

T.C.
AYDIN ADNAN MENDERES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

BEYİN MANYETİK REZONANS GÖRÜNTÜLERİNDE
DERİN ÖĞRENME YÖNTEMİ İLE BEYİN TÜMÖRÜ
TESPİTİ

HAZIRLAYAN

Furkan EGE

SEMİNER DANIŞMANI

Prof. Dr. Muhsin ÖZDEMİR

Aydın - 2023

İÇİNDEKİLER

KISALTMALAR DİZİNİ	4
1. BÖLÜM	5
1. GİRİŞ	5
1.1. Beyin Manyetik Rezonans Görüntüleri	6
1.2. Derin Öğrenme Yöntemleri	7
1.3. Beyin Tümörü ve Tespiti	8
1.4. Amaç	9
2. BÖLÜM	10
2. BEYİN MANYETİK REZONANS GÖRÜNTÜLERİNİN İŞLENMESİ ...	10
2.1. Beyin MR Görüntüleri	10
2.2. Görüntü Ön işleme	11
2.3. Görüntü Dönüştürme	11
3. BÖLÜM	12
3. DERİN ÖĞRENME YÖNTEMLERİ	12
3.1. Yapay Sinir Ağları	13
3.2. Farklı Yapıdaki Derin Öğrenme Yöntemleri	13
3.3. Derin Öğrenme Yöntemleri ile Beyin Tümörü Tespiti	14
4. BÖLÜM	15
4. BEYİN TÜMÖRLERİ HAKKINDA GENEL BİLGİ	15
4.1. Beyin Tümörleri Türleri	16
4.2. Beyin Tümörlerinin Belirtileri	16
4.3. Beyin Tümörlerinin Teşhisi	18

5. BÖLÜM	19
5. DERİN ÖĞRENME YÖNTEMLERİ İLE BEYİN TÜMÖRÜ TESPİTİ ..	19
5.1. Derin Öğrenme Yöntemleri ile Beyin Tümörü Tespiti Süreci	19
5.2. Nitelikli Beyin MR Görüntülerinin Elde Edilmesi	20
5.3. Uygulama 1: %77'ye Varan Kesinlik	22
5.4. Uygulama 2: %99'a Varan Kesinlik	28
6. BÖLÜM	34
6. GÖRÜNTÜ SINIFLANDIRMADA EN ÇOK KULLANILAN TEKNOLOJİLER	34
6.1. Görüntü Sınıflandırmada En Çok Kullanılan Algoritmalar	34
6.2. Görüntü Sınıflandırmada En Çok Kullanılan Programlama Dilleri .	35
6.3. Görüntü Sınıflandırmada En Çok Kullanılan Kütüphaneler	36
6.4. Görüntü Sınıflandırmada En Çok Kullanılan Modeller	37
7. BÖLÜM	39
7. SONUÇ VE ÖNERİLER	39
7.1. Derin Öğrenme Yöntemlerinin Beyin Tümörü Tespitindeki Başarıları	39
7.2 Öneriler	40
KAYNAKLAR	41

KISALTMALAR DİZİNİ

MR : Manyetik Rezonans

AI : Yapay Zeka

BT : Bilgisayarlı Tomografi

CNN : Evrişimli Sinir Ağları

RNN : Tekrarlayan Sinir Ağları

AN : Autoencoder Networks

1. BÖLÜM

1. GİRİŞ

Bilim ve teknoloji, sağlık bilimleri alanında büyük ilerlemelere kapı açmıştır. Tıbbi görüntüleme teknolojileri özellikle son yıllarda hızlı bir gelişim göstermiş, sağlık profesyonellerinin hastalıkları daha iyi anlamasını ve teşhis etmesini sağlamıştır. Bu teknolojilerden biri Manyetik Rezonans (MR) Görüntüleme teknolojisi olmakta ve MR organları, dokuları ve kemikleri ayrıntılı bir şekilde göstererek beyin tümörleri dahil birçok farklı hastalığının teşhisinde yardımcı olmaktadır (Maxwell vd., 2017). Beyin tümörü gibi anormallikleri tespit etmek için elde edilen bu görüntülerin analiz edilmesi oldukça hassas ve deneyim gerektiren bir süreçtir. Son yıllarda tıbbi görüntü analizi alanında özellikle görüntü tanıma gibi karmaşık görevlerde başarılı olan derin öğrenme yöntemleri kullanılmaya başlanmıştır. Derin öğrenme, yapay zekanın (AI) bir alt dalıdır ve genellikle görüntü tanıma gibi karmaşık görevlerde kullanılır (Li vd., 2018). Derin öğrenme yöntemleri sayesinde daha doğru ve hızlı tümör tespiti yapılabilmektedir.

