



Sosyal Bilimler
Enstitüsü

T.C
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İKTİSAT ANABİLİM DALI
EKONOMETRİ DALI

**DERİN ÖĞRENME YÖNTEMLERİ İLE BEYİN TÜMÖRÜ TESPİTİNE
İLİŞKİN BİR ANALİZ**

Dönem Projesi

EREN YALÇIN

İstanbul, 2024



**Sosyal Bilimler
Enstitüsü**

T.C
MARMARA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İKTİSAT ANABİLİM DALI
EKONOMETRİ BİLİM DALI

**DERİN ÖĞRENME YÖNTEMLERİ İLE BEYİN TÜMÖRÜ TESPİTİNE
İLİŞKİN BİR ANALİZ**

Dönem Projesi

EREN YALÇIN

Danışman : Prof. Dr. İLKNUR ESEN YILDIRIM

İstanbul, 2024

KABUL VE ONAY SAYFASI

ÖZET

Beyinde diğ er normal h creleri kontrol eden, mekanizmalar tarafından kontrol edilemeyen, kontrols z bir  ekilde b y yen anormal doku k tlesine beyin t m r  adı verilir. En  nemli belirtileri;  iddetli ba  ağrısı, bulantı ve kusmadır. Genel olarak, bir beyin t m r  te hisinde manyetik rezonans g r nt leme (MRG) cihazları kullanılır. MRG cihazları beyinde bir t m r n  tespit ettikten sonra, beyin t m r n n tipini belirlemenin en yaygın yolu biyopsi ve/veya ameliyat sonrası alınan doku  rneđi sonu larına bakmaktır. Erken te his burada b y k bir  neme sahiptir. Son yıllarda geli tirilen yapay zeka teknolojileri ile beraber beyin vb. t m r tespitleri daha kolay bir hale gelmi tir. Bu  alı manın amacı da tıbbi segmentasyon y ntemlerini kullanarak makine  ğrenmesi teknikleri ile MRG verilerini analiz edip t m r n varlığını tespit etmektir. Derin  ğrenmenin alt y ntemlerinden biri olan CNN y nteminin VGG-16 modeli kullanılmı tır. İ lem sonucunda dođruluk: %92, kesinlik: %90, duyarlılık: %94 ve f1 skor: %92 olarak hesaplanmı tır.  alı manın sonucunda VGG-16 modelinin diğ er  alı malar ile kar ıla tırıldığında gayet ba arılı bir sonu   ıkardığı g r lmektedir.

Anahtar Kelimeler: Beyin T m r , Makine  ğrenmesi, Derin  ğrenme, VGG-16, CNN, MRG

ABSTRACT

An abnormal tissue mass that grows uncontrollably and cannot be controlled by the mechanisms that control other normal cells in the brain is called a brain tumor. The most important symptoms are; severe headache, nausea and vomiting. Generally, magnetic resonance imaging (MRI) devices are used to diagnose a brain tumor. Once MRI devices detect a tumor in the brain, the most common way to determine the type of brain tumor is to look at the results of a biopsy and/or tissue sample taken after surgery. Early diagnosis is of great importance here. With the artificial intelligence technologies developed in recent years, brain etc. Tumor detection has become easier. The aim of this study is to analyze MRI data with machine learning techniques using medical segmentation methods and detect the presence of tumor. The VGG-16 model of the CNN method, one of the sub-methods of deep learning, was used. As a result of the process, accuracy: 92%, precision: 90%, sensitivity: 94% and f1 score: 92%. As a result of the study, it is seen that the VGG-16 model produces a very successful result compared to other studies.

Key Words: Brain Tumor, Machine Learning, Deep Learning, VGG-16, CNN, MRI

İÇİNDEKİLER

	Sayfa No.
ÖZET	I
ABSTRACT	ii
İÇİNDEKİLER.....	iii
KISALTMALAR.....	IV
TABLolar LİSTESİ	V
ŞEKİLLER LİSTESİ	VI
GRAFİKLER LİSTESİ	VII
GİRİŞ.....	1
LİTERATÜR TARAMASI	2
1. MATERYAL VE METEDOLOJİ	4
1.1 Makine Öğrenmesi ve Derin Öğrenme	4
1.2 CNN Modeli Nedir ve Nasıl Çalışır	5
1.3 Veri Seti Açıklaması	7
2. UYGULAMA	8
2.1 Veri Ön İşleme	8
2.2 CNN Model Oluşturma	12
2.3 VGG-16 Modelinin Kurulması	13
2.4 Modelin Metriklerinin Ölçülmesi	15
SONUÇ	21
KAYNAKÇA	22

KISALTMALAR

MRG	Manyetik Rezonans Görüntüleme
BT	Bilgisayarlı Tomografi
PET	Pozitron Emisyon Tomografisi
CNN	Convolutional Neural Networks
VGG	Visual Geometry Group
ANN	Yapay Sinir Ağları
RNN	Tekrarlayan Sinir Ağları
LSTM	Uzun Kısa Vadeli Bellek
GAN	Generative Adversarial Networks
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

TABLÖLER LİSTESİ

Tablo 1: Örnek Sınıflandırma Görüntüleri	9
Tablo 2: Adım Adım Pyimagesearch'ün Nasıl Çalıştığı.....	11
Tablo 3: Data Augmentation'un Çıktıları ve Örnek Görseller	12
Tablo 4: Flatten, Dropout ve Dense Çıktıları	14
Tablo 5: Cohen Kappa Skoru Açıklama	16
Tablo 6: Test Verisi Doğruluk Oranı	17
Tablo 7: Cohen Kappa Skoru Açıklama	19
Tablo 8: Beyin Tümörü Bulunmamasına Rağmen Varmış Gibi Sınıflandırılan Görseller	20

ŞEKİLLER LİSTESİ

Şekil 1: Beyin Tümörü Belirtileri	1
Şekil 2: Yapay Zekâ Katmanları, Makine Öğrenmesi ve Derin Öğrenme İlişkisi	4
Şekil 3: Geleneksel Öğrenme ve Öğrenme Aktarımı Arasındaki Fark	5
Şekil 4: CNN Model Oluşturma	6
Şekil 5: CNN'in Bir Görsel Üzerinden Gösterimi	6
Şekil 6. Çalışmanın Aşamaları	13

GRAFİKLER LİSTESİ

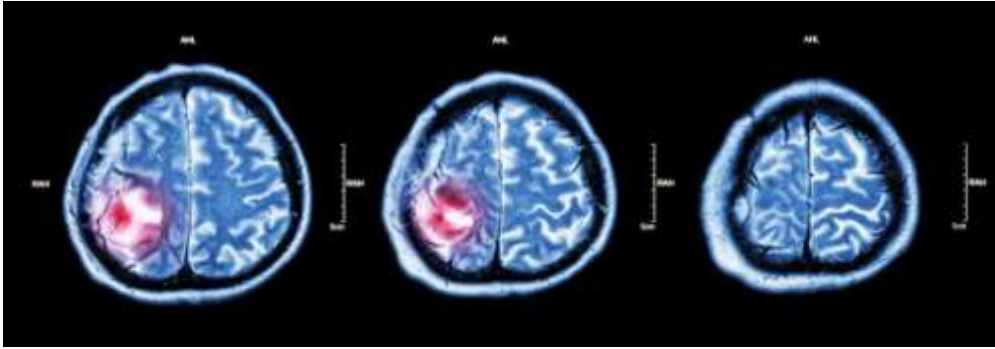
Grafik 1: Her Setteki Sınıfların Sayısı	8
Grafik 2: Görüntü Oranlarının Dağılımı	10
Grafik 3: Sigmoid, Tanh, ReLu Karşılaştırma	14
Grafik 4: VGG-16 Kayıp ve Doğruluk Grafiği	16
Grafik 5: ROC AUC Eğrisi	19

GİRİŞ

Kafatası bölgesinde kontrolsüzce büyüyen hücreler beyin tümörü olarak adlandırılmaktadır. Bu tümörlü hücreler, beynin içinde oluşabildiği gibi beynin çevresindeki dokulardan veya vücuttaki başka organlardan da kaynaklanabilir, kafatasının içine doğru büyüyebilir. Kafatasının içine yerleşen tümörler basınç artışına bağlı olarak, belirgin belirtiler gösterebilir. Beyin tümörlerinin en önemli belirtileri; şiddetli baş ağrısı, bulantı ve kusmadır. Dünyada nüfustaki yaşlılık oranı arttıkça beyin tümörlerinin görülme oranı da artmaktadır. Birçok kanser türünde de olduğu gibi, beyin tümörlerinin hangi sebeplerden geliştiğine dair net bir açıklama bulunmamakla birlikte beyin tümörleri ve yaş ilişkisi üzerine yapılan çalışmalara göre; dünya nüfusu yaşlandıkça beyin tümörlerinin görülme olasılığı artmaktadır. Ancak beyin tümörü her yaş grubunda görülebilmektedir.

