miktarı bulunmuş olur.

$$\nabla w_{gh,ap} = -\eta \sum_{q=1}^n 2(T_q - F_{cq}) \cdot F_{cq}(1 - F_{cq}) \cdot w_{ap,cq} \cdot F_{ap}(-F_{ap}) \cdot x_h \quad (42)$$

$$w_{gh,ap}(T) = w_{gh,ap}(T - 1) + \nabla w_{gh,ap} \quad (43)$$

Sonuçta, sinir ağındaki bütün ağırlıklar değiştirilmiş olur. Hem ileri hem de geriye doğru hesaplama yapılarak herhangi bir iterasyon tamamlanmış olur.



Sonrasında, ağa başka bir örnek verilerek bir sonraki iterasyona geçilir. Bu işlemler öğrenme gerçekleşene kadar tekrarlanır. (Öztemel, 2006:80)

## F. Sınıflandırma Ölçüm Metrikleri

Sınıflandırma, etiketlenmemiş bir örneği, otomatik olarak, birbirinden farklı ve önceden belirlenmiş sınıflara atama problemidir. Burada, etiket, sınıflar adlı bir kümenin elemanıdır. Sınıflar kümesi iki elemanlı ise ikili sınıflandırma (hasta, sağlıklı), daha fazla elemandan oluşuyorsa çok sınıflı sınıflandırmadan söz edilir. Yüz tanıma, ses tanıma, nesne tanıma ve span tespiti gibi birçok makine öğrenmesi problemi sınıflandırma öğrenme algoritması ile çözülür (Okatan vd., 2021:17).

Bir sınıflandırma öğrenme algoritmasını uyguladıktan sonra modelin ne kadar iyi olduğu sınıflandırma metriklerine göre değerlendirilir. Birçok sınıflandırma metriği bulunmasına rağmen bir modelin performansını ölçmek için hangi metriğin kullanılacağına dair genel bir yöntem bulunmamaktadır (Akgül, 2022:39).

### 1. Doğruluk Oranı

Sınıflandırma problemlerinde, genellikle, performans ölçütü olarak doğruluk oranı (accuracy rate) kullanılmaktadır. Doğruluk değeri, modelde doğru tahmin edilen örnek sayısının toplam örnek sayısına oranı ile bulunmaktadır. Denklem 44'deki gibi hesaplanmaktadır.

$$\text{Doğruluk} = \frac{\text{Doğru tahmin edilen örnek sayısı}}{\text{Toplam örnek sayısı}} \quad (44)$$

Dengesiz veri setlerinde doğruluk oranı yanıltıcı olabilir. Sınıflardan birinde çok fazla örnek diğerinde ise çok az örnek varsa bu problem ortaya çıkabilir. Örneğin, 990 örnekli A sınıfı ve 10 örnekli B sınıfı olsun. Sınıflandırma algoritması tüm örnekleri A olarak sınıflandırsa bile doğruluk oranı  $990/1000 = \%99$  olacaktır. Bu örnekte, doğruluk oranı yüksek olmasına rağmen, aslında, sınıflandırma algoritmasının performansı çok kötüdür.

## 2. Karışıklık Matrisi

Karışıklık matrisi (confusion matrix), sınıflandırma modelinin, tahmin ettiği sınıflar ile gerçek sınıf bilgilerinin yer aldığı tablodur (Cihan vd., 2019:387). Şekil 13’de ikili sınıflandırma için karışıklık matrisi verilmiştir.

		Gerçek Sınıflar	
		Pozitif	Negatif
Tahmin Edilen Sınıflar	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 13: İkili sınıflandırma için karışıklık matrisi

Şekil 13’de görüldüğü üzere, ikili sınıflandırma karışıklık matrisinde, pozitif ve negatif olarak isimlendirilen iki sınıf vardır. Örneğin, tümörlü ve normal hücrelerin yer aldığı bir veri setinde tümörlü hücreler pozitif, normal hücreler ise negatif olsun.

- **TP (True Positive):** Pozitif olarak sınıflandırılan bir örnek gerçekten de pozitifse buna true positive denir. Örneğin, sınıflandırma modelinin tümörlü bir hücreyi tümörlü olarak sınıflandırması.
- **TN (True Negative):** Negatif sınıflandırılan bir örnek gerçekten de negatifse buna true negative denir. Örneğin, sınıflandırma modelinin normal bir hücreyi normal olarak sınıflandırması.
- **FP (False Positive):** Pozitif sınıflandırılan bir örnek gerçekte negatif ise buna false positive denir. Örneğin, sınıflandırma modelinin normal bir hücreyi tümörlü olarak sınıflandırması.
- **FN (False Negative):** Negatif olarak sınıflandırılan bir örnek gerçekte pozitif ise buna false negative denir. Örneğin, sınıflandırma modelinin tümörlü bir hücreyi normal olarak sınıflandırması.

### 3. F1-Puanı

Özellikle dengesiz veri setlerinde, bir modelin başarısını ölçmek için tek başına doğruluk (accuracy) metriği yetersiz kalmaktadır. Eşitlik 45'te doğruluk değerinin hesaplanması verilmiştir.

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (45)$$

Doğruluk metriği güvenilir olmadığından bir modelin başarısını değerlendirmek için F1-puanı (F1-score) yaygın olarak kullanılmaktadır. F1-puanı, kesinlik (precision) ve duyarlılık (recall) değerleri hesaplandıktan sonra bulunur.

Kesinlik, sınıflandırıcının doğru tahmin ettiği pozitif örneklerin tahmin edilen pozitif örneklere oranıdır. Kesinlik değeri ne kadar yüksekse sınıflandırıcı o kadar başarılıdır. Eğer, kesinlik değeri küçükse çok sayıda false positive (FP) var demektir. Matematiksel gösterimi Eşitlik 46'da verilmiştir.

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (46)$$

Duyarlılık (recall), sınıflandırıcının pozitif olarak tahmin etmesi gereken örneklerin ne kadarını pozitif olarak tahmin ettiğini ifade eder. Duyarlılık değeri ne kadar yüksekse sınıflandırıcı o kadar başarılıdır. Eğer, duyarlılık değeri küçükse çok sayıda false negative (FN) var demektir. Matematiksel gösterimi Eşitlik 47'de verilmiştir.

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (47)$$

Yüksek duyarlılık ve düşük kesinlik veya tam tersi durumda sınıflandırma modellerini karşılaştırmak zordur. F1-puanı, duyarlılık ve kesinliğin aynı anda ölçülmesini imkan tanır. Duyarlılık ve kesinliğin harmonik ortalaması alınarak hesaplanır. Matematiksel gösterimi Eşitlik 48'de verilmiştir.

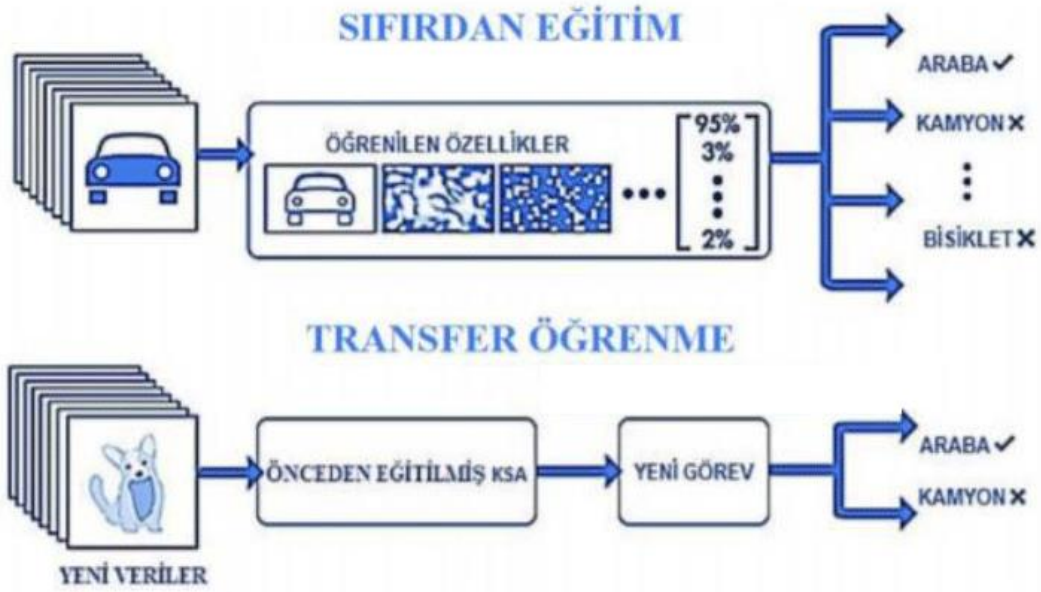
$$\text{F1 - puanı} = \frac{2 * \text{duyarlılık} * \text{kesinlik}}{\text{duyarlılık} + \text{kesinlik}} \quad (48)$$

## G. Transfer Öğrenme

Klasik makine öğrenmesi yöntemleri, bir problemi çözerken geçmiş bilgi ve deneyimlerden yararlanmamaktadır. Her problemi ayrı ayrı değerlendirip, sıfırdan çözümler bulmaya çalışmaktadır. İnsanlarda öğrenme ise geçmiş bilgi ve deneyimlerden yararlanarak gerçekleştirilir (Koçer, 2012:10). Örneğin, bisiklet sürmeyi öğrenen bir kişi, elde ettiği bu bilgi ve deneyimi motorsiklet sürmeyi öğrenirken kullanabilir. Transfer öğrenme (Transfer learning), bir makine öğrenmesi problemini çözerken elde edilen bilgiyi benzer başka problemlerin çözümünde kullanmaktır. Örneğin, otomobilleri tanımayı öğrenirken elde edilen bilgi otobüsleri tanıma probleminde kullanılabilir.

Klasik makine öğrenmesi yöntemlerinde görüntülerden manuel olarak çıkarılan özellikler ile sınıflandırma yapılmaktadır. Evrişimli sinir ağlarında ise evrişim ve havuzlama katmanları tarafından otomatik olarak çıkarılan özellikler kullanılarak sınıflandırma yapılmaktadır. Bir ESA modelinde, gerekli özelliklerin çıkarılabilmesi için büyük veri setlerine ihtiyaç duyulmaktadır. Eğer, eğitim veri seti yeterince büyük değilse aşırı öğrenme (overfitting) meydana gelebilir. Bu sorun, transfer öğrenme yöntemi kullanılarak çözülmektedir (Akgül, 2022:45).

ESA modellerinin eğitimi için büyük veri setlerine ihtiyaç duyulmaktadır. Ancak, bu veri setlerini elde etmek pahalı, emek yoğun ve zaman alıcı olabilir. Ayrıca, modelin eğitimi için yüksek işlem kaynağı gerekmektedir. Eğitim veri setlerinin oluşturulmasının maliyetli olduğu durumlarda transfer öğrenme yaygın olarak kullanılmaktadır. Böylece, daha az maliyetle daha yüksek başarımler elde edilmektedir. Şekil 21’de transfer öğrenme diyagramı üzerinde gösterilmiştir.



Şekil 14: Transfer öğrenme diyagram gösterimi (Kılıç, 2021:60)

Şekil 14’te görüldüğü gibi bir sınıflandırma için belirli miktarda veri seti ile ESA eğitilmektedir. Transfer öğrenmede ise tanımlı nesnelere için ağırlık eğitilmesine ihtiyaç yoktur. Uygulamanın ihtiyacına göre, önceden eğitilmiş modelin belirli katmanları değiştirilerek ağırlıklar ve bias güncellenir. Veri setleri birbirine benzer ise transfer öğrenme aşağıdaki yöntemler kullanılarak uygulanabilir (Kılıç, 2021:60).

**Özellik Çıkarıcı Metot:** Daha önceden eğitilmiş bir ESA modelinin evrişim katmanından elde edilen ağırlık değerleri mevcut modele aktarılır. Sonrasında, sadece sınıflandırma katmanı kullanılarak yeni modelin eğitimi yapılır. Böylece, daha önceden eğitilmiş modelin özellik çıkarma kabiliyeti yeni modele aktarılmış olur. Sınıflandırma katmanında yapay sinir ağları kullanılabileceği gibi istenirse destek vektör makineleri gibi diğer doğrusal sınıflandırıcılar da kullanılabilir (Fırıldak vd., 2019:90).

**Kısmi Özellik Çıkarıcı Metot:** Bu yöntemde, önceden eğitilmiş modelin evrişim katmanındaki ağırlıkların tümü yerine belirli bir kısmı yeni modele aktarılır. Boş kalan evrişim katmanı ve sınıflandırma katmanı probleme göre düzenlenir. Sonrasında, yeni eğitim seti kullanılarak ağırlık eğitimi yapılır.