Bu seminerde beyin MR görüntülerini kullanarak beyin tümörlerini tespit etmek için derin öğrenme yöntemlerinin nasıl kullanılabileceğini detaylı bir şekilde ele alınacaktır. Öncelikle beyin MR görüntülemeyi, derin öğrenme yöntemlerini ve beyin tümörlerini tanımlayarak temel bilgiler sunulacaktır. Ardından bu teknolojilerin birleşimini ve beyin tümörü tespitinde nasıl kullanıldığı açıklanarak mevcut uygulamalar ve yöntemler üzerinde durulacaktır. Son olarak bu yöntemlerin başarılarını ve gelecekteki potansiyelleri değerlendirilecektir. Bu sayede tıbbi teşhis ve tedavi alanında bir devrim yaratılıp yaratılamayacağı ortaya konulacaktır (Kumar vd., 2016). Seminerin amacı, derin öğrenme yöntemlerinin beyin MR görüntüleri üzerindeki uygulamalarını ve bu yöntemlerin beyin tümörü tespitinde nasıl kullanıldığını açıklamaktır. Bu bilgilerin, tıbbi görüntüleme teknolojilerinin ve AI uygulamalarının daha geniş kapsamlı bir anlayışına katkı sağlaması umulmaktadır.

Derin öğrenme yöntemleri, sağlık bilimleri alanında birçok farklı uygulama bulmuştur ve bu seminer bu uygulamaların beyin tümörü tespitine nasıl uyarlanabileceği konusuna özellikle odaklanacaktır. Önümüzdeki bölümlerde beyin MR görüntülerinin işlenmesi, derin öğrenme yöntemlerinin farklı yapıları ve beyin tümörlerinin tespiti

konularında daha ayrıntılı bilgi verilecektir. Sonuç ve öneriler bölümünde ise derin öğrenme yöntemlerinin beyin tümörü tespitindeki başarılarına ve bu alandaki gelecekteki araştırma ve uygulama potansiyeline değinilecektir.

Bu seminerin tıbbi görüntüleme ve AI teknolojilerini birleştiren bir yaklaşımın, tıp alanındaki karmaşık sorunları çözme potansiyelini vurgulaması umulmaktadır. Bilgi ve teknolojinin bu entegrasyonu tıbbi teşhis ve tedavi süreçlerini dönüştürme ve hastaların yaşam kalitesini iyileştirme potansiyeline sahiptir (Maxwell vd., 2017; Li vd., 2018; Kumar vd., 2016).

1.1. Beyin Manyetik Rezonans Görüntüleri

MR Görüntüleme, dokuları ve yapıları detaylı bir şekilde görüntülemek için kullanılan ileri bir tıbbi görüntüleme tekniğidir. MR görüntüleme teknolojisi, radyo dalgaları ve güçlü mıknatıslar kullanarak vücuttaki su moleküllerinin manyetik özelliklerinden yararlanır ve bu, dokuların ve organların çok ayrıntılı görüntülerini oluşturur ve bu görüntüler dokulardaki anormalliklerin veya hastalıkların tespit edilmesine yardımcı olur (Maxwell vd., 2017). Beyin MR görüntüleme, beyin ve omurilikteki yapıları ve dokuları ayrıntılı bir şekilde görüntüler. Bu teknoloji beyindeki anormalliklerin ve hastalıkların tespitinde önemli bir araçtır. Beyin tümörleri, inmelere, demans ve diğer birçok beyin hastalığı MR görüntüleme kullanılarak teşhis edilebilir. Ayrıca bu teknik beyindeki hasarın veya hastalığın boyutunu ve konumunu belirlemede, tedavi seçeneklerinin belirlenmesinde ve tedavinin etkinliğinin izlenmesinde önemli bir rol oynamaktadır (Li vd., 2018).