Beyin tümörünün tespitine ilişkin çalışmalarda, manyetik rezonans görüntüleme (MRG), bilgisayarlı tomografi (BT) ve kafa biyopsisi gibi yöntemler yaygın olarak kullanılmaktadır. MRG yöntemi, beynin yumuşak dokusu hakkında yüksek çözünürlüklü bilgi sağladığı gibi aynı dokuya ait birden fazla ilişkisel bilgi de sunar. Bu ve benzeri nedenlerle beyin cerrahları, beyindeki en küçük anormalliklerin tespiti için yeterli bilgi sağladığından MRG cihazlarını yaygın olarak kullanmaktadır.

Beynin yapısı ve tümörün bulunduğu bölge, tümörün türü ile bu tümörden beynin, sinirlerin ve diğer organların ne kadar etkilendiği konusunda detaylı bilgi sağlayabilir. Bunların dışında anjiyografi ve PET gibi farklılaştırılmış bazı yöntemlere de ihtiyaç duyulabilmektedir. Bunlar ile birlikte tümörün kanlanma şekilleri, kötü huylu olup olmadığı gibi pek çok konuda da bilgi elde edilebiliyor.



Şekil 1. Beyin Tümörü Belirtileri

Kaynak: Neolife (t.y.), “Beyin Tümörü Belirtileri” <https://www.neolife.com.tr/beyin-tumoru-belirtileri/> adtesinden alındı.

LİTERATÜR TARAMASI

Literatürde beyin tümörü tespitine dair çalışmalarda makine öğrenmesi yaklaşımları sıkça kullanılmıştır.

-2023 yılında O. Kırılı, M. Sansarcı, O. Özkara ve G. Çetin tarafından hazırlanan çalışmada Doğrusal-DVM (doğrusal- destek vektör makineleri) algoritması ile %84,4 doğruluk oranı ve rassal orman (random forrest) algoritması ile %81,5 doğruluk oranı elde edilmiştir.

-P. Afshar, A. Mohammadi ve K. Plataniotis'in 2018 yılındaki çalışmasında beyin tümörünü sınıflandırmak için CapsNet adı verilen bir yöntem geliştirildi. Bu geliştirilen yöntem ile daha düşük veri sayısı ile daha yüksek doğruluk oranı amaçlandı. Tek evrişim katmanı ile 64 öznitelik kullanıldı ve %86,56 doğruluk oranı elde edildi.

-M. Oğuz Taş ve S. Ergin'in 2020 yılındaki çalışmasında, MR görüntülerinde beyin tümörü tespitinde k-ortalama (k-means)sınıflandırmayı kapsayan bir yöntem kullanıldı. %84,45 doğruluk, %95,04 hassasiyet ile sonuçlandı.

-Vani, Sowya ve Jayamma'nın 2017'deki çalışmasında, makine öğrenmesi tabanlı Destek Vektörü Makinesi (DVM) yöntemi kullanıldı. Çalışma, beyin tümörünü pozitif olarak %82 negatif olarak %81,48 doğrulukla tahmin edildi.

-Shahzadi, Tang, Meriadeau'nun 2016'deki çalışmasında, ESA tabanlı hibrit bir model ile oluşturulan AlexNet ve VggNet kullanılan çalışmada AlexNet-UKSB (Uzun Kısa Süreli Bellek) ile %71 doğruluk, VggNet-UKSB ile de %84 doğruluk elde edildi.

-2019'da hazırlanan başka bir çalışmada ise yine ALexNet %89,95 doğruluk elde ederken, Vgg16 %94,65 ve Vgg19 %94,82 doğruluk oranlarına ulaştı.

-S. Sedlar'ın 2018 yılında yayımladığı çalışmada, çok yollu CNN (Convolutional Neural Network) temelli otomatik bir beyin tümörü sınıflandırma yöntemi tanıtılmıştır. Bu modelin eğitimi ve hiper-parametre ayarları BraTS (Brain Tumor Segmentation) eğitim veri kümesi ile gerçekleştirilmiştir. Modelin doğrulama işlemleri BraTS doğrulama (validation) veri seti üzerinde yapılmış olup, nihai sonuçlar BraTS test veri seti kullanılarak raporlanmıştır. Test veri seti üzerinde sonuçlanan ortalama skorlar; tümör için %60,49, tüm tümör için %84,36 ve tümör çekirdeği için %69,38 olarak belirlenmiştir.

- Balaji'nin 2021 yılında gerçekleştirdiği çalışmada, Generative Deep Belief modeli kullanılarak insan vücudundaki tümörlerin teşhis edilmesi hedeflenmiştir. Bu çalışmada, geleneksel derin ResNet modeli %92, AdaResU-Net modeli %89 ve HSMA-WOA modeli %90 doğruluk oranları elde etmiştir. Ancak, önerilen yöntem %99,9 doğruluk oranı ile bu geleneksel yöntemlerin hepsinden daha üstün bir performans göstermiştir.

Genel olarak geleneksel yöntemlerin bu alanda yetersiz olduđu bu nedenle Transfer Learning gibi yöntemlerin daha işlevsel olduđu sonucuna varılabilir.

Bu çalışmada, beyin tümörünün tespiti için CNN (Convolutional Neural Networks) tabanlı VGG-16 modeli kullanılmıştır. VGG-16, temel olarak basit bir ağı modelidir. Önceki derin öğrenme modellerinden en önemli farkı, VGG-16'nın evrişim katmanlarını 2'li veya 3'lü gruplar halinde kullanmasıdır. Diğer modellerde olduğu gibi, VGG-16'da da girişten çıkışa doğru ilerledikçe matrislerin genişlik ve yükseklik boyutları azalırken derinlik (kanal sayısı) artar.

1. MATERYAL VE METEDOLOJİ

1.1- Makine Öğrenmesi Ve Derin Öğrenme

Makine öğrenmesi en temelinde, insan müdahalesi olmadan verilerden matematiksel ve istatistiksel yöntemleri kullanarak çıkarımlar yapan, bu çıkarımlarla bilinmeyen sonuçlara dair tahminlerde bulunan bir algoritma bütünüdür.

Makine öğrenmesi, sonuç sağlamak için iki temel teknik kullanır. İlki, modelin gelecekteki gereksinimleri tahmin edebilmek ve kendi başına öğrenebilmesi için ilgili girdi ve çıktı verilerine dayalı bir modelin eğitimini ifade eden denetimli öğrenmedir. Öte yandan, denetimsiz öğrenme, bütün bilgileri aramasına ve verilerdeki gizli kalıpları ve eğilimleri bulmasını sağlar.

Derin öğrenmede ise model, yalnızca tahmin etmeyi değil, aynı zamanda ham verilerden özelliklerin nasıl çıkarılacağını da öğrenir.

Makine öğrenmesi, karar vermek amacıyla önceden yüklenmiş bilgileri kullanan bir yapay zeka alt kategorisidir. Derin öğrenme, bundan daha da derin olan yapay zeka biçimidir. Bu teknoloji, büyük miktarda veriden kalıpları öğrenmek ve bu kalıpları almak için derin sinir ağlarını kullanır.



Şekil 2. Yapay Zekâ Katmanları, Makine Öğrenmesi ve Derin Öğrenme İlişkisi

Kaynak: Ateş, E. (2021), “Yapay Zekâ (Artificial Intelligence)”, Jandarma ve Sahil Güvenlik Akademisi. Siber Ansiklopedi: Siber Ortama Çok Disiplinli Bir Yaklaşım, 644-647.

Birçok derin öğrenme metodu bulunmaktadır. Bunlar:

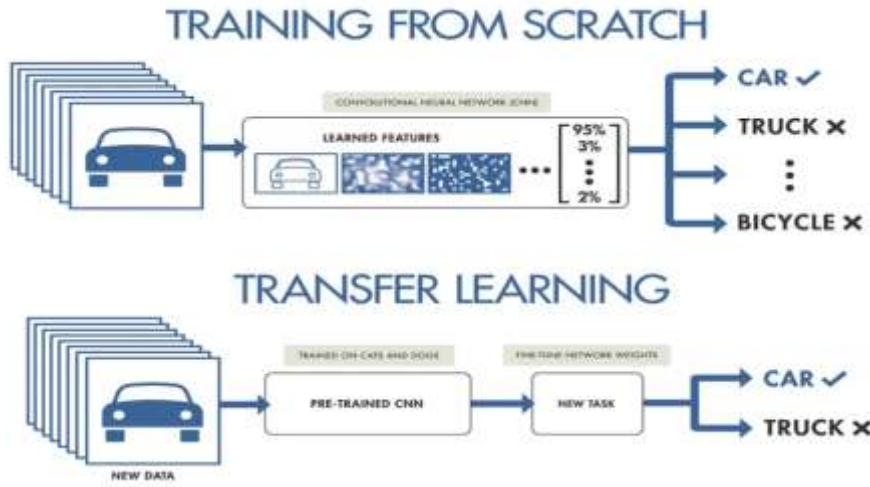
1. Yapay Sinir Ağları (ANN)
2. Evrişimli Sinir Ağları (CNN)
3. Tekrarlayan Sinir Ağları (RNN)
4. Uzun Kısa Vadeli Bellek (LSTM)
5. Generative Adversarial Networks (GAN)

Bu çalışmada CNN metodu üzerine duracağız.