**Ön Eğitim Modeli:** Bu yöntemde, büyük veri setleri ile eğitilen ve başarımı kanıtlanmış bir modelin ağırlıkları yeni modele aktarılır. Günümüzde, yaygın olarak ön eğitim modeli yaklaşımı kullanılmaktadır. Inception v3 gibi çok katmanlı yapıya sahip

ağların imagenet gibi büyük veri setleri ile eğitimi GPU destekli bilgisayarlarda bile günlerce sürmektedir. Böyle büyük ağlar eğitildikten sonra ağırlık değerleri açık erişime sunulmaktadır. Bir ağın eğitimine rastgele ağırlıklarla başlamak yerine eğitilmiş bir ağın ağırlıklarını kullanarak başlamak daha iyi sonuçlar vermektedir (Kılıç, 2021:63).

### III. DERİN ÖĞRENME

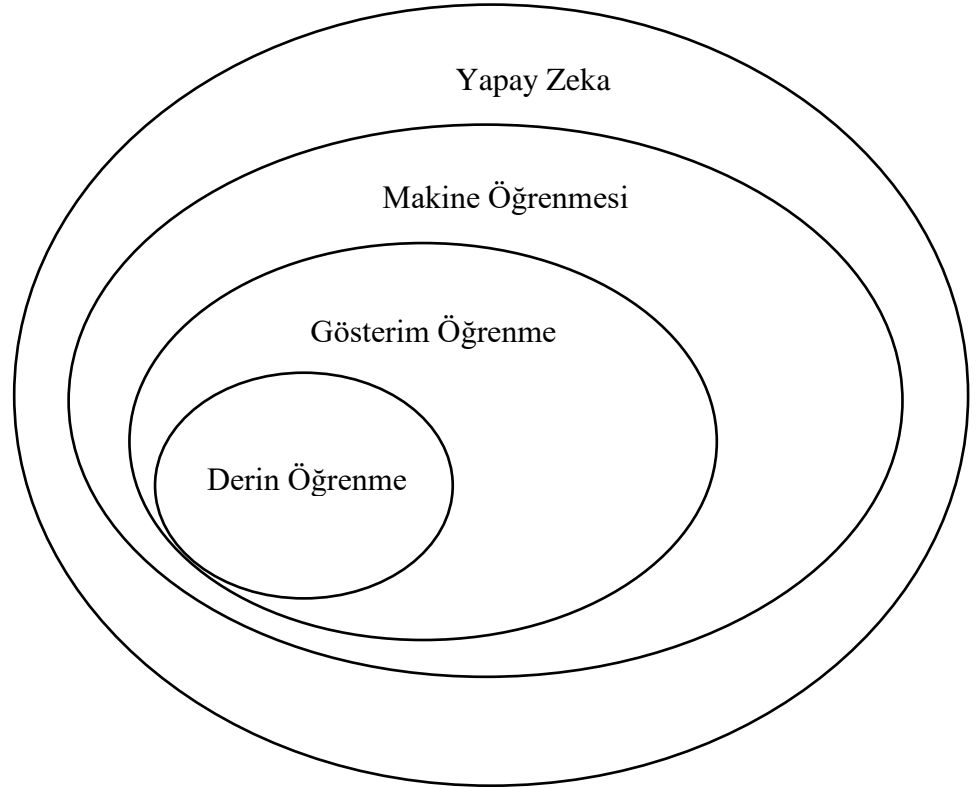
İnsanlar programlanabilir bilgisayarları icat ettiklerinde bu makinelerin insan zekasını gerektiren görevleri yerine getirip getiremeyeceklerini merak ettiler. Bu ise yapay zeka'yı, makine öğrenmesi, robotik, doğal dil işleme ve derin öğrenme gibi birçok konusu bulunan bir bilim dalına dönüştürdü.

İnsanlarda öğrenme, beynin bilgi ve deneyimlerden yararlanarak gelecekteki durumlar hakkında doğru karar verebilmesidir. Biyolojik olarak, sinir sisteminde bulunan sinir hücreleri arasında bilgi aktarımı yapılarak öğrenme gerçekleşmektedir. Yapay sinir ağları (YSA) ve benzeri öğrenme algoritmaları insan beyninin yapısı modellenerek geliştirilmişlerdir. Bu tür modellerin amacı biyolojik olarak öğrenmenin nasıl gerçekleştiğini saptayıp buna göre öğrenmeyi modellemektir (Yurtsever, 2019:63).

Yapay zekanın başlangıç dönemlerinde insanlar tarafından zorlukla çözülen problemler bilgisayarlar tarafından kolaylıkla çözüldü. Ama asıl problemin insanlar tarafından kolaylıkla gerçekleştirilebilen yüz tanıma ve ses tanıma gibi matematiksel olarak tanımlanması zor olan problemlerde olduğu ortaya çıktı. Bu zorlukları ortadan kaldırmak için ham veriden örüntüler çıkararak kendi bilgilerini elde eden makine öğrenmesi yaklaşımı ortaya çıktı. Makine öğrenmesi terimindeki öğrenme kavramı ile insan doğasındaki öğrenme kavramı benzer özellikler taşımaktadır. İnsanlar öğrenme sayesinde benzer davranışları yerine getirirken performanslarını iyileştirmektedir. Benzer şekilde, makine öğrenmesi, bilgisayarların veri ve algoritmaları kullanarak performanslarını arttırdıkları bir yapay zeka dalıdır. Makine öğrenmesi algoritmalarının performansı girdi verilerinin gösteriminin kalitesine büyük ölçüde bağlıdır. Birçok yapay zeka problemi, bu problem için gereken doğru öznitelik kümesini belirleyerek ve sonrasında bu öznitelikleri bir makine öğrenmesi algoritmasına vererek çözülebilir. Örneğin, bir kişinin ses yolunun uzunluğu kişiyi tanımak için bir öznitelik olarak kullanılabilir. Bu öznitelige bakarak kişinin kadın mı, erkek mi veya çocuk mu olduğu tespit edilebilir (Goodfellow vd., 2018:3).

Birçok problem için hangi özneliklerin toplanması gerektiği bilinemeyebilir. Bu sorunun çözümü için sadece gösterimden çıktıya eşlemenin değil, gösterimden kendisinin elde edilmesi için makine öğrenmesini kullanmaktır. Bu yaklaşım gösterim öğrenme olarak bilinmektedir. Genellikle, öğrenilen gösterimler elle oluşturulan gösterimlerden daha iyi performans veririler. Ayrıca, minimum insan müdahalesiyle yapay zeka sistemleri geliştirilmiş olur (Goodfellow vd., 2018:4).

Son yıllarda popüler olan derin öğrenmenin temelleri 1940'lı yıllara kadar dayanmaktadır. Derin öğrenmeye, 1940-1960 yılları arasında sibernetik (cybernetics), 1980-1990 yılları arasında ise bağlantıcılık (connectionism) gibi farklı isimler verilmiştir. Geçmişte farklı isimlerle anılmış olmasından dolayı günümüzde derin öğrenme yeni bir çalışma alanı gibi görülmektedir. Derin öğrenme, bilgisayarların veri ve deneyimleri kullanarak kendilerini geliştirdikleri bir makine öğrenmesi türüdür. (Goodfellow vd., 2018:11).



Şekil 15: Farklı YZ disiplinleri arasındaki ilişkiyi gösteren Venn şeması

Şekil 15'te yapay zeka, makine öğrenmesi, gösterim öğrenme ve derin öğrenme arasındaki ilişki gösterilmiştir. Şekilde görüldüğü gibi YZ bu teknolojilerin en üst kapsayıcısı olarak yer almaktadır. Zamanla YZ farklı alt disiplinlere ayrılmıştır. Bu



disiplinlerden biri olan makine öğrenmesi YZ'nin bir alt kolu olarak gösterilmiştir. Son yıllarda popüler olan derin öğrenme ise gösterim öğrenmenin bir türüdür ve şekil üzerinde gösterim öğrenme kapsamı içine alınmıştır.

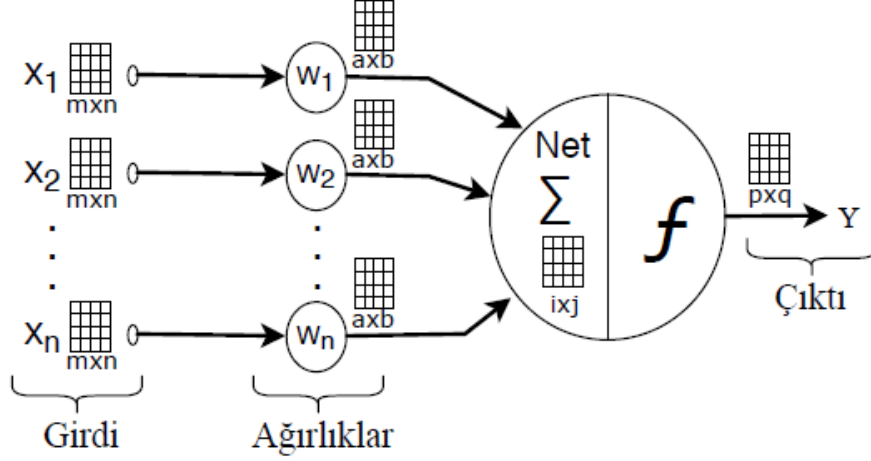
Her ne kadar yapay zeka, makine öğrenmesi ve derin öğrenmenin temelleri çok eski tarihlere dayansa da bu teknolojilerden yapay zeka 1950'li yıllarda, makine öğrenmesi 1980'li yıllarda, derin öğrenme ise 2006'lı yıllardan itibaren popüler hale gelmeye başlamıştır (Yurtsever, 2019:66).

### **A. Evrişimli Sinir Ağları**

Görüntü analizinde şimdiye kadarki en başarılı derin öğrenme mimarilerinden olan Evrişimli Sinir Ağları (ESA) medikal görüntülerden hastalık teşhisi, görüntü bölütleme, sınıflandırma, nesne tanıma, yüz tanıma ve video görüntülerini sınıflandırma gibi birçok alanda kullanılmaktadır (Eker ve Duru, 2021:460).

ESA'nın temelleri 1989 yılında LeCun ve arkadaşları tarafından atılmıştır (LeCun et al., 1989). Ancak, 2012 yılına kadar, büyük miktarda eğitim verisine ve güçlü hesaplama kaynaklarına ihtiyaç duyduğundan popüler olamamıştır. Büyük veri çağına girilmesi ve güçlü hesaplama kaynaklarına sahip olunması ile birlikte 2012 yılından itibaren ESA büyük ilgi görmeye başlamıştır. Özellikle, 2012 yılında düzenlenen ImageNet görüntü sınıflandırma yarışmasında ESA modellerinden AlexNet'in şampiyonluğu elde etmesi büyük yankı uyandırmıştır (Yurtsever, 2019:67).

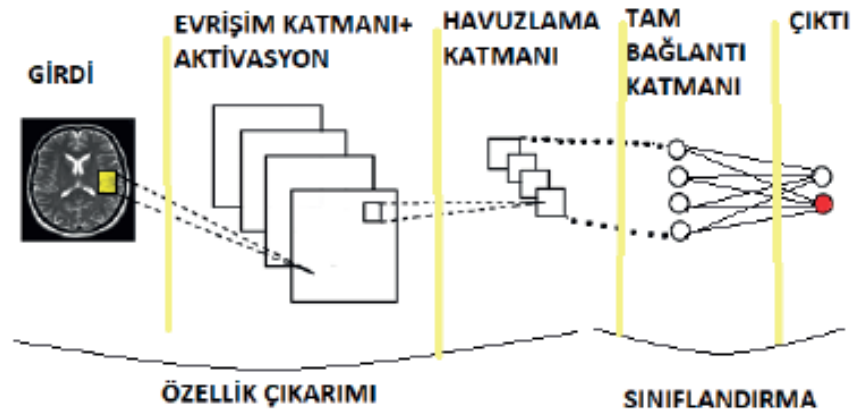
İnsan görme sisteminin modellenmesiyle oluşturulan ESA'lar görüntülerin analiz edilmesi için grid benzeri bir yapı kullanan özel bir sinir ağı türüdür. Başka bir deyişle katmanlarda matris çarpımı yerine evrişim ismi verilen işlem yapılmaktadır. Evrişimli sinir ağlarının ilk katmanlarında kenar bilgisi gibi daha basit öznitelikler elde edilmektedir. Daha derin katmanlara geçildikçe görüntüyü temsil eden daha karmaşık öznitelikler çıkarılmaktadır (Kızrak ve Bolat, 2018:266).