MR görüntüleme diğer tıbbi görüntüleme tekniklerine göre çeşitli avantajlara sahiptir. Örneğin MR görüntülerinde, beyin ve omurilik dokusunun ayrıntılı ve yüksek çözünürlüklü görüntülerini elde etmek mümkündür. Bu beyindeki anormalliklerin ve hastalıkların daha doğru ve erken tespitine olanak sağlar. Ayrıca MR görüntüleme, radyasyona maruz kalma riski olmadan güvenli bir şekilde gerçekleştirilebilir ve bu da hastalar için daha az riskli bir seçenek sunduğu için genellikle diğer tıbbi görüntüleme tekniklerine göre tercih edilir (Kumar vd., 2016). Beyin MR görüntülerinin analiz ve değerlendirilmesi, deneyim ve uzmanlık gerektirir. Ancak son yıllarda derin öğrenme ve AI teknolojilerinin hızla gelişmesi, bu süreci

otomatikleştirmeye ve daha hızlı ve doğru tümör tespitine olanak sağlamıştır. Bu teknolojiler, beyin MR görüntülerindeki karmaşık özellikleri ve desenleri öğrenerek, anormal dokuları veya tümörleri otomatik bir şekilde tespit edebilir. Bu, tıbbi görüntüleme alanında bir devrim yaratmış ve teşhis süreçlerini büyük ölçüde hızlandırmıştır (Maxwell vd., 2017). Derin öğrenme teknolojileri, MR görüntülerindeki karmaşık özellikleri ve desenleri öğrenerek, beyindeki anormal dokuları veya tümörleri otomatik olarak tespit edebilir. Bu tür otomatik teşhis sistemleri, uzmanlar için değerli bir yardımcı olabilir ve tümör tespit sürecini hızlandırabilir. Ayrıca, bu sistemlerin doğruluğu ve hızı, beyin tümörlerinin erken tespitinde ve tedavi planlamasında büyük bir potansiyel sunar. Ancak, bu teknolojilerin başarılı bir şekilde uygulanabilmesi için, öncelikle beyin MR görüntülerinin doğru bir şekilde işlenmesi ve analiz edilmesi gerekmektedir (Li vd., 2018).

Sonuç olarak, beyin MR görüntüleme, beyin hastalıklarının teşhisinde hayati bir araçtır ve derin öğrenme teknolojileri, bu süreci daha hızlı ve doğru hale getirebilir. Bu seminerin ilerleyen bölümlerinde, beyin MR görüntülerinin işlenmesi ve analizinde kullanılan teknikleri ve derin öğrenme yöntemlerinin beyin tümörü tespitinde nasıl uygulanabileceğini ayrıntılı bir şekilde inceleyeceğiz (Kumar vd., 2016).

1.2. Derin Öğrenme Yöntemleri

Derin öğrenme, AI ve makine öğrenmesi alanlarının en önemli ve en hızla gelişen dallarından biridir. Derin öğrenme, insan beyninin bilgi işlemeyi ve öğrenmeyi gerçekleştirdiği şekilde modellenen yapay sinir ağları kullanır. Bu modeller, genellikle çok sayıda işleme katmanına sahip olup, bu sayede karmaşık özellikler ve desenler öğrenebilirler (Goodfellow vd., 2016). Derin öğrenme algoritmaları genellikle çok miktarda veri ile eğitilir ve bu veriler üzerinde öğrenme gerçekleştirirler. Bu özellikle tıbbi görüntüleme gibi alanlarda büyük veri setlerinden karmaşık özellikler ve desenler öğrenilmesini sağlar. Derin öğrenme tıbbi görüntülerde anormallikleri ve hastalıkları otomatik olarak tespit etme yeteneği ile tıbbi teşhis süreçlerini büyük ölçüde iyileştirebilir (LeCun vd., 2015).