1.2- CNN Modeli Nedir Ve Nasıl Çalışır

Temel model olarak CNN'in alt dallarından olan VGG-16 mimarisi ve ağırlıkları ile Transfer Learning mimarisini kullandım.

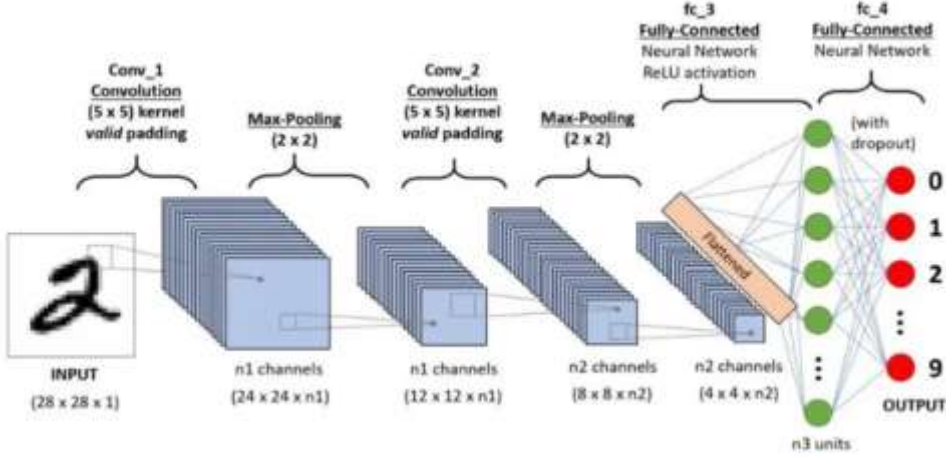
Öğrenme Aktarımı (Transfer Learning): Makine öğrenmesi yöntemlerinden biri olan, bir problemi çözerken elde ettiği bilgiyi saklayarak başka bir problemi çözerken bu bilgiyi kullanmasına dayanan bir yaklaşımdır. Öğrenme aktarımı sayesinde, önceki bilgilerden yararlanılarak daha az eğitim verisiyle daha yüksek başarı sağlayan ve daha hızlı öğrenen modeller geliştirilebilir.



Şekil 3. Geleneksel Öğrenme ve Öğrenme Aktarımı Arasındaki Fark

Kaynak: Şener, M. (2023, 14 Kasım), "Makine Öğrenmesinde Aktarımlı Öğrenme Yaklaşımı ve Aktarımlı Öğrenme Modelleri". <https://blog.architechtt.com/blog/2468/makine-ogrenmesinde-aktarimli-ogrenme-yaklasimi-ve-aktarimli-ogrenme-modelleri/> adresinden alındı.

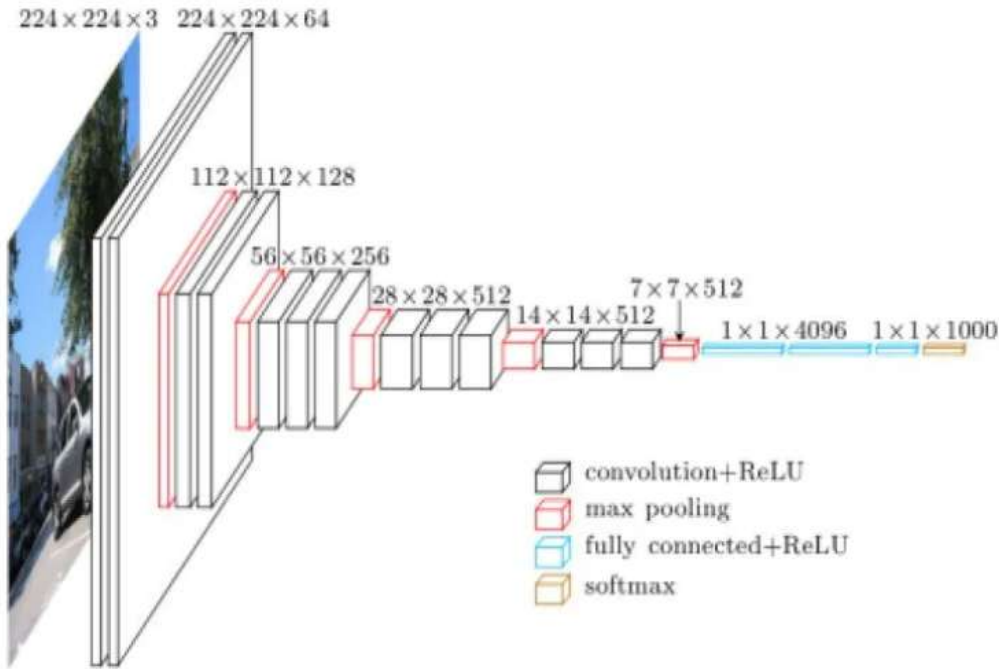
CNN (Convolutional Neural Networks), görselleri girdi olarak alan bir derin öğrenme algoritmasıdır. Bu algoritma, görsellerdeki özellikleri farklı operasyonlarla yakalayıp sınıflandırır ve 3 farklı katmandan oluşur. Convolutional Layer, Pooling Layer ve Fully Connected Layer. Çok etiketli yapılarda softmax katmanı da bulunur. Görsel, bu katmanlardan geçerek çeşitli işlemlerle derin öğrenme modeline uygun hale getirilir. Temel döngüsü ise aşağıdaki gibidir:



Şekil 4. CNN Model Oluşturma

Kaynak: Bozkurt, B. (2021, 30 Eylül), “CNN (Convolutional Neural Networks) Nedir?”. <https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1> adresinden alındı

VGG, genel olarak bir konvansiyonel sinir ağının başarımını artırmak amacıyla daha derinleştirilmesi tekniğine dayanır. Bu mimari, 13 evrişimli katman ve 3 tam bağlı katmandan oluşur. MaxPooling, Fully Connected Layer, ReLU Layer, DropOut Layer ve Softmax Layer katmanlarıyla birlikte toplam 41 katman içermektedir. Girdi katmanında yer alacak görüntünün boyutları 224x224x3 (yükseklik, genişlik ve derinlik) olarak belirlenmiştir. Son katman ise sınıflandırma katmanıdır. Yapısı şu şekildedir:



Şekil 5. CNN’in Bir Görsel Üzerinden Gösterimi

Kaynak: Savaş, S. (2020, 1 Haziran), “Derin Öğrenme Mimarileri”. <https://medium.com/yapay-zeka-makine-ogrenmesi-derin-ogrenme/derin-ogrenme-78e3f5d2765> adresinden alındı.

1.3- Veri Seti Açılması

Bu çalışmada kullanılan veri seti, Kaggle’da bulunan ‘Brain MRI Images for Brain Tumor Detection’ ve ‘Brain Tumor Classification (MRI)’ adlı veri setlerinin birleşiminden oluşturulmuştur. Toplamda 1142 adet MRI (MR, MRG) görseli bulunmaktadır. Veriler, tümör var-yok şeklinde ikiye ayrılmıştır. Beyin tümörü olan görsel sayısı 714 iken, beyin tümörü olmayan görsel sayısı 428’dir. Görseller kendi içlerinde eğitim (train), test ve doğrulama (validation) verileri olarak ayrıştırılması gerekmektedir.

2. UYGULAMA

2.1- Veri Ön İşleme

Eğitim seti: Temel modelleme denemeleri yapılarak en uygun makine öğrenmesi algoritması belirlenmeye çalışılır. Veri miktarına bağlı olarak, verilerin %60-%90'ı eğitim verisi olarak kullanılır.

Doğrulama seti: Doğrulama aşamasında, uygulanan modelin performansı iyileştirilmeye çalışılır. Bu süreçte, hiperparametre ayarlamaları yapılarak en uygun katsayılar ve ağırlıklar bulunmaya çalışılır.