Şekil 16: ESA sinir düğümü (Yurtsever, 2019)

Şekil 16’da görüldüğü gibi bir ESA geleneksel sinir ağlarından farklı olarak sayılı veriler yerine matrisleri kullanmaktadır. Bu sayede yerel algı alanı (receptive field) şeklinde çalışırlar ve yerel uzaysal komşulukları korurlar (Mercan, 2015:12). Temel bir ESA, girdi katmanı, evrişim katmanı (convolution layer), havuzlama katmanı (pooling layer), tam bağlantı katmanı (fully connected layer) ve çıktı katmanı olmak üzere 5 katmandan oluşmaktadır.

Şekil 17’de bir evrişimli sinir ağı mimarisi görülmektedir. Girdi katmanında veri ham olarak ağı sunulmaktadır. Evrişim katmanında, girdi görüntüsünün üzerinde bir filtre dolaştırılarak özellik haritası (feature map) elde edilmektedir. Havuzlama katmanı ağı parametre sayısını ve boyutunu azaltmak için kullanılmaktadır. Tam bağlantı katmanı ise önceki katmanlardan gelen verileri ağırlandırarak birleştirir. Sonrasında, bir kayıp fonksiyonu kullanarak nöronlara eğitim boyunca verilecek en uygun ağırlık değerlerini hesaplar (Eker ve Duru, 2021:461).



## 1. Evrişim Katmanı

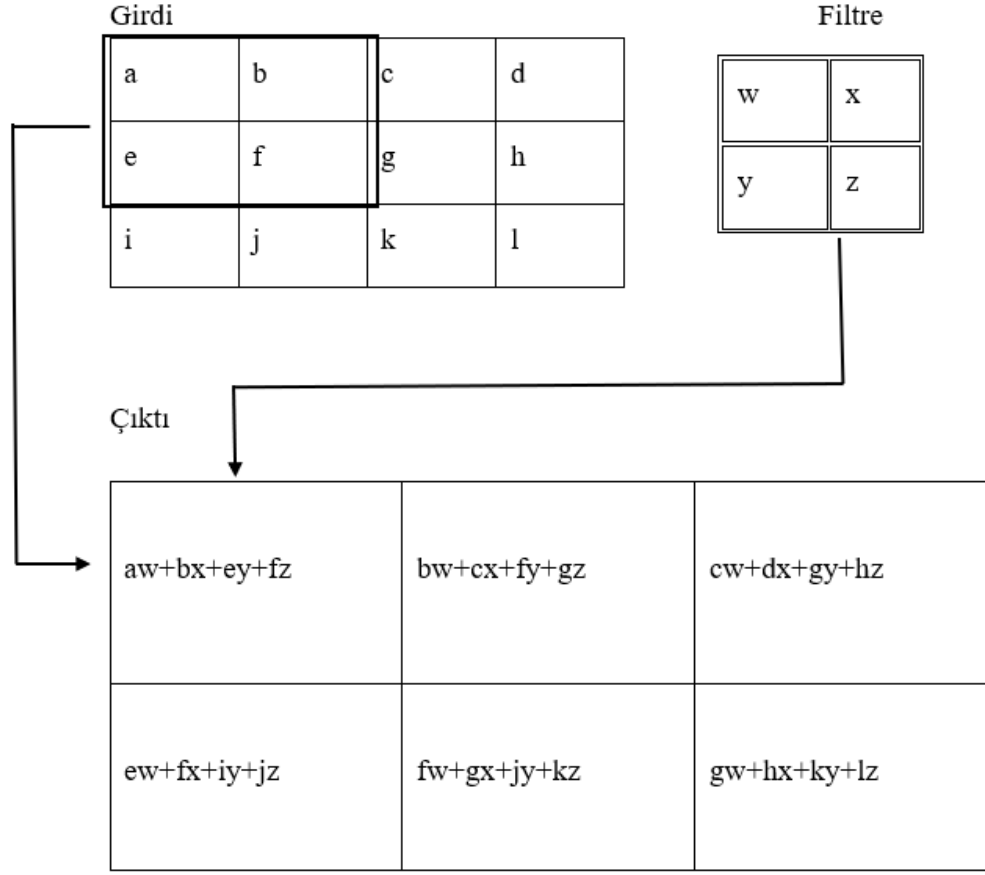
Evrişim katmanı, ESA'nın temel bileşenidir ve en az bir katmandan oluşmaktadır. Bu katman, girdi verilerinden özellikler çıkarmak amacıyla kullanılmaktadır. Evrişim işlemi, iki fonksiyonun matematiksel bir işlemi olarak tanımlanmaktadır. ESA'da evrişim işlemi, girdi verisi üzerinde evrişim filtresi gezdirilerek yapılmaktadır (Çifçi, 2021:62). Evrişim işleminde kullanılacak filtrenin x eksenine ve y eksenine göre simetriği alınmaktadır. Denklem 49'da evrişim işleminin formülü verilmiştir. Burada \* işareti evrişim işlemi, I giriş görüntüsünü, K ise filtreyi ifade etmektedir.

$$S(i,j) = (K*I)(i,j) = \sum_m \sum_n I(m,n)K(i - m, j - n) \quad (49)$$

Evrişim filtresinin simetriği alınmadan yapılan evrişim işlemine çapraz korelasyon (cross-correlation) denilmektedir (Kızrak ve Bolat, 2018:266). Formülü, Denklem 50'de verilmiştir.

$$S(i,j) = (K*I)(i,j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (50)$$

Şekil 9'da 2x2'lik bir filtre 4x3 boyutundaki bir görüntüde gezdirilerek evrişim işlemi uygulanmıştır. Görüldüğü gibi girdinin matris değerleri ile filtrenin matris değerlerinin örtüşen noktaları çarpılmakta ve tüm çarpım değerleri toplanmaktadır. Elde edilen sonuç, çıkış matrisinin ilgili elemanın değerini oluşturmaktadır. Sonuçta, 4x3'lük bir girdi matrisi taranmış ve 3x2'lik bir çıkış matrisi elde edilmiştir.



Şekil 18: İki boyutlu evrişim örneği

İki boyutlu matris için evrişim işlemi Şekil 18’de gösterildiği gibi yapılmaktadır. Eğer, giriş görüntüsü üç kanaldan oluşuyorsa her kanal için ayrı ayrı evrişim işlemi yapılmaktadır. Yani, giriş görüntüsündeki kanal sayısı ile filtre sayısı eşit olmaktadır. Filtre matrisi, geri yayılımla sürekli olarak güncellenen ağırlıklar matrisidir. Sonrasında, çıkış matrisine skaler bir bias değeri eklenerek aktivasyon fonksiyonundan geçirilmektedir (Kızrak ve Bolat, 2018:266).

Şekil 18’de gösterilen evrişim işleminde girdi ve çıktı matrislerinin boyutu farklıdır. Girdi matrisi boyutu 4x3, filtre matrisi boyutu 2x2 ve kaydırma adım sayısı 1 olduğunda çıktı matrisinin boyutu 3x2 olmaktadır. Çıktı matrisinin boyutu Denklem 15’e göre hesaplanmaktadır.

$$m = \frac{n - f}{s} + 1 \quad (51)$$

Denklem 21’de m çıktı yüksekliği veya uzunluğunun boyutu, n girdi yüksekliği veya uzunluğunun boyutu, f filtre matrisinin boyutu, s ise kaydırma adım sayısıdır.

Kaydırma adım sayısı, girdi matrisi üzerinde filtre gezdirilirken yatay ve düşey ekseninde kaç adım kaydırılacağını belirler. Çıktı matrisinin boyutu kaydırma adım sayısı değiştirilerek belirlenebilir. Çıktı matrisinin boyutu azaltılmak isteniyorsa kaydırma adım sayısı arttırılmalıdır. Denklem 51 ile bulunan çıktı matrisi boyutu  $m$  tamsayı bir değer çıkmalıdır. Eğer tam sayı bir değer çıkmıyorsa filtre boyutu veya kaydırma adım sayısı değiştirilmelidir (Yurtsever, 2019:80).

Çok katmanlı bir derin ağda, her evrişim işleminde çıktı görüntüsünün boyutu küçülmektedir. Bu nedenle, girdi görüntüsü üzerinde filtre gezdirilirken, kenarlardaki piksellerle filtre daha az temas etmekte dolayısıyla çıktı görüntüsünde kenar piksellerin etkisi azalmaktadır. Tam tersine, merkezdeki pikseller, filtre ile daha fazla temas ettiğinden çıktı görüntüsü üzerindeki etkisi artmaktadır. Bu sorunların çözümü için girdi görüntüsüne dolgulama (padding) işlemi yapılmaktadır. Dolgulama ile girdi görüntüsünün çevresine ilave sıfır veya kenar piksellerle aynı değerler eklenmektedir. Bu sayede, girdi görüntüsü boyutu ile çıktı görüntüsü boyutunun aynı olması sağlanmaktadır. Şekil 19’da girdi görüntüsünün boyutunu arttırmak için girdi görüntüsünün çevresine sıfır dolgulama (padding) işleminin nasıl yapıldığı gösterilmiştir.

0	0	0	0	0	0	0	0
0	1	2	0	2	1	1	0
0	2	1	0	2	2	1	0
0	1	2	1	0	1	2	0
0	1	1	0	0	1	1	0
0	1	0	2	1	2	1	0
0	1	2	0	0	0	1	0
0	0	0	0	0	0	0	0

Şekil 19: Sıfır dolgulama işlemi

Evrişim işleminden sonra girdi matrisi ile çıktı matrisi boyutunun aynı olması için girdi matrisine eklenecek dolgulama katman sayısı Denklem 52’da verilmiştir.

$$p = \frac{f - 1}{2} \quad (52)$$

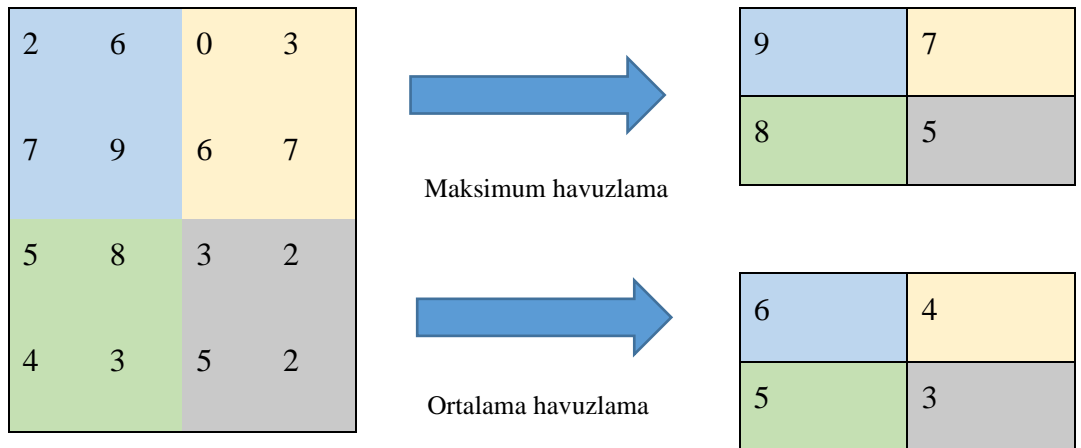
Denklem 52’de p dolgulama katman sayısı ifade ederken, f filtre boyutunu ifade etmektedir. Denklem 51’de verilen formüle dolgulama parametresi de eklenince çıkış matrisi boyutu Denklem 53’de verildiği gibi hesaplanır.

$$m = \frac{n - f + 2p}{2} + 1 \quad (53)$$

## 2. Havuzlama Katmanı

ESA’nın temel yapı taşlarından biri de havuzlama (Pooling) katmanıdır. Amacı, görüntü boyutunu küçülterek ağıdaki parametre sayısını ve hesap yükünü azaltmaktır. Görüntü boyutunun küçülmesi veride kayıplara yol açmaktadır. Ancak, bu işlem, ağız ezberleme (overfitting) yapmasını önleyebilmektedir. Havuzlama katmanının kullanımı zorunlu değildir. Kullanıldığında ya evrişim katmanları arasına ya da evrişim katmanlarının en sonuna konmaktadır (Sürücü, 2021:44).