Derin öğrenme algoritmaları çeşitli türlerde olabilir. Bunlar arasında evrişimli sinir ağları (CNN), tekrarlayan sinir ağları (RNN), ve otoenkoderler gibi derin öğrenme modelleri

bulunmaktadır. CNN, özellikle görüntü ve ses verilerinin işlenmesinde etkilidir ve tıbbi görüntüleme alanında sıklıkla kullanılır. Tekrarlayan sinir ağları ise genellikle zaman serisi verileri veya dil modellemesi gibi sıralı verilerin işlenmesinde kullanılır. Otoenkoderler ise genellikle veri boyutunu azaltma ve gürültüyü azaltma işlemlerinde kullanılır (Schmidhuber, 2015).

1.3. Beyin Tümörü ve Tespiti

Beyin tümörleri, genellikle hücrelerin kontrolsüz bir şekilde çoğalması sonucunda beyinde oluşan anormal büyümelerdir. Tümörler genellikle iyi huylu (benign) veya kötü huylu (malign) olarak sınıflandırılır.

İyi huylu tümörler, çoğunlukla yerinde büyürler ve genellikle çevreleyen dokulara sızma veya yayılma eğiliminde değildirler. Bu tür tümörler genellikle tamamen çıkarılabilir ve genellikle tekrarlamazlar (Louis vd., 2016). Ancak, beyindeki iyi huylu tümörler bile ciddi sorunlara yol açabilir, çünkü beynin sınırlı alanı içinde herhangi bir ekstra kitle baskı oluşturabilir.

Kötü huylu tümörler ise genellikle daha agresiftir ve çevreleyen dokulara sızabilir. Beyinde kötü huylu tümörler, genellikle hızla büyürler ve genellikle cerrahi ile tamamen çıkarılamazlar çünkü genellikle çevreleyen normal beyin dokusuna yayılırlar. Bu tümörler sıklıkla radyasyon tedavisi, kemoterapi veya diğer tedavilere ihtiyaç duyarlar ve genellikle tekrarlarlar (Louis vd., 2016).

Beyin tümörlerinin belirtileri genellikle baş ağrısı, nöbetler, bulantı, denge sorunları, kişilik değişiklikleri, konuşma veya görme problemleri gibi belirtilerdir. Ancak, bu belirtiler birçok başka sağlık sorununda da görülebileceği için, tespit için genellikle görüntüleme teknikleri gereklidir (Mayo Clinic, 2023a).

Tümörlerin tespiti genellikle bir dizi test ve prosedürle yapılır. MR, beyin tümörlerinin tespiti ve tanısında sıkça kullanılan bir yöntemdir çünkü tümörün yerini, büyüklüğünü ve şeklini ayrıntılı bir şekilde görüntüleyebilir. Diğer tespit yöntemleri arasında bilgisayarlı

tomografi (CT) taramaları ve biyopsi bulunmaktadır. Bu testler genellikle bir bireyin semptomları olduğunda veya bir doktorun bir tümörden şüphelendiği durumlarda uygulanır (Smith-Bindman vd., 2019). Ancak, geleneksel görüntüleme teknikleri genellikle tümörlerin manuel olarak tespit edilmesini gerektirir ve bu süreç zaman alıcı ve hata oranı yüksektir. Bu nedenle, son yıllarda, derin öğrenme teknolojilerinin beyin tümörlerinin otomatik tespiti için kullanılması büyük bir ilgi görmüştür.

1.4. Amaç

Bu seminerin amacı, beyin MR görüntülerinde derin öğrenme yöntemlerini kullanarak beyin tümörlerinin nasıl tespit edilebileceğini ayrıntılı bir şekilde incelemektir. Beyin tümörleri, genellikle belirgin semptomlar göstermeden önce belirli bir büyüklüğe ulaşır ve bu hastaların tedaviye geç başlamasına neden olabilir (Ostrom vd., 2018). Derin öğrenme yaklaşımlarının tıbbi görüntüleme alanındaki potansiyeli ve etkisi, daha erken ve daha doğru teşhislerin yanı sıra hastalar için daha iyi sonuçlara yol açabilir (Litjens ve vd., 2017). Bu, özellikle agresif türdeki beyin tümörleri için, erken tanı hayatta kalma oranlarını önemli ölçüde artırabilir. Bu sebeple, derin öğrenme yöntemlerinin beyin tümörü tespitinde nasıl kullanılabileceğini anlamak, bu yaşamsal konuda ilerleme kaydetmek için önemlidir.