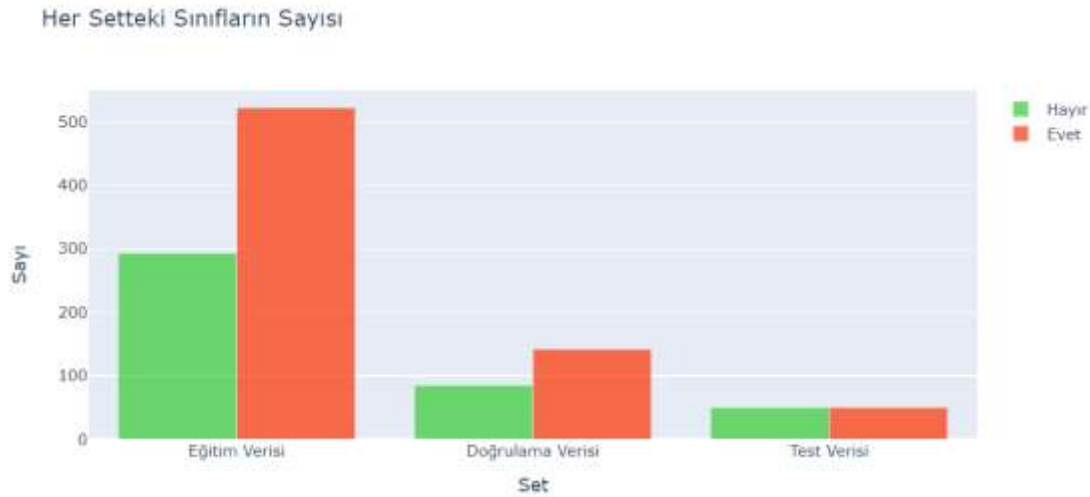
Test seti: Testte tahminler ile gerçek veriler karşılaştırılır.

```
100% ██████████ 2/2 [00:01<00:00, 1.83it/s]
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:18: VisibleDeprecationWarning:
Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different
lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray

50% ██████████ 1/2 [00:00<00:00, 9.61it/s]
815 images loaded from TRAIN/ directory.
100% ██████████ 2/2 [00:00<00:00, 8.83it/s]
50% ██████████ 1/2 [00:00<00:00, 6.00it/s]
100 images loaded from TEST/ directory.
100% ██████████ 2/2 [00:00<00:00, 3.58it/s]
227 images loaded from VAL/ directory.
```

İşlem sonucunda 815 görsel eğitim verisi, 100 görsel test verisi, 227 görsel doğrulama verisi olarak kullanılacağı kesinleşmiş oldu.

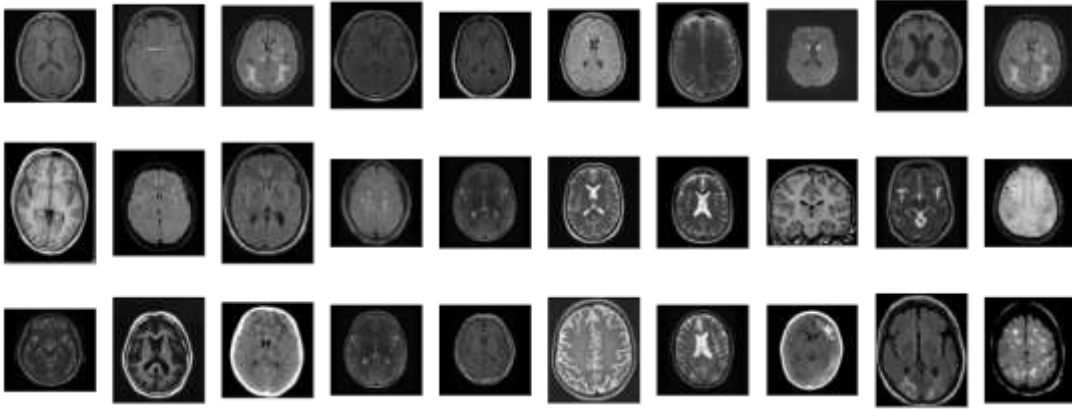
Her setteki sınıf sayısı grafiği:



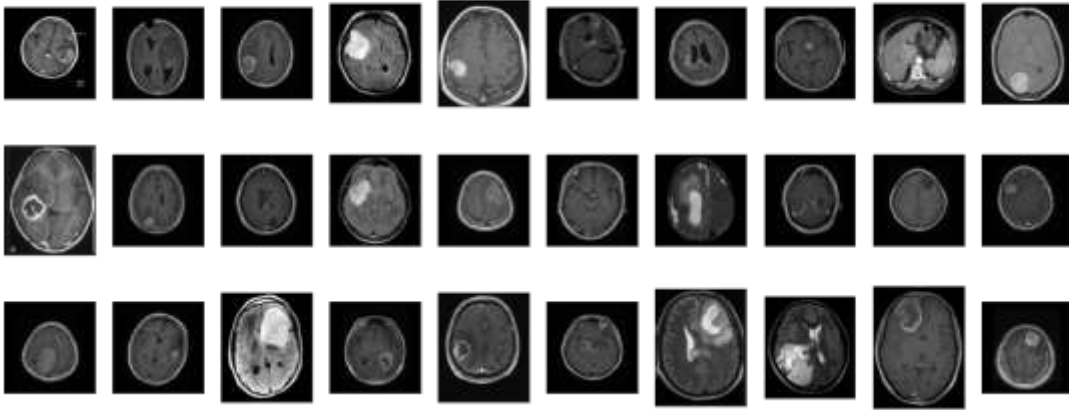
Grafik 1. Her Setteki Sınıfların Sayısı

Belirlenen kümeden istenen sayıda görüntü (n) için bir gridplot grafiği oluşturuldu. Tablo aşağıdaki gibidir;

Tumor: NO

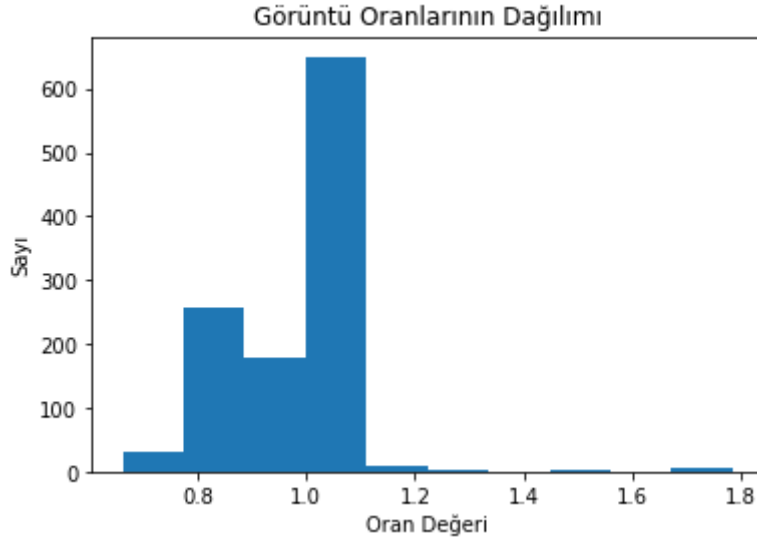


Tumor: YES



Tablo 1. Örnek Görüntüler. Beyin Tümörü Var ve Yok Şeklinde Sınıflandırılmıştır.

Görüldüğü gibi, görseller farklı genişlik, yükseklik ve farklı siyah köşelere sahiptir. VGG-16 algoritması, giriş katmanı için görüntü boyutunu (224,224) olarak sınırlamaktadır. Bu nedenle bazı geniş ve dar görüntüler yeniden boyutlandırılmalı, katmana uygun hale getirilmelidir. Genel olarak bu dağılımı incelemek için oluşturulan histogram grafiği;



Grafik 2. Görüntü oranlarının dağılımı

(oran = genişlik / yükseklik)

Görüldüğü üzere oranı yüksek olan görseller için bir normalleştirme işlemi uygulamak gerekmektedir. Normalleştirme için ise ‘pyimagesearch’ kullandım.

Pyimagesearch, OpenCV destekli bir bilgisayarlı görüntü uygulamasıdır. Temelde görseldeki uç noktaları bulur ve bunlardan dikdörtgenler kırpar. Bu şekilde MR görsellerindeki fazla kısımlar kırpılmış olur.

Kod:

```
def crop_imgs(set_name, add_pixels_value=0):
    set_new = []
    for img in set_name:
        gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
        gray = cv2.GaussianBlur(gray, (5, 5), 0)

        thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
        thresh = cv2.erode(thresh, None, iterations=2)
        thresh = cv2.dilate(thresh, None, iterations=2)

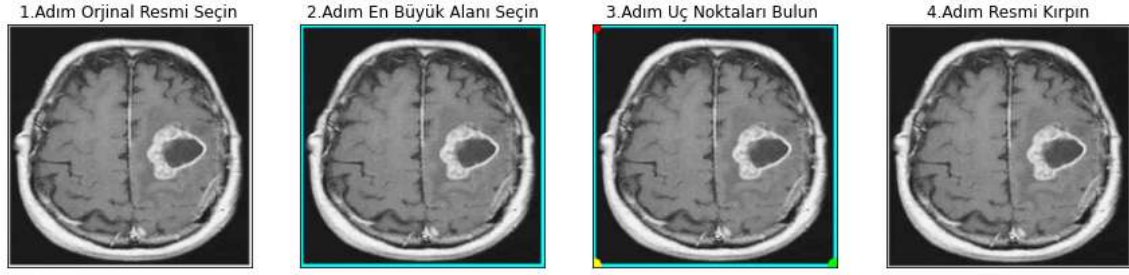
        cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        cnts = imutils.grab_contours(cnts)
        c = max(cnts, key=cv2.contourArea)

        # uç noktaları bul
        extLeft = tuple(c[c[:, :, 0].argmin()][0])
        extRight = tuple(c[c[:, :, 0].argmax()][0])
        extTop = tuple(c[c[:, :, 1].argmin()][1])
        extBot = tuple(c[c[:, :, 1].argmax()][1])

        ADD_PIXELS = add_pixels_value
        new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS, extLeft[0]-ADD_PIXELS:extRight[0]+
            ADD_PIXELS].copy()
        set_new.append(new_img)

    return np.array(set_new)
```

MR görüntüleri üzerinden örnek;



Tablo 2. Adım Adım Pyimagesearch’ün Nasıl Çalıştığı

Veri ön işlemenin son adımı olarak görüntüleri (224,224) formatına sıkıştırdım.