Havuzlama işlemi, evrişim işlemi gibi, havuzlama penceresinin, adım sayısına ve havuzlama türüne bağlı olarak, girdi matrisi üzerinde geçtiği her konum için bir değer hesapladığı ve bu değeri çıktı matrisine yazdığı bir işlemidir. Havuzlama türlerinden en yaygın kullanılanları maksimum havuzlama (max pooling) ve ortalama havuzlamadır (average pooling). Maksimum havuzlamada, girdi görüntüsü üzerinde havuzlama penceresine denk gelen piksellerden en büyük olanı seçilir. Ortalama havuzlamada ise girdi görüntüsü üzerinde havuzlama penceresine denk gelen piksellerin ortalaması alınır (Kurt, 2018:32).



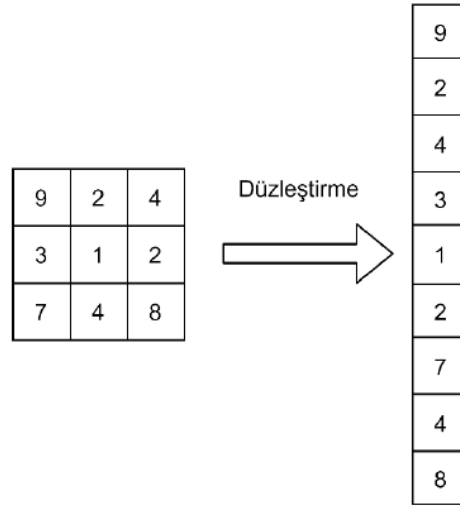
Şekil 19: Maksimum ve ortalama havuzlama işlemi

Şekil 19’da havuzlama işleminin matris olarak gösterimi yer almaktadır. 4x4 boyutlarındaki girdi matrisi üzerinde, iki adım kaydırmalı, 2x2 boyutlarında

havuzlama penceresi kullanılarak maksimum havuzlama ve ortalama havuzlama işlemleri gösterilmiştir. Görüldüğü gibi her iki yöntemde de çıktı matrisinin boyutu yarı yarıya azalarak 2x2 boyutuna düşmüştür. Havuzlama işlemi evrişim işlemine benzer olduğundan çıktı matrisinin boyutu Denklem 53'de verilen formül ile hesaplanmaktadır.

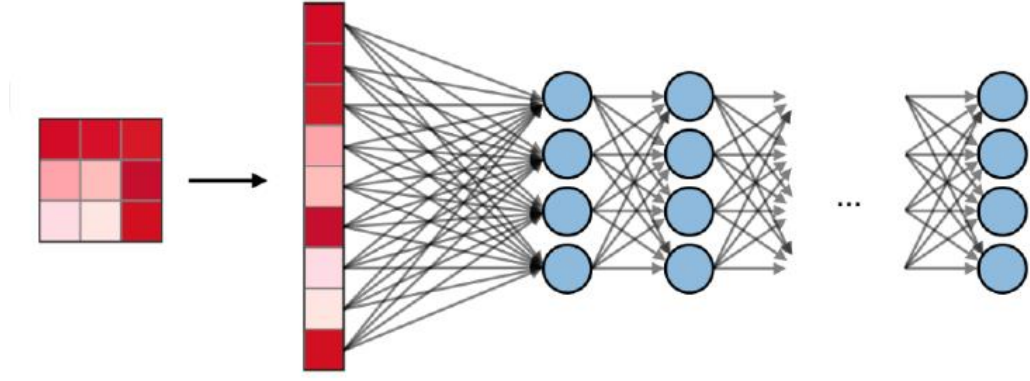
### 3. Tam Bağlantı Katmanı

Tam bağlantı katmanı (fully connected layer) çok katmanlı algılayıcıyla aynı şekilde çalışmaktadır. Öncelikle, girdi matrisi düzleştirilerek (flattening) bir boyutlu vektöre dönüştürülmektedir. Şekil 20'de düzleştirme işlemi gösterilmiştir.



Şekil 20: Düzleştirme işlemi

Şekil 21'de görüldüğü gibi tam bağlantı katmanı kendinde önce gelen katmanların ürettiği değerleri girdi olarak düzleştirme işlemi yapmakta sonrasında ise çıktı katmanındaki sınıf sayısı kadar sonuç üretmektedir (Kurt, 2018:33).



Şekil 21: Tam bağlantı katmanı gösterimi (Amidi ve Amidi, 2019)

## B. Evrişimli Sinir Ağı Modelleri

2012 yılında düzenlenen ImageNet görüntü sınıflandırma yarışmasında ESA modellerinden AlexNet şampiyonluğu kazandı. ImageNet yarışmasında milyonlarca görüntü ve 1000 civarında sınıf bulunmaktadır. Görüntülerin ait olduğu sınıfı doğru bir şekilde tespit eden model yarışmayı kazanmaktadır. Yarışma sonucunda AlexNet modeli büyük bir başarı sağladı ve sonrasında birçok ESA modeli geliştirilmeye başlandı. Tarihsel sıraya göre; LeNet, AlexNet, ZFNet, GoogLeNet, VGGNet ve ResNet gibi birçok derin öğrenme modeli bulunmaktadır. Özellikle, AlexNet ve GoogLeNet modelleri birçok alanda yaygın olarak kullanılmaktadır.

### 1. LeNet

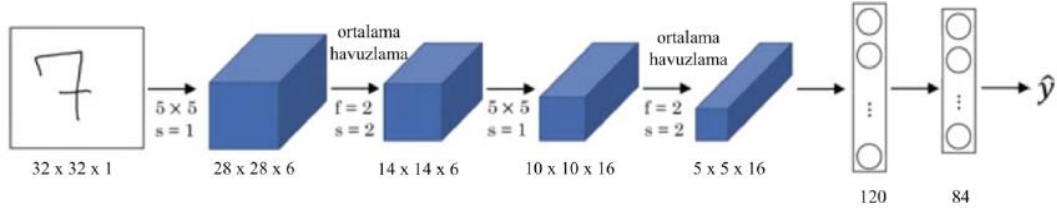
İlk ESA modellerinden bir olan LeNet Yann LeCun ve arkadaşları tarafından 1998 yılında tanıtıldı (LeCun et al., 1998). Model, görüntülerdeki el yazısı rakamlarının tanınması amacıyla MNIST (Modified National Institute of Standards and Technology) veri setiyle eğitilerek başarıyla uygulanmıştır. Şekil 22’de MNIST veri seti verilmiştir.





Şekil 22: MNIST veri seti

Şekil 22’de görüldüğü gibi MNIST veri setinde farklı kişiler tarafından el yazısı ile yazılmış rakamların görüntüleri yer almaktadır. Şekil 23’te LeNet modeli gösterilmiştir.

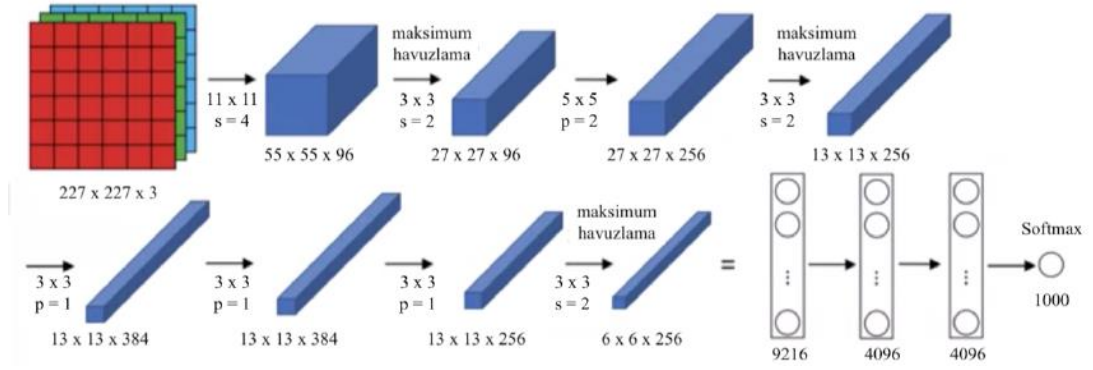


Şekil 23: LeNet modeli (Yurtsever, 2019:84)

Şekil 23’de görüldüğü üzere girdi görüntüsü boyutu  $32 \times 32 \times 1$ ’dir. Her bir evrişimli katmanda  $5 \times 5$ ’lik bir filtre 1 kaydırma adım sayısı ile uygulanmıştır. Aktivasyon fonksiyonu olarak sigmoid seçilmiştir. Boyut azaltmak için her havuzlama katmanında ortalama havuzlama yöntemi,  $2 \times 2$ ’lik bir pencere ve 2 kaydırma adım sayısı ile uygulanmıştır. Evrişim ve havuzlama işlemlerinden sonra görüntü yüksekliği ve genişliği azalırken kanal sayısı artmaktadır. Sonuçta görüntü boyutu  $5 \times 5 \times 16$ ’ya düşmüştür. Evrişim ve havuzlama katmanlarından sonra elde edilen çıktıyı tam bağlantı katmanına geçirmek için düzleştirme işlemi yapılmıştır. Düzleştirme işleminden sonra tam bağlantı katmanına giren parametre sayısı 400 olmuştur. Bu 400 düğüm 120 nörona sahip bir tam bağlantı katmanına bağlanmıştır. Sonrasında, 84 nörona sahip bir başka tam bağlantı katmanı kullanılmıştır. Çıktı katmanında ise 0-9 arasındaki rakamları sınıflandırdığımız için 10 sınıflı bir esnek eşikleyici (softmax) yer almaktadır (Yurtsever, 2019:84; Kızırcak ve Bolat, 2018:267).

## 2. AlexNet

Krizhevsky ve arkadaşları tarafından geliştirilen AlexNet modeli 2012 yılında düzenlenen ImageNet görüntü sınıflandırma yarışmasında birinci olmuştur. Bu başarının ardından derin öğrenme ve evrişimli sinir ağı modelleri yeniden popüler hale gelmeye başlamıştır. Şekil 24’te AlexNet modeli görülmektedir.



Şekil 24: AlexNet modeli (Yurtsever, 2019:85)

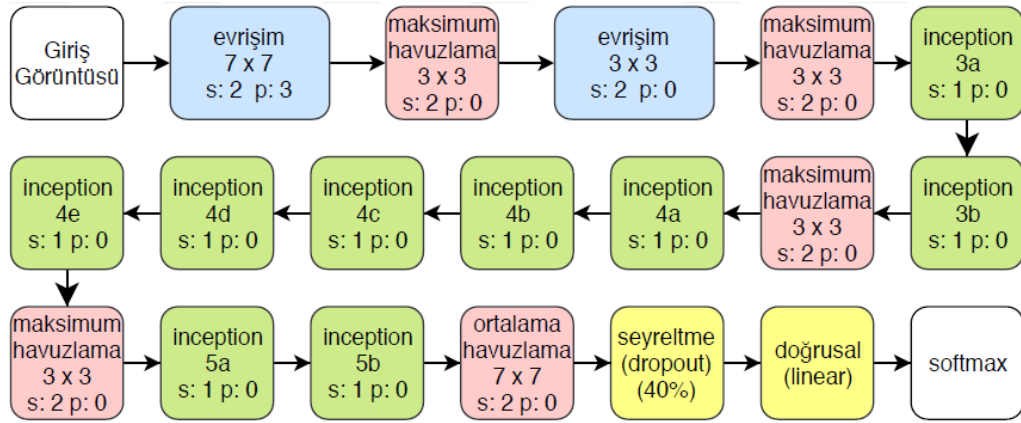
Şekil 24’te görüldüğü üzere AlexNet modeli beş evrişim katmanı ve üç tam bağlantı katmanından oluşmaktadır. Bu model paralel çift GPU (Grafik İşlem Birimi- Graphics Processing Unit) üzerinde çalışacak şekilde tasarlanmıştır. Her katmandan sonra aktivasyon fonksiyonu olarak ReLU (Rectified Linear Unit) uygulanmaktadır. Havuzlama katmanlarında ise maksimum havuzlama işlemi yapılmaktadır (Yurtsever, 2019:85; Kızrak ve Bolat, 2018:267).

Girdi katmanındaki görüntünün boyutları 227x227x3 olarak belirlenmiştir. İlk evrişim katmanında girdi görüntüsüne 96 adet 11x11x3 boyutlarında ve 4 adım kaydırmalı filtre uygulanarak 55x55x96 boyutlarında çıktı görüntüsü elde edilmiştir. İkinci evrişim katmanı havuzlama uygulanmış çıktıya 256 adet 5x5x48 boyutlarında ve 2 adım kaydırmalı filtre uygular. Üçüncü evrişim katmanı 384 adet 3x3x256 boyutlarında, dördüncü evrişim katmanı 384 adet 3x3x192 boyutlarında, beşinci katman 256 adet 3x3x192 boyutlarında filtre uygular. Tam bağlantı katmanında 4096 nöron bulunan iki katman yer almaktadır. Son katmanda ise 1000 sınıf arasından bir dağılım gerçekleştiren esnek eşikleyici (softmax) bulunmaktadır (Kurt, 2018:41).