Derin öğrenme, büyük veri setlerinden karmaşık özellikler ve desenler öğrenme yeteneğine sahip olup, tıbbi görüntülerde anormallikleri ve hastalıkları otomatik olarak tespit etme yeteneği ile tıbbi teşhis süreçlerini büyük ölçüde iyileştirebilir (LeCun vd., 2015). Bu seminerde, bu potansiyeli nasıl gerçekleştirebileceğimizi ve derin öğrenme teknolojilerinin beyin tümörü tespitinde nasıl etkin bir şekilde kullanılabileceğini inceleyeceğiz. Bu seminerin sonunda, dinleyicilerin, beyin MR görüntülerinde derin öğrenme yöntemlerini kullanarak beyin tümörlerinin tespiti konusunda detaylı bir anlayışa sahip olmaları beklenmektedir. Ayrıca, bu seminer, derin öğrenme teknolojilerinin tıbbi teşhis süreçlerinde nasıl daha etkin bir şekilde entegre edilebileceği konusunda da fikirler sunmayı hedeflemektedir (Goodfellow vd., 2016).

2. BÖLÜM

2. BEYİN MANYETİK REZONANS GÖRÜNTÜLERİNİN İŞLENMESİ

Beyin MR görüntülerinin işlenmesi, görüntü önileme, görüntü dönüştürme ve tümör tespiti gibi bir dizi adımı içerir. Bu işlemler, MR görüntülerinden maksimum bilgiyi elde etmek ve daha sonra bu bilgiyi derin öğrenme algoritmaları kullanarak beyin tümörlerini tespit etmek için kullanılır (Zhou vd., 2018).

2.1. Beyin MR Görüntüleri

Beyin MR görüntüleri, beyin ve merkezi sinir sistemi dokularının ayrıntılı görüntülerini sağlar. MR görüntüleme, bir manyetik alan ve radyo dalgaları kullanarak vücuttaki su moleküllerinin enerji seviyelerini manipüle eder. Bu enerji seviyelerinin değişimleri, ayrıntılı görüntüler oluşturmak için bir bilgisayar tarafından işlenir. Bu teknik, beyin dokusunun çeşitli alanları arasındaki hafif yoğunluk farklarını bile tespit edebilir, bu da beyin tümörlerinin tespitinde oldukça yararlıdır (National Institute of Biomedical Imaging and Bioengineering, 2023). Beyin MR görüntüleri genellikle bir dizi dilim şeklinde elde edilir ve her dilim, beyin dokusunun farklı bir kesitini temsil eder. Bu görüntüler, beyin tümörlerinin boyutunu, şeklini ve konumunu belirlemeye yardımcı olabilir. Ayrıca, bazı MR teknikleri, tümörün çevresindeki kan akışı ve metabolik aktivite gibi daha karmaşık özellikleri de tespit edebilir. Bu bilgiler, tümörün ne kadar hızlı büyüdüğünü ve tedaviye ne kadar iyi yanıt verdiğini belirlemeye yardımcı olabilir (Mayo Clinic, 2023b).

Ancak, MR görüntülerinin yorumlanması genellikle karmaşıktır ve uzman bir radyologun müdahalesini gerektirir. Ayrıca, bu görüntüler genellikle yüksek miktarda veri içerir ve bu verinin etkin bir şekilde işlenmesi, geleneksel görüntü işleme teknikleri kullanılarak zor olabilir. Bu bağlamda, derin öğrenme yöntemleri, beyin MR görüntülerinin analizini otomatikleştirebilir ve tümör tespit sürecini hızlandırabilir (Zhou vd., 2018).