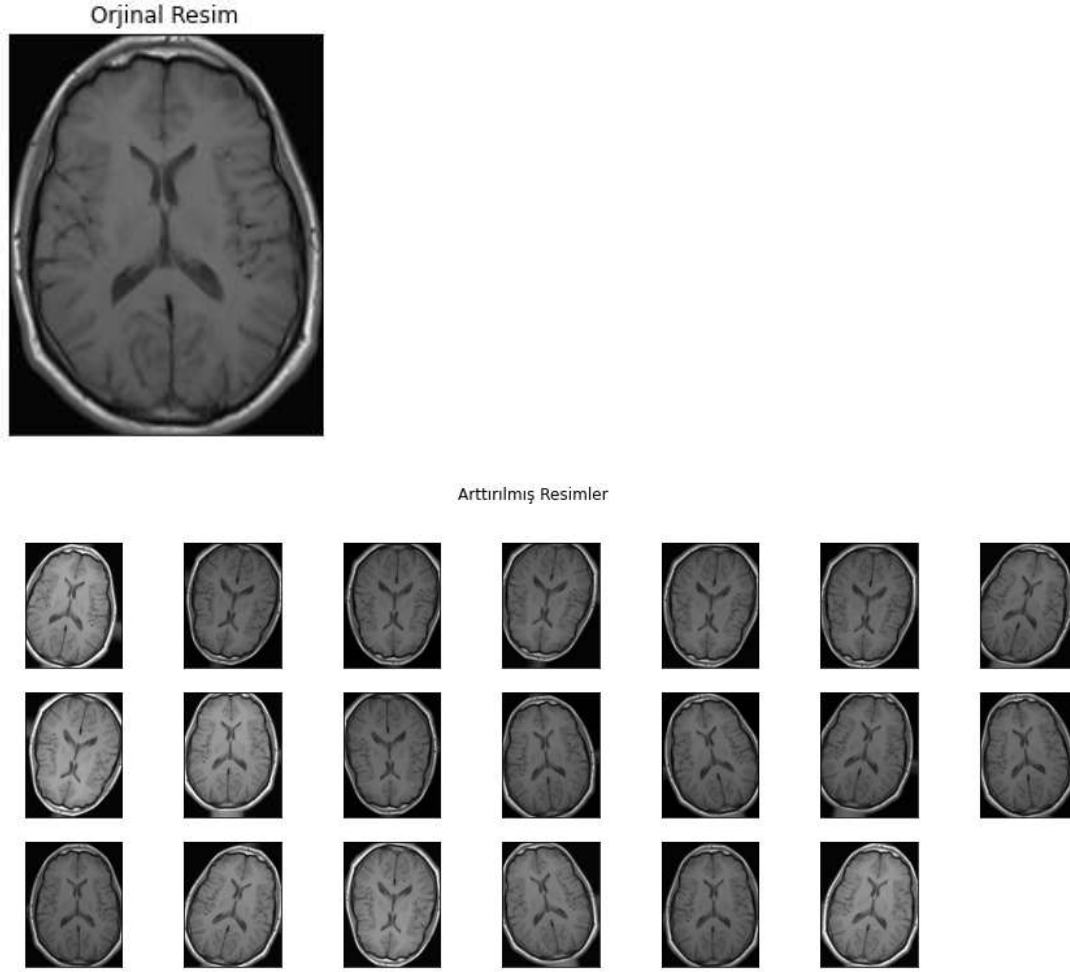
```
def preprocess_imgs(set_name, img_size):
    set_new = []
    for img in set_name:
        img = cv2.resize(
            img,
            dsize=img_size,
            interpolation=cv2.INTER_CUBIC
        )
        set_new.append(preprocess_input(img))
    return np.array(set_new)
```

Son olarak veri setinin boyutunu büyütme amacıyla ‘Data Augmentation’ yöntemini kullandım. Bu sayede veri sayım artmakla beraber görsellerin farklı açılarda ve farklı derinliklerde çeşitlendirmiş oldum. Daha sonraki makine öğrenmesinde overfitting ve underfitting’in de önüne geçilmeyi amaçlamış oldum.

Kod:

```
demo_datagen = ImageDataGenerator(
    rotation_range=15,
    width_shift_range=0.05,
    height_shift_range=0.05,
    rescale=1./255,
    shear_range=0.05,
    brightness_range=[0.1, 1.5],
    horizontal_flip=True,
    vertical_flip=True
)
```

Örnek çıktı ise şu şekildedir;

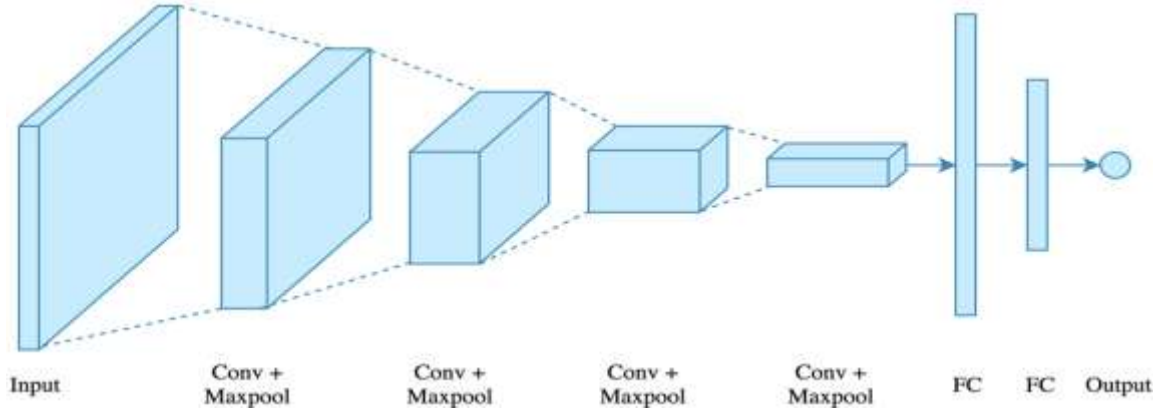


Tablo 3. Data Augmantation'un Çıktıları ve Örnek Görseller

Bununla beraber VGG-16 için ön işlem tamamlanmış oldu.

2.2- CNN Model Oluşturma

Görsel ilk olarak $224 \times 224 \times 3$ şeklinde algoritmaya giriş yapar. İlk konvansiyonel işlem den sonra görsel, $224 \times 224 \times 64$ boyutuna getirilir. Bu şekilde görsel derinleşmeye başlar. Sonrasında MaxPooling katmanı ile görsel daha yoğun bilgi içeren fakat daha küçük boyuta getirilir. Bu işlem birkaç tekrar daha devam eder. Görsel son olarak $7 \times 7 \times 512$ boyutuna geldiğinde ise boyut olarak çok küçük fakat çok derin bir görsel oluşturulmuş olur. Bu şekilde görsel FullyConnectedLayer'a bağlanmaya hazır hale gelir. Artık elimizde 4096 uzunluğunda giriş vektörü oluşmuş olur. 2 FullyConnectedLayer'dan sonra sınıflandırıcı katmanı bağlamak amacıyla CNN algoritmalarında sıkça kullanılan Softmax kullanılır. Çalışma için oluşturulan şema;



Şekil 6. Çalışmanın Aşamaları

Kaynak: Klestov, R., Klyuev, A., Stolbov V. (2018), “About some approaches to problem of metals and alloys microstructures classification based on neural network technologies” Advances in Engineering Research (AER), volume 157

2.3- VGG-16 Modelinin Kurulması

Kod:

```
# Ana model
vgg16_weight_path = '../input/keras-pretrained-models/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5'
base_model = VGG16(
    weights=vgg16_weight_path,
    include_top=False,
    input_shape=IMG_SIZE + (3,)
)
```

VGG-16'nın daha önceden buna benzer amaçlarla eğitilmiş modelini kullandım. Model ağırlıklarını ise 'vgg16_weight_path' metodu ile belirlemesini istedim.