### 3. GoogLeNet

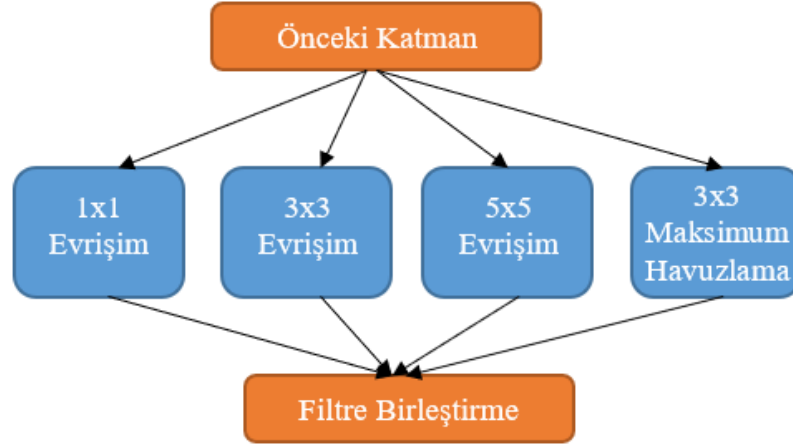
Google çalışanlarından Szegedy ve arkadaşları tarafından geliştirilen GoogLeNet modeli 2014 yılında düzenlenen ImageNet yarışmasında birinci olmuştur. GoogLeNet modeli 22 katmandan oluşmaktadır. GoogLeNet modelinin, AlexNet modelinden 12 kat daha az parametresi bulunmaktadır. Ayrıca, AlexNet'ten daha düşük olan %5,7 hata oranı ile daha yüksek bir doğruluğa sahiptir (Szegedy et al., 2015).

LeNet modelinden esinlenerek geliştirilen GoogLeNet modeli inception ismi verilen modüllerden oluşmaktadır. Bu isim, Inception filminde geçen "We need to go deeper (Daha derine inmeliyiz)" sözünden esinlenerek verilmiştir (Yurtsever, 2019:86). Şekil 25'de GoogLeNet modeli blok şeması verilmiştir.



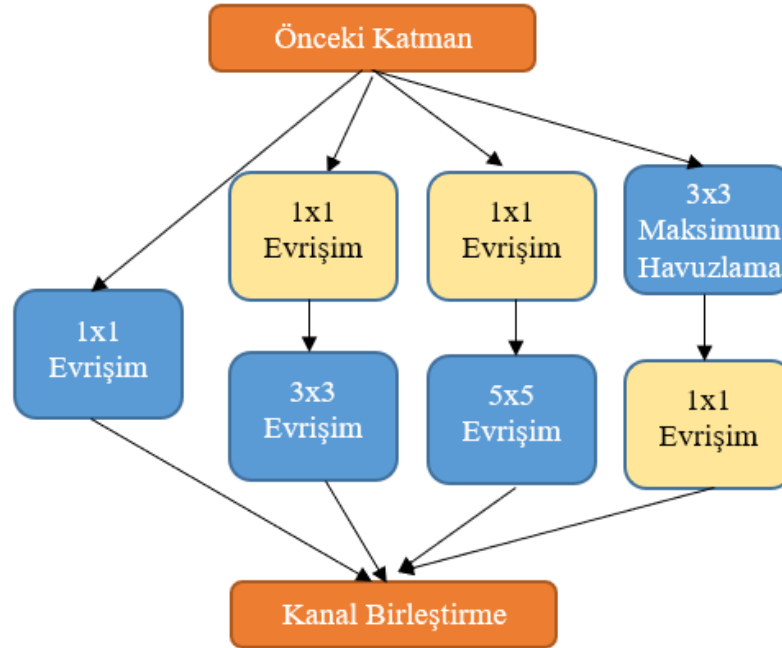
Şekil 25: GoogLeNet modeli blok şeması (Yurtsever, 2019:86)

GoogLeNet ağının temel yapı taşı inception modülüdür. Şekil 18'de basit inception modülü görülmektedir. Girdiye, paralel olarak, 1x1, 3x3 ve 5x5 boyutlarındaki filtreler ile evrişim işlemi ve 3x3 boyutunda havuzlama penceresi ile maksimum havuzlama işlemi uygulanmaktadır. Bu işlemler sonucu elde edilen çıktılar birleştirilerek tek bir çıktı elde edilmektedir. Böylece ağın, aynı anda hem genel hem de spesifik özellikleri çıkarabilmesi sağlanmaktadır (Kurt, 2018:45).



Şekil 26: Basit inception modülü

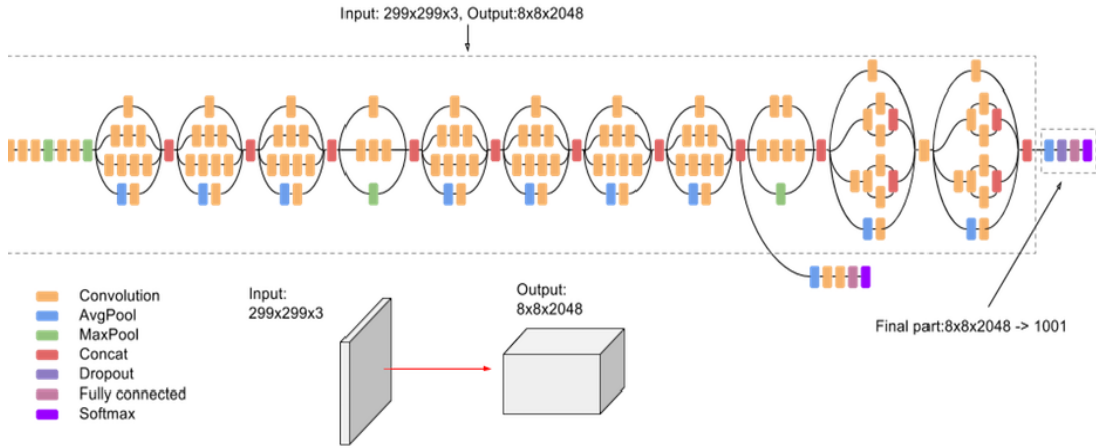
Şekil 26’da verilen basit inception modülünün kullanılması paralel işlemler yüzünden çıktı boyutunun, parametre sayısının ve işlem karmaşıklığının artmasına yol açmaktadır. Bu sorunların çözümü için 3x3 ve 5x5 evrişim katmanları öncesine ve 3x3 maksimum havuzlama katmanı sonrasına 1x1 evrişim katmanı eklenerek boyut düşürülmektedir (Kızırak ve Bolat, 2018:268). AlexNet modelinde 60 milyon olan parametre sayısı GoogLeNet modeliyle 5 milyona düşürülmüştür. Şekil 27’de inception modülü diyagramı verilmiştir.



Şekil 27: Inception modülü

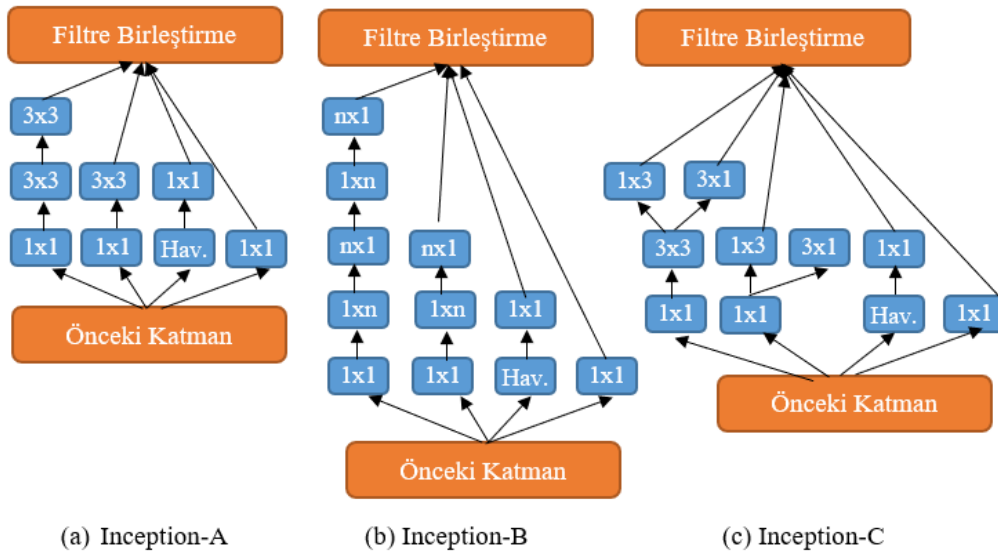
#### 4. Inception V3

Inception v3 modeli, inception mimarisine dayalı GoogLeNet modelinin üçüncü sürümüdür. GoogLeNet'te 22 olan katman sayısı Inception v3'te 42'ye çıkarılmıştır. Katman sayısındaki bu artışa rağmen hesaplama maliyeti yalnızca 2,5 kat daha yüksektir (Akgül, 2022:47). Bunu sağlamak için daha fazla sayıda ve daha küçük boyutlarda evrişim işlemi yapılmaktadır. Şekil 28'de Inception v3 modeli verilmiştir.



Şekil 28: Inception v3 modeli (cloud.google.com, 2022)

Inception v3 modelinde GoogLeNet'teki inception modülü Inception-A, Inception-B ve Inception-C olarak üç farklı şekilde yeniden düzenlenmiştir.



Şekil 29: Inception-A, Inception-B ve Inception-C modülleri

Şekil 29a'da görüldüğü gibi orijinal inception modülündeki 5x5 evrişim filtresi yerine inception-A modülünde 2 tane 3x3 evrişim filtresi kullanılmıştır. Böylece,

parametre sayısı  $5 \times 5 = 25$ 'ten  $2 \times (3 \times 3) = 18$ 'e düşürülmüştür. Inception-B modülünde  $n \times n$  boyutlu evrişim filtreleri yerine  $1 \times n$  ve  $n \times 1$  kombinasyonlarında evrişim filtreleri kullanılmıştır. Inception-C modülünde ise  $3 \times 3$  boyutlu evrişim filtreleri  $1 \times 3$  ve  $3 \times 1$  evrişim filtreleri ile değiştirilmiştir. Böylece, parametre sayısı  $3 \times 3 = 9$ 'dan  $(1 \times 3) + (3 \times 1) = 6$ 'ya düşürülmüştür. Sonuçta, Inception v3 modelinde, ağın derinliği artarken parametre sayısı azalmaktadır (Akgül, 2022:48). Bu çalışmada, beyin tümörlerinin sınıflandırılması için Inception v3 modeli kullanılmıştır.

#### IV. DENEYSEL SONUÇLAR VE TARTIŞMA

Deneyisel çalışmalar, Windows 10 işletim sisteminin kurulu olduğu sekiz çekirdekli Intel Core i7-11700 CPU, GeForce RTX 3060 GPU ve 16GB RAM belleğe sahip kişisel bir bilgisayarda gerçekleştirilmiştir. Uygulama CUDA 11 kullanılarak Python 3.9 sürümünde yazılmıştır. İki boyutlu ADD Python'un PyWavelet kütüphanesi kullanılarak hesaplanmıştır. Modelin eğitimi ve testi TensorFlow alt yapısını kullanan Keras Kütüphanesi üzerinde uygulanmıştır.

Deneyisel çalışmalarda, öğrenme katsayısı (learning rate) 0,0001 ve en küçük yığın boyutu (batch size) 32 olarak belirlenmiştir. Hata oranını en aza indirmek için derin sinir ağlarında etkin bir optimizasyon yöntemi olan RMSProp kullanılmıştır.

Bu uygulamada kullanılan çok katmanlı algılayıcı yapısında 224x224x3 sinir hücresinden oluşan girdi katmanı, 4 sinir hücresine sahip gizli katman ve 4 sinir hücresinden oluşan çıktı katmanı yer almaktadır. Çıktı katmanından sonra dört sınıftan her birisi için olasılık değeri üreten softmax sınıflandırıcısı bulunmaktadır.

Bütün eğitim setiyle bir kez eğitim yapıldığında bir eğitim adım sayısı (epoch) tamamlanır. Tüm eğitim setinin ağa kaç kez gösterileceği eğitim tur sayısı ile belirlenir. Bu parametre başlangıçta 100 olarak belirlenmiştir ancak erken durdurma (early stopping) tekniği kullanılmıştır. Erken durdurma tekniği ile modelin eğitim seti üzerinde gereğinden fazla çalışıp aşırı öğrenmeyi (over fitting) engellemesi sağlanabilmektedir. Bu yöntemde kayıp (loss) hata değeri belirli bir eşiğin altına düştüğünde eğitim aşaması sonlandırılmaktadır. Çizelge 2'de verilen eğitim adım sayıları tamamlandığında eğitim aşaması erken durdurma fonksiyonu tarafından durdurulmuştur.

Çizelge 2: Doğrulama setleri için erken durdurma eğitim adım sayıları

Doğrulama seti	Eğitim adım sayısı
Doğrulama seti-1	15
Doğrulama seti-2	11
Doğrulama seti-3	8
Doğrulama seti-4	4
Doğrulama seti-5	6

Model başarımını ölçmek için 5 kat çapraz doğrulama (cross validation) yöntemi kullanılmıştır. 5 kat çapraz doğrulama ile veri seti 5 parçaya bölünmekte ve tüm parçalar hem eğitim hem de doğrulama seti olarak kullanılmaktadır. Bu sayede dağılım ve parçalanma nedeniyle doğrulama setinin doğru olarak seçilip seçilmediği sorunu ortadan kalkmaktadır. Nihai model doğruluğu, çapraz doğrulama boyunca oluşturulan 5 modelin ortalaması alınarak bulunmaktadır.

Dalgacık özneliklerini çıkartmak için görüntülere Daubechies dalgacık ailesinden db2 dalgacığı kullanılarak 1 seviyeli ADD ayrıştırması yapılmıştır. Ayrıştırma sonucunda bulunan alt bileşenlerden yaklaşım bileşeni Inception V3 ağına girdi olarak verilmiştir. Model doğruluğu, test veri setindeki doğru tahmin edilen örnek sayısının toplam örnek sayısına oranı ile bulunmaktadır. Çizelge 3, önerilen modelin 5 kat çapraz doğrulama sürecinin her testindeki doğruluk oranlarını göstermektedir.

Çizelge 3: Doğrulama setleri için doğruluk oranları

Doğrulama seti	Doğruluk
Doğrulama seti-1	98,29%
Doğrulama seti-2	99,72%
Doğrulama seti-3	99,93%
Doğrulama seti-4	100%
Doğrulama seti-5	100%
Ortalama	99,58%



Çizelge 4: İlgili çalışmaların doğruluk oranları

Çalışma	Yöntem	Veri Seti	Sınıf	Doğruluk
Özkaraca ve arkadaşları	VGG16Net/DenseNet - Softmax	Figshare, BR35H ve SARTAJ	4	94-97%
Gomez Guzman ve arkadaşları	Inception V3-Softmax	Figshare, BR35H ve SARTAJ	4	97,12%
Ali ve arkadaşları	ShuffleNet-DVM	Figshare, BR35H ve SARTAJ	4	98,40%
Önerilen model	ADD-Inception V3-Softmax	Figshare, BR35H ve SARTAJ	4	99,58%

Çizelge 4’te bu çalışmada kullanılan 4 sınıflı veri seti ile yapılan çalışmaların doğruluk oranları verilmiştir. Bu tezde kullanılan yöntem ile elde edilen doğruluk oranının diğer çalışmalara kıyasla daha yüksek olduğu görülmektedir.

Karışıklık matrisi (confusion matrix), bir sınıflandırma modelinin doğru ve yanlış tahmin ettiği sınıf bilgileri ile veri setinde bulunan gerçek sınıf bilgilerinin yer aldığı tablodur. Doğrulama seti-1 için karışıklık matrisinin sonuçları Çizelge 5’te verilmiştir. Çizelge 5’teki sonuçlara göre 1 tane normal, 2 tane gliyom ve 21 tane menenjiyom türü tümör görüntüsü yanlış olarak tahmin edilmiştir. Hipofiz bezi tümörleri ise doğru olarak sınıflandırılmıştır.

Çizelge 5: Doğrulama seti-1 için karışıklık matrisi

		Tahmin edilen sınıflar			
		Normal	Gliyom	Menenjiyom	Hipofiz bezi
Gerçek sınıflar	Normal	382	0	1	0
	Gliyom	0	322	2	0
	Menenjiyom	1	11	331	9
	Hipofiz bezi	0	0	0	346

Çizelge 6’da doğrulama seti-2’nin karışıklık matrisi sonuçları verilmiştir. Buna göre 1 tane gliyom ve 3 tane menenjiyom türü tümör görüntüsü yanlış olarak

sınıflandırılmıştır. Normal beyin MRG görüntüleri ile hipofiz bezi türü tümörler ise doğru olarak sınıflandırılmıştır.

Çizelge 6: Doğrulama seti-2 için karışıklık matrisi

		Tahmin edilen sınıflar			
		Normal	Gliyom	Menenjiyom	Hipofiz bezi
Gerçek sınıflar	Normal	398	0	0	0
	Gliyom	0	329	1	0
	Menenjiyom	0	2	340	1
	Hipofiz bezi	0	0	0	334

Çizelge 7’de doğrulama seti-3’ün karışıklık matrisi sonuçları verilmiştir. Buna göre 1 hipofiz bezi türü tümör görüntüsü yanlış olarak sınıflandırılmıştır. Normal beyin MRG görüntüleri doğru olarak sınıflandırılmıştır. Tümörlü görüntülerden gliyom ve menenjiyom türü tümörler de doğru olarak tahmin edilmiştir.

Çizelge 7: Doğrulama seti-3 için karışıklık matrisi

		Tahmin edilen sınıflar			
		Normal	Gliyom	Menenjiyom	Hipofiz bezi
Gerçek sınıflar	Normal	412	0	0	0
	Gliyom	0	303	0	0
	Menenjiyom	0	0	318	0
	Hipofiz bezi	0	0	1	354

Çizelge 8’de doğrulama seti-4’ün karışıklık matrisi sonuçları verilmiştir. Buna göre tüm beyin MRG görüntüleri doğru olarak sınıflandırılmıştır.

Çizelge 8: Doğrulama seti-4 için karışıklık matrisi

		Tahmin edilen sınıflar			
		Normal	Gliyom	Menenjiyom	Hipofiz bezi

Gerçek sınıflar	Normal	411	0	0	0
	Gliyom	0	341	0	0
	Menenjiyom	0	0	298	0
	Hipofiz bezi	0	0	0	354

Çizelge 9’da doğrulama seti-5’in karışıklık matrisi sonuçları verilmiştir. Buna göre tüm beyin MRG görüntüleri doğru olarak sınıflandırılmıştır.

Çizelge 9: Doğrulama seti-5 için karışıklık matrisi

		Tahmin edilen sınıflar			
		Normal	Gliyom	Menenjiyom	Hipofiz bezi
Gerçek sınıflar	Normal	396	0	0	0
	Gliyom	0	323	0	0
	Menenjiyom	0	0	334	0
	Hipofiz bezi	0	0	0	351

Karışıklık matrisinden faydalanarak Denklem 6, 7 ve 8’de verilen kesinlik, duyarlık ve F1 puanı gibi farklı performans ölçüm metrikleri hesaplanabilir.

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (7)$$

$$\text{F1_puanı} = \frac{2 * \text{duyarlık} * \text{kesinlik}}{\text{duyarlık} + \text{kesinlik}} \quad (8)$$

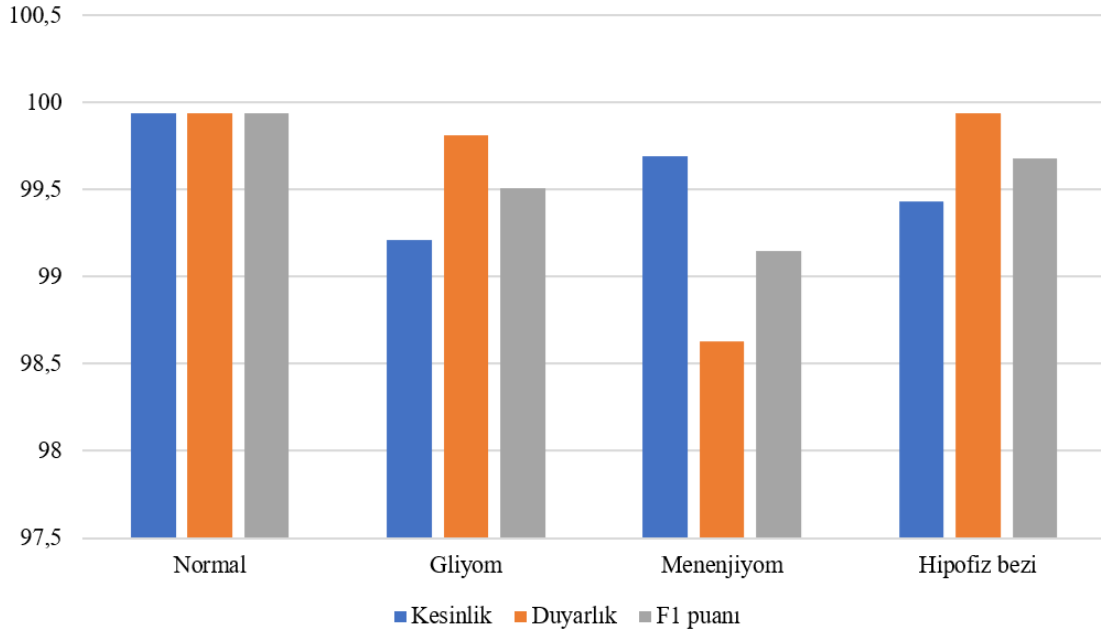
Yukarıdaki denklemlerde yer alan TP, FP, TN ve FN sırasıyla doğru pozitif, yanlış pozitif, doğru negatif ve yanlış negatif olarak tahmin edilen örnek sayısını göstermektedir. F1 puanı kesinlik ve duyarlık değerlerinin harmonik ortalaması alınarak hesaplanmaktadır. Uç durumların gözardı edilmesi için aritmetik ortalama yerine harmonik ortalama alınmaktadır. Örneğin, kesinlik değeri 1 ve duyarlık değeri 0 olan bir modelin F1 puanı aritmetik ortalama ile hesaplanırsa sonuç 0,5 olarak çıkacaktır. Bu ise modelin performansını değerlendirirken bizi yanıltabilir. Çizelge

10’da önerilen modelin sınıflara göre ortalama kesinlik, duyarlılık ve F1 puanı değerleri verilmiştir.

Çizelge 10: Sınıflara göre performans metriklerinin ortalama değerleri

Kategori	Kesinlik	Duyarlılık	F1 puanı
Normal	99,94%	99,94%	99,94%
Gliyom	99,21%	99,81%	99,51%
Menenjiyom	99,69%	98,63%	99,15%
Hipofiz bezi	99,43%	99,94%	99,68%

Şekil 30’da sınıflara göre performans metriklerinin ortalama değerleri görsel olarak verilmiştir. Görüldüğü gibi normal beyin MRG görüntülerinin performans metrik değerleri en yüksek olarak bulunmuştur.



Şekil 30: Kategoriye göre performans metriklerinin ortalama değerleri

## V. SONUÇLAR VE ÖNERİLER

Bu tezde beyin MRG görüntülerinden beyin tümörlerinin sınıflandırılması için derin öğrenme tabanlı bir model önerilmiştir. Günümüzde beyin tümörleri nedeniyle meydana gelen ölümlerin artması beyin tümörlerinin otomatik olarak sınıflandırılması çalışmalarını önemli bir araştırma konusu haline getirmiştir. Önerilen model sayesinde, uzmanlar, beyin tümörlerini daha doğru ve daha kısa sürede sınıflandırabileceklerdir.

Derin öğrenme tabanlı yöntemler, sınıflandırmadaki başarılı performansları nedeniyle yapay zeka alanında yaygın olarak kullanılmaktadır. Derin öğrenme, insan müdahalesi olmadan, birçok veri işleme katmanı kullanarak, verilerden otomatik olarak özellikler çıkarabilmektedir. Bu tezde derin öğrenme tabanlı yöntemlerden olan ESA modelleri değerlendirilmiş ve beyin tümörlerinin sınıflandırmasında uygulanmıştır.