Kod:

```
NUM_CLASSES = 1

model = Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(NUM_CLASSES, activation='sigmoid'))

model.layers[0].trainable = False

model.compile(
    loss='binary_crossentropy',
    optimizer=RMSprop(lr=1e-4),
    metrics=['accuracy']
)

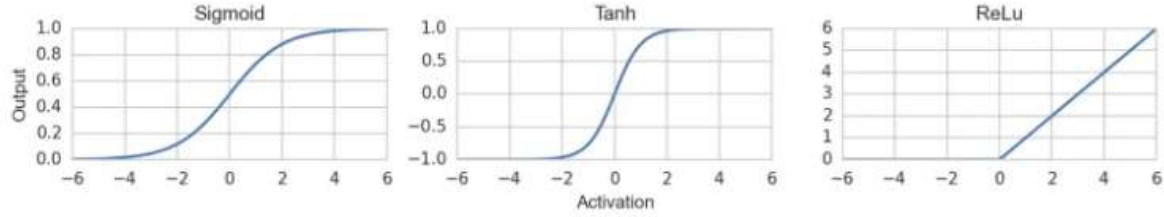
model.summary()
```

Model ile ilgili genel bilgiler ve istekler burada mevcut. Modelin ardışık olarak ilerlemesini; flatten, dropout ve dense katmanları ile gösterim istedim. Aynı zamanda activation için sigmoid bir fonksiyon kullanmasını istedim. Sigmoid fonksiyon gösterimi:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x} = 1 - \sigma(-x).$$

Küçük değerler için (<-5) sigmoid fonksiyonu sıfıra yakın bir değer döndürürken, büyük değerler için (>5) fonksiyonun sonucu 1'e yaklaşır.

Sigmoid, ikinci elemanın sıfır olduğu kabul edilen iki elemanlı softmax'a eşdeğerdir. Her zaman 0 ile 1 arasında değer üretir.



Grafik 3. Sigmoid, Tanh, ReLu Karşılaştırma.

Kaynak: Savaş, S. (2020, 30 Mayıs), “Derin Öğrenme”. <https://medium.com/yapay-zeka-makine-ogrenmesi-derin-ogrenme/derin-ogrenme-78e3f5d2765> adresinden alındı.

Flatten Katmanı: Çok boyutlu bir çıktı elde edildiğinde bunu Dense katmanına aktarmak için doğrusallaştırma işlemi yapılır. Herhangi bir ağırlıklandırma işlemi yapılmaz.

Dropout Katmanı: Overfitting de denilen modelin aşırı öğrenme (ezberleme) yapmasını önlemek için kullanılan katmandır. Sinir ağındaki görünür veya gizli birimleri ortaya çıkartır.

Dense Katmanı: Bir katmanın bir önceki katmana derinden bağlandığı katmandır.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 1)	25089
Total params: 14,739,777		
Trainable params: 25,089		
Non-trainable params: 14,714,688		

Tablo 4. Flatten, Dropout ve Dense Çıktıları

Eğitilemeyen parametreler (non-trainable params) maskeleye anlamına gelir. Tensorflow gibi yapılarda ağırlıkların herbirine hangi ağırlıkların düzeltilmesi gerektiğini belirtmek için eğitilemeyen maskeler ekler. Bu maskeler 0 ve 1'den oluşur.


```

EPOCHS = 120
es = EarlyStopping(
    monitor='val_acc',
    mode='max',
    patience=6
)

history = model.fit(
    train_generator,
    steps_per_epoch=25, #TrainData/batchsize
    epochs=EPOCHS,
    validation_data=validation_generator,
    validation_steps=10, #ValidationData/batchsize/2
    callbacks=[es],
    verbose=1
)

```

Epoch: Model eğitimi sırasında, verilerin tamamı aynı anda eğitime dahil edilmez. Veriler, belli sayıda parçalara bölünerek eğitimde kullanılır. İlk parça eğitilir, modelin performansı test edilir ve performansa göre geriye yayılım ("backpropagation") ile ağırlıklar güncellenir. Ardından yeni bir eğitim kümesi ile model yeniden eğitilir. Eğitimden sonra ağırlıklar yeniden güncellenir. Bu işlem her eğitim adımında tekrarlanarak model için en uygun ağırlık değerleri bulunmaya çalışılır. Her bir eğitim adımına 'epoch' denir.

İlk epoch:

```

Epoch 1/120
25/25 [=====] - 15s 477ms/step - loss: 4.6707 - accuracy: 0.6011 - val_loss: 2.3074 - val_accu
acy: 0.7375
Epoch 2/120
25/25 [=====] - 11s 447ms/step - loss: 2.9972 - accuracy: 0.6870 - val_loss: 1.6046 - val_accu
acy: 0.8313

```

Son epoch:

```

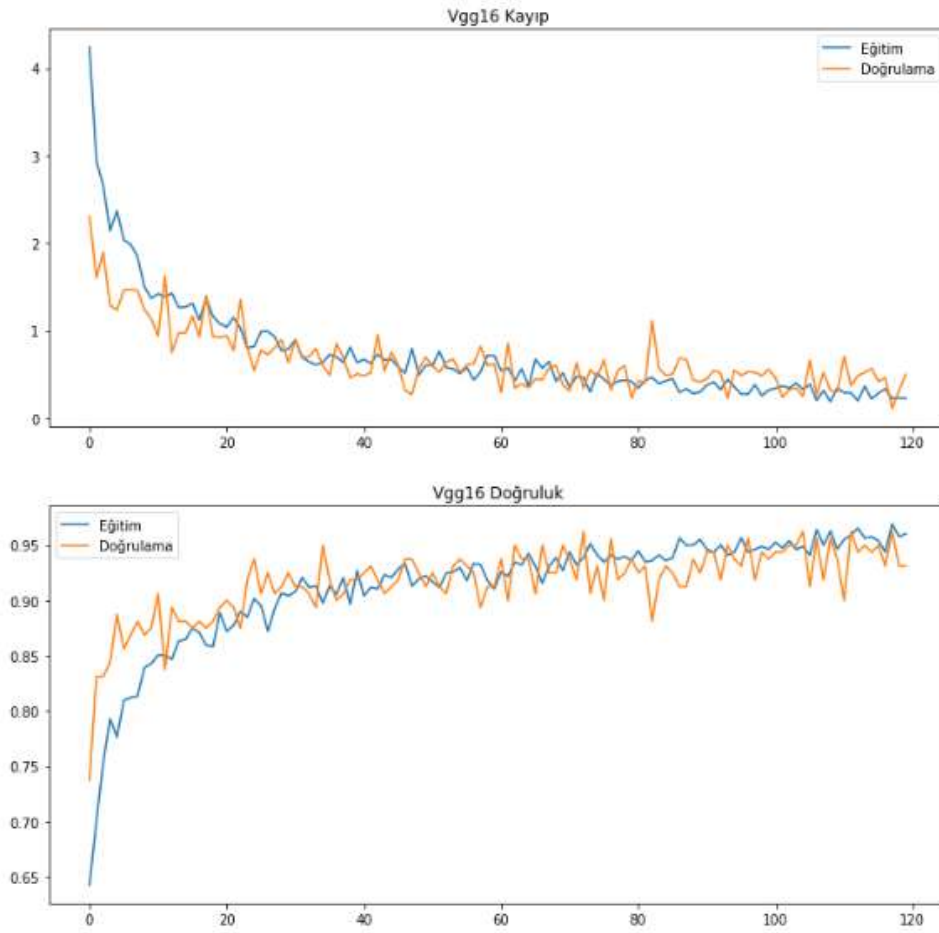
acy: 0.9438
Epoch 120/120
25/25 [=====] - 11s 446ms/step - loss: 0.4683 - accuracy: 0.9492 - val_loss: 0.6758 - val_accu
acy: 0.9438

```

Eğitim için aldığımız bütün epoch parçası eğitilip test edildikten sonra doğruluk 0.9438 olduğu görülmektedir. Oran iyi olmasına rağmen açıklayıcılığı yeterli değildir. Bunun için başka metrikler ile karşılaştıracacağız.

2.4- Modelin Metrikler İle Ölçülmesi

VGG-16'nın eğitim(train) ve doğrulama(validation) verileri üzerindeki kayıp(loss) ve doğruluk(accuracy) grafikleri:



Grafik 4. VGG-16 Kayıp ve Doğruluk Grafiği

Eğitim Verisi: 0.943, Test Verisi: 0.920

Görüldüğü gibi 120 epoch dosyası tek tek işleme alınmış olup her birine ait doğruluk ve kayıp değerleri grafikleştirilmiştir. Doğruluğun devamlı olarak artıp belli bir noktada platolaması istenen bir durumdur. Bazı noktalarda çok sert düşüşler ve çıkışlar yapsa da genel olarak başarılı bir sonuç çıkardığı söylenebilir.