Literatürde yer alan beyin MRG görüntülerini sınıflandırma çalışmalarının çoğu görüntü bölütlemeye dayanmaktadır. Önce görüntü bölütleme işlemi yapılmakta sonra beyin MRG görüntüsünün türü sınıflandırılmaktadır. Bu çalışmada önerilen yöntemde ise girdi görüntüsüne herhangi bir bölütleme işlemi yapılmamıştır. Girdi görüntüsünden özellik çıkarımı ve boyutunun azaltılması için dalgacık dönüşümü kullanılmıştır. Sonrasında, elde edilen dalgacık öznitelikleri Inception V3 ağının girdisi olarak kullanılmıştır.

Önerilen model ön işlem, özellik çıkarımı ve sınıflandırma olmak üzere üç aşamadan oluşmaktadır. Figshare, SARTAJ ve Br35H veri setlerinin birleşiminden oluşan dört sınıflı veri seti kullanılarak önerilen model test edilmiştir. Elde edilen performans metrik sonuçları hem sayısal hem de görsel olarak değerlendirilmiştir. Yapılan çalışmada modelin ortalama doğruluk başarımı %99,58 olarak hesaplanmıştır. Modelin doğruluk başarımı literatürde yer alan 4 sınıflı veri seti ile yapılan çalışmalarla karşılaştırılmış ve önerilen modelin daha yüksek bir doğruluk başarımına sahip olduğu görülmüştür.

Gelecek alıřmalarda, nerilen modelin bir karar destek sistemine tařınarak tıp doktorlarına beyin MRG grntlerinin sınıflandırılması konusunda kolaylık saęlaması amalanmaktadır. Bu ama doęrultusunda web tabanlı aık kaynak kodlu bir yazılım geliřtirilmesi dřnlmektedir. Geliřtirilen uygulamaya tıp doktoru beyin MRG grntsn ykleyecek ve sonrasında beyin MRG grntsnn hangi sınıfa ait olduęu bilgisini ıktı olarak alabilecektir.

## VI. KAYNAKÇA

### KİTAPLAR

ALPAYDIN, E., (2013). **Yapay Öğrenme**, İstanbul, Boğaziçi Üniversitesi Yayınları, 2. Baskı.

ARI, N., ÖZEN, Ş., ÇOLAK, Ö.H., (2008), **Dalgacık Teorisi**, Ankara, Palme Yayıncılık.

BURKOV, A. (2021). **The Hundred-Page Machine Learning Book**, (Çev. Ali Okatan, Taner Karatekin, Kağan Okatan), İstanbul, Papatya Bilim.

BURRUS, C. S., GOPINATH, R. A., GUO, H. (1998). **Introduction to Wavelets and Wavelet Transforms**. Texas, Prentice Hall.

ELMAS, Ç., (2010). **Yapay Zeka Uygulamaları**, Ankara, Seçkin Yayıncılık, 2. Baskı.

GONZALES, R. C., & WOODS, R. E. (2014). **Sayısal Görüntü İşleme**. (Çev. Z. Telatar, H. Tora, A. Kalaycıoğlu, F. Arı), Ankara, Palme Yayıncılık.

GOODFELLOW, I., BENGIO, Y., COUVILLE, A. (2018). **Derin Öğrenme**. (Çev. F. Yarman Vural, R. G. Cinbiş, S. Kalkan), Ankara, Buzdağı Yayınevi.

NABİYEYEV, V.G., (2012). **Yapay Zeka**, Ankara, Seçkin Yayıncılık, 4. Baskı.

ÖZTEMEL, E. (2006). **Yapay Sinir Ağları**, İstanbul, Papatya Yayıncılık, 2. Baskı.

### MAKALELER

AGGARWAL, N., AGRAWAL R.K., (2012). “First and second order statistics features for classification of magnetic resonance brain images”, **Journal of Signal and Information Processing**, cilt 3, ss.146-153.

AHMED, N., NATARATAN, T., RAO, K., (1974). “Discrete cosine transform”, **IEEE Transactions on Computers**, cilt 23, sayı 1, ss. 90-93.

- ÖZKARACA, O., BAĞRIAÇIK, O.İ., GÜRÜLER, H., KHAN, F., HUSSAIN, J., KHAN, J., LAİLA, U., (2023) "Multiple brain tumor classification with dense CNN architecture using brain MRI images", **Life**, cilt 13, sayı 349.
- ALZUBAIDI, L., ZHANG, J., HUMAIDI, A. J., AL-DUJAILI, A., DUAN, Y., AL-SHAMMA, SANTAMARIA, J., FADHEL, M. A., AL AMIDIE, M., FARHAN, L. (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", **Journal of Big Data**, cilt 8, sayı 53.
- ANITHA, R., SIVA SUNDHARA RAJA, D., (2017), "Segmentation Of Glioma Tumors Using Convolutional Neural Networks", **Int J Imaging Syst Technol.** 2017; cilt 27, sayı 4, ss.354– 360
- ASLAN, M., (2022). "Derin Öğrenme Tabanlı Otomatik Beyin Tümör Tespiti", **Fırat Üniversitesi Müh. Bil. Dergisi**, cilt 34, sayı 1, ss 399-407.
- ASLAN, M., (2022). "Derin Öğrenme Tabanlı Otomatik Beyin Tümör Tespiti", **Fırat Üniversitesi Müh. Bil. Dergisi**, cilt 34, sayı 1, ss.399-407.
- CİHAN, P., KALIPSIZ, O., GÖKÇE, E. (2019). "Yenidoğan Kuzularda Bilgisayar Destekli Tanı", **Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi**, cilt 26, sayı 2, ss.385-391.
- DEAN, J., (2020). "The Deep Learning Revolution And Its Implications For Computer Architecture And Chip Design", **2020 IEEE International Solid-State Circuits Conference-(ISSCC)**.
- EKER, A.G., DURU, N. (2021). "Medikal Görüntü İşlemede Derin Öğrenme Uygulamaları", **Acta Infologica**, cilt 5, sayı 2, ss. 459-474.
- ERSOY, E., KARAL, Ö., (2012). "Yapay Sinir Ağları ve İnsan Beyni", **İnsan ve Toplum Bilimleri Araştırmaları Dergisi**, cilt 1, sayı 2, ss 188-205.
- FIRILDAK, K., TALU, M.F., (2019). "Evrişimsel Sinir Ağlarında Kullanılan Transfer Öğrenme Yaklaşımlarının İncelenmesi", **Anatolian Journal of Computer Science**, cilt 4, sayı 2, ss. 88-95.
- KIZRAK, M.A., BOLAT, B., (2018). "Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma", **Bilişim Teknolojileri Dergisi**, cilt 11, sayı 3, ss. 263-285.



- KRUZHEVSKY, A., SUTSKEVER, I., HUNTON, G. E., (2012). “Imagenet classification with deep convolutional neural networks”, **In Advances in neural information processing systems**, cilt 25, sayı 2, ss. 1097–1105.
- KÜÇÜK, M., AĞIRALIOĞLU, N. (2006). “Dalgacık Dönüşüm Tekniği Kullanılarak Hidrolojik Akım Serilerinin Modellenmesi”, **İTÜ Dergisi**, cilt 9, sayı 1, ss.69-80.
- LECUN, Y., BOSER, B., DENKER, J.S., HENDERSON, D., HOWARD, R.E., HUBBARD, W., JACKEL, L.D. (1989) “Backpropagation Applied to Handwritten Zip Code Recognition”, **Neural Computatuon**, cilt 1, sayı 5, ss. 541-551.
- LECUN, Y., LEON, B., YOSHUA, B., HAFFNER, P., (1998). “Gradient-Based Learning Applied To Document Recognition”, **Proceedings of the IEEE**, cilt 86, sayı 11, ss. 2278-2323.
- MALLAT, S., (1989). “A theory for multiresolution signal decomposition: The wavelet representation”, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, cilt 11, sayı 7, ss.674–693.
- OYAR O., (2008). “Magnetik Rezonans Görüntüleme (Mrg)’Nin Klinik Uygulamaları ve Endikasyonları”, **Harran Üniversitesi Tıp Fakültesi Dergisi**, cilt 5, sayı 2, ss.31-40.
- REHMAN, Z.U., NAQVI, S.S., Khan, T.M., KHAN, M.A., BASHIR, T. (2019). “Fully automated multi-parametric brain tumour segmentation using superpixel based classification”, **Expert Systems with Applications**, cilt 118, ss 598–613.
- ROSENBLATT, F., (1958). “The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain”, **Psychological Review**, cilt 65, sayı 6, ss 386-408.
- Deepak, S., Ameer P.M., (2019). “Brain tumor classification using deep CNN features via transfer learning”, **Computers in Biology and Medicine**, cilt 111.
- SIDDIQI, A.H., SEVİNDİR, H.K. (2012). “A Wavelet-based Energetic Approach for the Analysis of Electroencephalogram”, **Sultan Qaboos University Journal for Science**, cilt 17, sayı 2, ss. 232-244.

- SIEGEL, RL., MİLLER, KD., WAGLE, NS., JEMAL, A., (2023) Cancer statistics, 2023, **CA Cancer J Clin**, cilt 73, sayı 1, ss.17-48.
- SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., WOJNA, Z., (2016). “Rethinking The İnception Architecture For Computer Vision”, **IEEE Conference on Computer Vision and Pattern Recognition**, ss. 2818-2826.
- SOUMIK, F.I., HOSSAIN, A., (2020). “Brain Tumor Classification With İnception Network Based Deep Learning Model Using Transfer Learning”, **2020 IEEE Region 10 Symposium**, ss. 1018-1021
- SWATI Z.N.K., KABIR M., ALİ, F., ALİ, Z., AHMED S., JIANFENG L., (2019). “Content-Based Brain Tumor Retrieval for MR Images Using Transfer Learnin”, **IEEE Access**, cilt 7, ss. 17809-17822.
- YILMAZ, S., KILCI, S.B., (2020). “İleri Sürümlü Yapay Sinir Ağları Eğitim ve Geliştirme Aracı Tasarımı”, **Kocaeli Üniversitesi Fen Bilimleri Dergisi**, cilt 3, sayı 1, ss. 133-146.
- YÜKSEL, Z., (2019). “Manyetik Rezonans Görüntüleme Fizik Temelleri ve Sistem Bileşenleri”, **Black Sea Journal of Engineering and Science**, cilt 2, sayı 2, ss.57-65.
- TÜRKDAMAR, M.U., TAŞYÜREK, M., ÖZTÜRK, C., (2023). “Transfer öğrenmeli ve transfer öğrenmesiz derin ağlar ile inşaat alanında kask tespiti”, **Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi**, cilt 12, sayı 1, ss.39-51.
- GOMEZ-GUZMAN, M.A., JIMENEZ-BERISTAIN, L., GARCIA-GUERRERO, E.E., LOPEZ-BONILLA, O.R., TAMAYO-PEREZ, U.J., ESQUEDA-ELIZONDO, J.J., PALOMINO-VIZCAINO, K., INZUNZA-GONZALEZ, E., (2023). “Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks”, **Electronics**, cilt 12, sayı 955, ss.1-22.
- ALİ, R., AL-JUMAILI, S.A, DURU, A.D., UÇAN, O.N., BOYACI, A., DURU, D.G., “Classification of brain tumors using MRI images based on convolutional neural network and supervised machine learning algorithms”, **2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)**, Ankara, 2022, s.822-827.

## **ELEKTRONİK KAYNAKLAR**

AMİDİ, A., AMİDİ, S., (2019). “Evrişimli Sinir Ağları El Kitabı”,  
<https://stanford.edu/~shervune/1/tr/teachung/cs-230/cheatsheet-convolutuonal-neural-networks/>, (Erişim Tarihi: 29.06.2022).

NICKPARVAR M., “Brain tumor MRI dataset”,  
<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>,  
(Erişim tarihi: 24.05. 2023).

CHENG, J., “Brain Tumor Dataset”,  
[https://figshare.com/articles/brain\\_tumor\\_dataset/1512427/5](https://figshare.com/articles/brain_tumor_dataset/1512427/5), (Erişim Tarihi:  
07.06.2022).

GÖKHAN, H., (2022). “Sayısal İşaret İşleme”.  
[https://acikders.ankara.edu.tr/pluginfile.php/16366/mod\\_resource/content/1/Hafta1\\_AcikDers.pdf](https://acikders.ankara.edu.tr/pluginfile.php/16366/mod_resource/content/1/Hafta1_AcikDers.pdf) (Erişim Tarihi: 9 Mart 2022).

KIZILKAYA, A., “Sayısal Görüntü İşleme”,  
[http://akizilkaya.pamukkale.edu.tr/B%C3%B6l%C3%BCm4\\_goruntu\\_isleme.pdf](http://akizilkaya.pamukkale.edu.tr/B%C3%B6l%C3%BCm4_goruntu_isleme.pdf), (Erişim Tarihi: 20.06.2022).

ORHAN, U., “Makine Öğrenmesi”,  
<https://ceng.cu.edu.tr/uorhan/DersNotu/Ders06.pdf>, (Erişim Tarihi:3 Ekim  
2022).

URL-1 “Advanced Guide to Inception v3”, Google Cloud TPU,  
<https://cloud.google.com/tpu/docs/inception-v3-advanced>, (Erişim Tarihi:3  
Eylül 2022).

## **TEZLER**

AKGÜL, B., (2022). “Derin Öğrenme Tabanlı Yapı Elektrik Plan Çizimi”, (Doktora Tezi), Lisansüstü Eğitim Enstitüsü, Karabük Üniversitesi.

ARI, A., (2019). “Derin Öğrenme Tabanlı Beyin Mr Görüntülerinden Beyin Tümörlerinin Tespit Edilmesi ve Sınıflandırılması”, (Doktora Tezi), Fen Bilimleri Enstitüsü, İnönü Üniversitesi.

- ARIKAN KARGI, V.S., (2013). “Yapay Sinir Ağ Modelleri ve Bir Tekstil Firmasında Uygulama”, (Doktora Tezi), Sosyal Bilimleri Enstitüsü, Uludağ Üniversitesi.
- AYDOĞDU, Ö. (2021). “Akan Veride Artımlı Öğrenme Yaklaşımları Geliştirilmesi ile Video Verilerinde Anomali Tespiti Çalışması”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Karadeniz Teknik Üniversitesi.
- AYTAÇ, U. C., (2022). “Derin Öğrenme Yöntemlerinin Tıbbi Teşhis Alanında Sorgulanması ve Beyin Tümörü Tanısında Uygulanması”, (Doktora Tezi), Fen Bilimleri Enstitüsü, İstanbul Aydın Üniversitesi.
- BİLGİN, S., (2008). “Kalp Hızı Değişkenliğinin Dalgacık Dönüşümü ve Yapay Sinir Ağları Kullanılarak Analizi”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Sakarya Üniversitesi.
- BÜLBÜL, M.A., (2022). “Akıllı Sulama Sistemi Modellemesi ve Tasarımı”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Erciyes Üniversitesi.
- ÇİFÇİ, M.A., (2021). “Derin Öğrenme ve Dalgacık Dönüşümüne Dayalı Akciğer Kanseri Erken Tanısında Belirsizlik Ölçeğinin Saptanması”, (Doktora Tezi), Lisansüstü Eğitim Enstitüsü, İstanbul Aydın Üniversitesi.
- DUMAN, E.A., (2021). “Derin Öğrenme Yöntemleri İle Beyin MRI Görüntülerinde Anomali Tespiti”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Gazi Üniversitesi.
- EMEK, S. (2006). “Sabit Görüntüler ve Video İşaretleri İçin Ayrık Dalgacık Dönüşümü Ayrık Kosinüs Dönüşümü Tabanlı Sayısal Damgalama Yöntemi”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Yıldız Teknik Üniversitesi.
- KAPLAN, K., (2020). “Beyin Tümör Tiplerinin Makine Öğrenmesi ve Derin Öğrenme Tabanlı Teknikler ile Sınıflandırılması”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Kocaeli Üniversitesi.
- KAZDAL, S., (2013). “Beyin Tümörlerinin İleri Görüntü İşleme ve Örüntü Tanıma Teknikleri Kullanılarak Bilgisayar Destekli Tespiti”, (Yüksel Lisans Tezi), Fen Bilimleri Enstitüsü, Marmara Üniversitesi.
- KILIÇ, Ş., (2021). “Derin Öğrenme Yöntemleri Kullanılarak Giyilebilir Sensörlerden Kişi Tanıma”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Ankara Üniversitesi.

- KOÇER, B., (2012). “Transfer Öğrenmede Yeni Yaklaşımlar”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.
- KURT, F., (2018). “Evrışimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi”, (Yüksek Lisans Tezi), Fen Bilimleri Enstitüsü, Hacettepe Üniversitesi.
- MERAL, T., (2013). “Veri Analizinde Dalgacık Teorisinin Etkinliği”, (Yüksel Lisans Tezi), Fen Bilimleri Enstitüsü, Kocaeli Üniversitesi.
- MERCAN, C.A., (2015). “An Approach For The Chest Tube Detection In Chest Radiographs Using Convolutional Neural Network”, (Doktora Tezi), Bilişim Enstitüsü, İstanbul Teknik Üniversitesi.
- ÖĞÜCÜ, M.O., (2006). “Yapay Sinir Ağları ile Sistem Tanıma”, (Yüksel Lisans Tezi), Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi.
- SALUR, M. U., (2021). “Derin Öğrenme Tabanlı Çok Modlu Duygu Analizi Yöntemlerinin Geliştirilmesi”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Fırat Üniversitesi.
- SÜRÜCÜ, M., (2021). “Normalize Edilmiş Kalp Hızı Değişkenliği Analizi ile Paroksizmal Atriyal Fibrilasyon Tespiti”, (Doktora Tezi), Lisansüstü Eğitim Enstitüsü, Düzce Üniversitesi.
- YAVUZ, A., (2021). “Derin Öğrenme Algoritmaları ile Trafik İşaret ve Levhalarının Tanımlanması”, (Yüksek Lisans Tezi), Fen Bilimleri Enstitüsü, Pamukkale Üniversitesi.
- YURTSEVER, U., (2019). “Bölütlenmiş Histopatolojik Görüntüler Üzerinde Derin Öğrenme Yöntemiyle Kolon Kanseri Tespiti”, (Doktora Tezi), Fen Bilimleri Enstitüsü, Sakarya Üniversitesi.

## EKLER

**EK A:** Kaynak Kodlar

### - <https://github.com/zihnikaya/beyin-tumor-siniflandirma>

```
import tensorflow as tf
```

```
import numpy as np
```

```
import os
```

```
import cv2
```

```
from sklearn.metrics import  
precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix
```

```
from sklearn.preprocessing import StandardScaler
```

```
import pywt
```

```
scaler=StandardScaler()
```

```
x_dataset=[]
```

```
y_dataset=[]
```

```
src=os.listdir('data_big/notumor')
```

```
i=0
```

```
for filename in src:
```

```
    full_file_name='data_big/notumor/'+filename
```

```
    img=cv2.imread(full_file_name)
```

```
    b, g, r = cv2.split(img)
```

```

b = cv2.equalizeHist(b)
g = cv2.equalizeHist(g)
r = cv2.equalizeHist(r)

LLb, (LHb, HLb, HHb) = pywt.dwt2(b, 'db2')
LLg, (LHg, HLg, HHg) = pywt.dwt2(g, 'db2')
LLr, (LHr, HLr, HHr) = pywt.dwt2(r, 'db2')
LL = cv2.merge((LLb, LLg, LLr))
x=np.array(LL,dtype='float32')
x = scaler.fit_transform(x.reshape(-1, x.shape[-1])).reshape(x.shape)
x = cv2.resize(x, (224, 224))
y=0
y=np.array(y,dtype='float32')
x_dataset.append(x)
y_dataset.append(y)
i=i+1
print(i)

src=os.listdir('data_big/glioma')
for filename in src:
    full_file_name='data_big/glioma/'+filename
    img = cv2.imread(full_file_name)

    b, g, r = cv2.split(img)

    b = cv2.equalizeHist(b)

```

```

g = cv2.equalizeHist(g)
r = cv2.equalizeHist(r)

LLb, (LHb, HLb, HHb) = pywt.dwt2(b, 'db2')
LLg, (LHg, HLg, HHg) = pywt.dwt2(g, 'db2')
LLr, (LHr, HLr, HHr) = pywt.dwt2(r, 'db2')
LL = cv2.merge((LLb, LLg, LLr))
x = np.array(LL, dtype='float32')
x = scaler.fit_transform(x.reshape(-1, x.shape[-1])).reshape(x.shape)
x = cv2.resize(x, (224, 224))
y=1
y=np.array(y,dtype='float32')
x_dataset.append(x)
y_dataset.append(y)
i=i+1
print(i)

src=os.listdir('data_big/meningioma')
for filename in src:
    full_file_name='data_big/meningioma/'+filename
    img = cv2.imread(full_file_name)

    b, g, r = cv2.split(img)

    b = cv2.equalizeHist(b)
    g = cv2.equalizeHist(g)

```



```

r = cv2.equalizeHist(r)

LLb, (LHb, HLb, HHb) = pywt.dwt2(b, 'db2')
LLg, (LHg, HLg, HHg) = pywt.dwt2(g, 'db2')
LLr, (LHr, HLr, HHr) = pywt.dwt2(r, 'db2')
LL = cv2.merge((LLb, LLg, LLr))
x=np.array(LL,dtype='float32')
x = scaler.fit_transform(x.reshape(-1, x.shape[-1])).reshape(x.shape)
x = cv2.resize(x, (224, 224))
y=2
y=np.array(y,dtype='float32')
x_dataset.append(x)
y_dataset.append(y)
i=i+1
print(i)

```

```

src=os.listdir('data_big/pituitary')
for filename in src:
    full_file_name='data_big/pituitary/'+filename
    img = cv2.imread(full_file_name)

    b, g, r = cv2.split(img)

    b = cv2.equalizeHist(b)
    g = cv2.equalizeHist(g)
    r = cv2.equalizeHist(r)

```

```

LLb, (LHb, HLb, HHb) = pywt.dwt2(b, 'db2')
LLg, (LHg, HLg, HHg) = pywt.dwt2(g, 'db2')
LLr, (LHr, HLr, HHr) = pywt.dwt2(r, 'db2')
LL = cv2.merge((LLb, LLg, LLr))
x=np.array(LL,dtype='float32')
x = scaler.fit_transform(x.reshape(-1, x.shape[-1])).reshape(x.shape)
x = cv2.resize(x, (224, 224))
y=3
y=np.array(y,dtype='float32')
x_dataset.append(x)
y_dataset.append(y)
i=i+1
print(i)

```

```

X_dataset=np.array(x_dataset)
Y_dataset=np.array(y_dataset)
Y_dataset=Y_dataset.reshape((7023,1))

```

```

from sklearn.model_selection import KFold
folds=list(KFold(n_splits=5,shuffle=True,random_state=1).split(X_dataset,Y_dataset
))

```

```

Inception=tf.keras.applications.InceptionV3(include_top=False,input_shape=(224,22
4,3))
input_image=tf.keras.layers.Input((224,224,3))
x=Inception (input_image)

```

```

x=tf.keras.layers.Flatten()(x)

hl=tf.keras.layers.Dense(units=3, activation='relu')(x)

x=tf.keras.layers.Dense(4)(hl)

out=tf.keras.layers.Activation(activation='softmax')(x)

callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)

model=tf.keras.Model(inputs=input_image,outputs=out)

model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.0001),loss='
sparse_categorical_crossentropy',metrics=['accuracy'])

model.summary()

for j, (train_idx, val_idx) in enumerate(folds):

    print("Fold " + str(j + 1))

    x_train = X_dataset[train_idx]

    y_train = Y_dataset[train_idx]

    x_val = X_dataset[val_idx]

    y_val = Y_dataset[val_idx]

    history = model.fit(x_train, y_train, batch_size=32, epochs=100,
callbacks=[callback], validation_data=(x_val, y_val))

    y_predict = model.predict(x_val)

    Y_predict = np.argmax(y_predict, axis=1)

    l = len(y_val)

    Y_predict = np.reshape(Y_predict, (l, 1))

    prescore = precision_score(y_val, Y_predict, average=None)

    recaller = recall_score(y_val, Y_predict, average=None)

    score = f1_score(y_val, Y_predict, average=None)

    con_mat = confusion_matrix(y_val, Y_predict)

```

```
print("Precision: ", prescore)
print("Recall: ", recaller)
print("F1-score ", score)
print("Confusion Matrix ", con_mat)
```

## **ÖZGEÇMİŞ**

**Ad-Soyad**

Zihni KAYA

### **ÖĞRENİM DURUMU**

**Lisans**

2004, Akdeniz Üniversitesi, İİBF, Maliye

**Yüksek Lisans**

2016, İstanbul Aydın Üniversitesi, Fen Bilimleri  
Enstitüsü, Bilgisayar Mühendisliği