Confution Matrix(Karışıklık Matrisi)

		GERÇEK	
		Pozitif	Negatif
TAHMİN	Pozitif	TP	FP
	Negatif	FN	TN

Tablo 5. Karmaşıklık Matrisi Açıklama

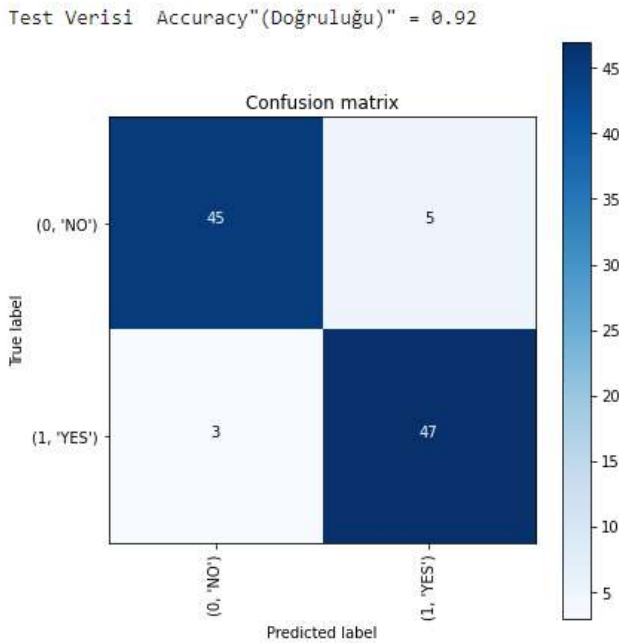
TP (True Positive): Pozitif olarak değerlendirilen ve gerçekten de pozitif olan doğru değerlerin sayısını ifade eder.

FP (False Positive): Pozitif olarak değerlendirilen fakat aslında negatif olan hatalı değerlerin sayısını ifade eder.

TN (True Negative): Negatif olarak değerlendirilen ve gerçekten de negatif olan doğru değerlerin sayısını ifade eder.

FN (False Negative): Negatif olarak değerlendirilen fakat aslında pozitif olan hatalı değerlerin sayısını ifade eder.

Tahminleme sonucunda oluşan confusion matrix:



Tablo 6. Test Verisi Doğruluk Oranı

Hangi sınıflandırma skorunun daha önemli olduğu, çalışmaya ve veri setinin özelliklerine göre değişmektedir. Accuracy (Doğruluk), özellikle hedef değişkendeki (target variable) sınıflar arasında bir denge var ise ve skorlar arası bir uyum söz konusu ise kullanılmaktadır. Dolayısıyla hedef değişkendeki sınıf dengesi değişken olduğunda yanıltıcı olabilmektedir. Bu sebeple precision, recall ve F1 skorların kullanılması önerilmektedir. Bu nedenle, bir modelin performansını değerlendirmek için, sınıf dağılımına bakmak ve özellikle sınıf dengesi değişken olduğunda precision, recall ve F1 skorun yanı sıra accuracy skoruna da bakmak önemli olmaktadır.

Recall ve Precision bulunduğu senaryoya göre önemi değişmektedir.

Bu skarlardan Duyarlılık (Recall) skoru, modelin pozitif sınıfı doğru bir şekilde tespit etme yeteneğini ölçmektedir. Bu sebeple Recall skoru yanlış negatiflerin kabul edilemeyeceği uygulamalarda önemli hale gelebilmektedir. Bunun gibi sağlık uygulamalarında yanlış negatif tahminler yani gerçekten hasta

olan kişilerin yanlışlıkla sağlıklı olarak sınıflandırılması sonucunda kaçırılan teşhisler, çok kritiktir. Bu nedenle, bu uygulamaya yönelik Recall skoru diğerlerine göre daha önemlidir.

Yanlış pozitiflerin kabul edilemeyeceği bir başka uygulamada hassasiyet (Precision) skoru önem arz etmektedir. Örneğin e-mail spam tespiti gibi uygulamalarda yanlış pozitif tahminler, yani spam olmayan e-postaların yanlışlıkla spam olarak sınıflandırılması sonucunda kaybedilen e-postalar, son derece önemli olabilir.

F1 Skoru, hem Precision hem de Recall skorlarının harmonik ortalaması sonucu hesaplanmaktadır. Dolayısıyla F1 skoru, yanlış pozitiflerin ve yanlış negatiflerin birbirine göre önemli olduğu durumlarda kullanılmaktadır.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Kod:

```
accuracy = accuracy_score(y_test, predictions)
print('Doğruluk: %f' % accuracy)
precision = precision_score(y_test, predictions)
print('Kesinlik: %f' % precision)
recall = recall_score(y_test, predictions)
print('Duyarlılık: %f' % recall)
f1 = f1_score(y_test, predictions)
print('F1 Skoru: %f' % f1)
```

Bu işlem sonucunda:

Doğruluk: 0.920000

Kesinlik: 0.903846

Duyarlılık: 0.940000

F1 Skoru: 0.921569

Sonuçlar incelendiğinde duyarlılık(recall) değerinin %94, F1 skorunun %92 olması bu alanda yapılan birçok çalışmadan daha iyi sonuç çıkardığını göstermektedir. Veri setinin diğerlerine göre daha kapsamlı ve detaylı olmasının yanı sıra yapılan veri manipülasyonları işlemlerinin ve seçilen makine öğrenmesi algoritmasının da sonuç iyileştirmede katkısı olduğunu göstermektedir.

Bunların dışında Cohen kappa ve roc auc skorlarının da incelenmesi gerekmektedir.

Cohen Kappa Skoru: İki gözlem arasındaki karşılaştırmalı uyuşmanın güvenilirliğini ölçer. Parametrik olmayan bir ölçekçeşididir. Landis ve Koch (1977) elde edilen çıktıları açıklamak için tablo oluşturdu. Oluşturulan yorum tablosu:

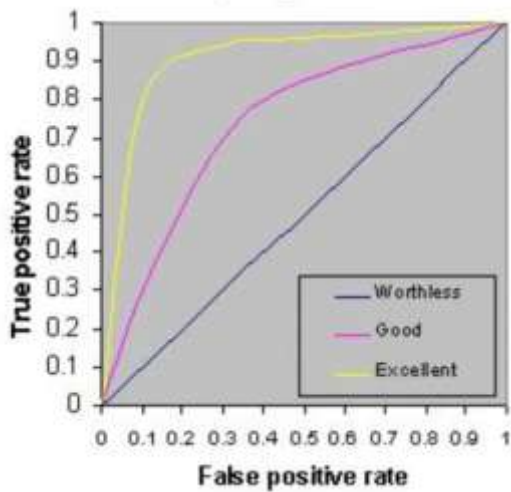
κ	Yorum
< 0	Hiç uyuşma olmaması
0.0 — 0.20	Önemsiz uyuşma olması
0.21 — 0.40	Orta derecede uyuşma olması
0.41 — 0.60	Ekseriyetle uyuşma olması
0.61 — 0.80	Önemli derecede uyuşma olması
0.81 — 1.00	Neredeyse mükemmel uyuşma olması

Tablo 7. Cohen Kappa Skoru Açıklama

Wikipedia (t.y.), “Cohen’in Kappa Katsayısı”. https://tr.wikipedia.org/wiki/Cohen%27in_kappa_katsayısı#:~:text=Cohen%27in%20kappa%20katsayısı%20iki,güvenirliğini%20ölçen%20bir%20istatistik%20yöntemidir adresinden alındı.

ROC AUC Skoru: ROC temelde bir olasılık eğrisidir. Eğrinin altında kalan alan, AUC ayrılabilirliğin derecesini ve/veya ölçüsünü gösterir. Sınıflandırma problemleri için önemli bir ölçüm aracıdır. Asıl sonuç pozitifken modelin pozitif sınıfı tahmin etmekte ne kadar başarılı olduğunu açıklamaya yarar.

Genel grafiği şu şekildedir:



Grafik 5. ROC AUC Eğrisi

Kaynak: Öğündür, G. (2020, 12 Ocak), “ROC ve AUC”. <https://medium.com/@gulcanogundur/roc-ve-auc-1fefcf71a14> adresinden alındı.

Kod:

```
kappa = cohen_kappa_score(y_test, predictions)
print('Cohens kappa: %f' % kappa)
# ROC AUC
auc = roc_auc_score(y_test, predictions)
print('ROC AUC: %f' % auc)
```

Bu işleme göre sonuçlar:

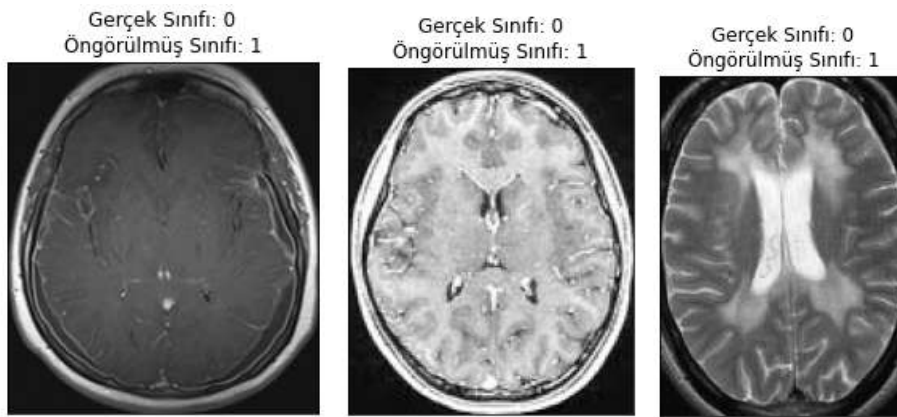
Cohens kappa: 0.840000

ROC AUC: 0.920000

Cohen kappa'nın %84 olması, iki gözlem arasındaki uyuşmanın mükemmele yakın olduğunu gösterir. ROC AUC'un %92 olması ise modelin sınıflar arasındaki ayırt etme performansının mükemmele yakın olduğunu gösterir. Bu testler sonucunda oluşturulan ve test edilen modelin çok iyi sonuç çıkardığı söylenebilir.

Oluşturulan modelin doğru tahmin ettiği gibi yanlış tahmin ettiği görseller de olduğu görülmektedir. Örneğin aşağıdaki görsellerde beyin tümörü bulunmamasına rağmen varmış gibi sınıflandırılmıştır.

0: Tümör yok, 1: Tümör var.



Tablo 8: Beyin Tümörü Bulunmamasına Rağmen Varmış Gibi Sınıflandırılan Görseller

Bu sonuçlar incelenerek modelin hangi tip görsellerde hata yaptığı tespit edilerek doğruluk artırılabilir.

SONUÇ

Bu çalışmada, MR görüntüleri kullanılarak otomatik beyin tümörü tespiti için başarılı bir yöntem önerdik. Yöntem, VGG-16 modelini kullanarak transfer öğrenme ye dayalı bir derin öğrenme yaklaşımı sunmaktadır. Veri sayısını arttırmak ve çeşitliliği sağlamak amacıyla Data Augmentation yöntemi kullanıldı. VGG-16 modeli sonucunda elde edilen doğruluk: %92, kesinlik: %90, duyarlılık: %94 ve F1 skoru: %92 oldu. Bu sonuçlar, benzer yöntemlerle karşılaştırıldığında daha üstün bulundu. Ayrıca, halihazırda eğitilmiş derin öğrenme modeli kullanıldığı için öğrenilebilir parametrelerin optimizasyon süresi kısaldı. Bu nedenle, önerilen yöntemin beyin tümörlerinin tespitinde uzmanlara karar vermede yardımcı olabileceği düşünülmektedir. Gelecekte yapılabilecek çalışmalarda, farklı beyin tümörlerini ve alt türlerini tespit etmek için daha büyük beyin tümörü veri setleri üzerinde farklı ve geliştirilmiş modeller kullanılarak daha kapsamlı ve etkili çalışmalar yapılabilir.

KAYNAKÇA

- Aslan, M. (2022, 12 Şubat), “Derin Öğrenme Tabanlı Otomatik Beyin Tümör Tespiti”, *Fırat Üniversitesi Müh. Bil. Dergisi*, 34/1: 399-407.
- Ateş, E. (2021), “Yapay Zekâ (Artificial Intelligence)”, *Jandarma ve Sahil Güvenlik Akademisi. Siber Ansiklopedi: Siber Ortama Çok Disiplinli Bir Yaklaşım*, 644-647.
- Beluga(2017),“Keras Pretrained Models”.<https://www.kaggle.com/datasets/gaborfodor/keras-pretrained-models> adresinden alındı.
- Bozkurt, B. (2021, 30 Eylül), “CNN (Convolutional Neural Networks) Nedir?”. <https://bartubozkurt35.medium.com/cnn-convolutional-neural-networks-nedir-a5bafc4a82a1> adresinden alındı.
- Ergin, T. (2018, 2 Kasım), “Convolutional Neural Network (ConvNet ya da CNN) Nedir, Nasıl Çalışır?”. <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad> adresinden alındı.
- Gülgün, O. D., Erol, H. (2020), “Medical Image Classification With Hybrid Convolutional Neural Network Models”, *Journal of Computer Science and Technologies*, 1/1: 28-41.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016), “Deep Residual Learning For Image Recognition.”. *IEEE Conference On Computer Vision And Pattern Recognition (CVPR)*, 770-778.
- Ilgın, U.(2021), Dataset: “braintumor”.<https://www.kaggle.com/datasets/uurilgin/mybraintumor/data> adresinden alındı.
- Ilgın, U.(2021), “VGG-16 Modelini Kullanarak Beyin Tümörü Algılama”. <https://www.kaggle.com/code/uurilgin/vgg-16-modelini-kullanarak-beyin-t-m-r-alg-lama> adresinden alındı.
- Keras (t.y.), “VGG16 and VGG19”. <https://keras.io/api/applications/vgg/> adresinden alındı.
- Kırlı, O., Sansarcı M., Özkaraca O., Çetin G. (2023), “Manyetik Rezonans Görüntülerinden Beyin Tümörü Tespitinde Sınıflandırma Algoritmalarının Karşılaştırmalı Analizi”, *Türk Mühendislik Araştırma ve Eğitimi Dergisi*, 2/2: 113-122.
- Klestov, R., Klyuev, A., Stolbov V. (2018), “About Some Approaches To Problem Of Metals And Alloys Microstructures Classification Based On Neural Network Technologies”, *Advances In Engineering Research (AER)*, 157.

- Kuruoğlu, E. (2020, 14 Mayıs), “Beyin Tümörü Nedir?”. <https://www.medicana.com.tr/saglik-rehberi-detay/10269/beyin-tumoru-nedir#:~:text=Beyin%20hücrelerinin%20yenilenmesi%20sırasında%20kopyalamada,kafatası%20içinde%20basınca%20neden%20olmaktadır> adresinden alındı.
- Neolife (t.y.), “Beyin Tümörü Belirtileri”. <https://www.neolife.com.tr/beyin-tumoru-belirtileri/> adresinden alındı.
- Nova Research Lab. (2020, 8 Ocak), “Öğrenme Aktarımı/ Transfer Learning”. <https://medium.com/novaresearchlab/öğrenme-aktarımı-transfer-learning-c0b8126965c4> adresinden alındı.
- Oracle (t.y.), “Makine öğrenimi nedir?”. <https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-machine-learning/> adresinden alındı.
- Öğündür, G. (2020, 12 Ocak), “ROC ve AUC”, <https://medium.com/@gulcanogundur/roc-ve-auc-1fefcfc71a14> adresinden alındı.
- Poyraz, S. (2023, 5 Mart), “Nedir Bu OpenCV?”. <https://zaryob.medium.com/nedir-bu-opencv-9091cca743c9> adresinden alındı.
- Savaş, S. (2020, 1 Haziran), “Derin Öğrenme Mimarileri”. <https://medium.com/yapay-zeka-makine-öğrenmesi-derin-öğrenme/derin-öğrenme-mimarileri-ce3c16a22ffc> adresinden alındı.
- Savaş, S. (2020, 30 Mayıs), “Derin Öğrenme”. <https://medium.com/yapay-zeka-makine-öğrenmesi-derin-öğrenme/derin-öğrenme-78e3f5d2765> adresinden alındı.
- Sedlar, S. (2018, Eylül), “Brain Tumor Segmentation Using a Multi-path CNN Based Method”. Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 403-422.
- Sharma, P., Sanyal, S., Maji, A., (2020, Ocak), “Deep Learning Techniques For Classification Of Brain MRI”, *International Journal Of Intelligent Systems Technologies And Applications*, 19/6: 571.
- Şener, A., Ergen, B. (2023), “Enhancing Brain Tumor Detection on MRI Images Using an Innovative VGG-19 Model-Based Approach”, *Sakarya University Journal of Science*, 27/5: 1128-1140.
- Şener, M. (2023, 14 Kasım), “Makine Öğrenmesinde Aktarımlı Öğrenme Yaklaşımı ve Aktarımlı Öğrenme Modelleri”. <https://blog.architech.com/blog/2468/makine-ogrenmesinde-aktarimli-ogrenme-yaklasimi-ve-aktarimli-ogrenme-modelleri/> adresinden alındı.
- Taş, M. O., Ergin S. (2020), “Beyin Tümörü Varlığının Geleneksel Derin Öğrenme Tekniği Kullanılarak Tespiti ve MR Görüntülerinden K-Means Segmentasyonu Kullanılarak Kesin Tümör Konumlarının Belirlenmesi”, *İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi*, 1/2: 91-97.

Wikipedia (t.y.), “Cohen'in kappa katsayısı”.

https://tr.wikipedia.org/wiki/Cohen%27in_kappa_katsayısı#:~:text=Cohen%27in%20kappa%20katsayısı%20iki,güvenirliğini%20ölçen%20bir%20istatistik%20yöntemidir adresinden alındı.