2.2. Görüntü Önişleme

Beyin MR görüntülerinin işlenmesindeki ilk adım genellikle görüntü önişlemedir. Görüntü önişleme, bir görüntünün kalitesini artırmak veya belirli özellikleri vurgulamak için uygulanan bir dizi teknikten oluşur (Gonzalez ve Woods, 2018). Beyin MR görüntülerinde, görüntü önişleme genellikle görüntü gürültüsünün azaltılması, görüntü kontrastının iyileştirilmesi ve görüntü çözünürlüğünün artırılması gibi adımları içerir (Zhou vd., 2018). Gürültü azaltma, MR görüntülerinin kalitesini etkileyen rastgele varyasyonları azaltmayı hedefler. Bu, genellikle düşük geçiş filtreleri veya medyan filtreleri gibi teknikler kullanılarak yapılır (Gonzalez ve Woods, 2018). Görüntü kontrastının iyileştirilmesi, beyin dokusu ve tümör arasındaki yoğunluk farkını artırmayı hedefler ve genellikle histogram eşitleme veya kontrast sınırlama gibi teknikler kullanılarak gerçekleştirilir (Zhou vd., 2018). Görüntü çözünürlüğünün artırılması, görüntüdeki detayları geliştirmek için uygulanır ve genellikle interpolasyon teknikleri kullanılır (Gonzalez ve Woods, 2018).

Bu adımlar, görüntünün daha sonra derin öğrenme algoritmaları tarafından daha etkin bir şekilde işlenebilmesi için önemlidir. Görüntü önişleme ayrıca, MR görüntülerinin kalitesini iyileştirerek ve tümör tespiti için gerekli olan özellikleri vurgulayarak, tümör tespitinin doğruluğunu artırabilir (Zhou vd., 2018).

2.3. Görüntü Dönüştürme

Görüntü dönüştürme, görüntü önişlemeden sonra gelir ve genellikle derin öğrenme algoritmalarının etkinliğini artırmak için kullanılır. Görüntü dönüştürme, bir görüntüyü farklı bir görüntüye dönüştürürken bazı özelliklerin korunmasını veya vurgulanmasını sağlar. Bu süreç, bir görüntünün özelliklerini çıkarır ve bu özellikler daha sonra bir tümörün varlığını ve konumunu belirlemek için kullanılır (Zhou vd., 2018). Beyin tümörü tespitinde en yaygın kullanılan görüntü dönüştürme tekniklerinden biri, görüntüdeki özelliklerin çıkarılması ve çıkarılan özelliklerin daha sonra bir sınıflandırma modeli tarafından analiz edilmesini sağlayan CNN'lerdir (LeCun vd., 2015). Bu ağlar, görüntünün farklı özelliklerini belirlemek için çeşitli filtreler kullanır ve bu özellikler daha sonra bir tümörün varlığını ve konumunu belirlemek için kullanılır.

Bir diğ er yaygın g r nt  d n şt rme tekniğı, g r nt y  ayrıık dalgacık d n ş m  kullanarak frekans alanına d n şt rmektir. Bu teknik, t m r n varlığını ve konumunu belirlemek i in kullanılan  zelliklerin  ıkarılmasını sağılar (Zhou vd., 2018). G r nt  d n şt rme, derin  ğrenme algoritmalarının t m r tespitindeki bařarısında kritik bir rol oynar. Bu s re , t m r n belirlenmesi ve sınıflandırılması i in gerekli olan  zelliklerin  ıkarılmasını sağılar ve bu nedenle t m r tespitinin doğıruluğunu ve etkinliğini artırır (LeCun vd., 2015).

3. B L M

3. DERİN  ĞRENME Y NTEMLERİ

Derin  ğrenme, bir ok farklı veri tipi  zerinde karmařık modeller oluřturmak i in kullanılan bir yapay sinir ağı t r d r.  zellikle, b y k ve karmařık veri k melerinde, derin  ğrenme modelleri genellikle insan uzmanlara eřdeğer veya daha iyi performans g sterebilir (Goodfellow vd., 2016). Derin  ğrenme, son yıllarda bir ok farklı uygulama alanında,  zellikle de tıbbi g r nt  analizinde  nemli bir ara  haline gelmiřtir. Bu alanda, derin  ğrenme algoritmaları,  zellikle MR g r nt lerinde beyin t m rlerinin tespitinde olduk a etkilidir (Litjens vd., 2017).

Derin  ğrenme, verilerden  zellik  ıkarımı ve sınıflandırma gibi g revlerde geleneksel makine  ğrenmesi y ntemlerine g re bir dizi avantaja sahiptir.  zellikle, derin  ğrenme, manuel  zellik  ıkarımına gerek duymadan verilerden karmařık  zellikler  ğrenebilir. Bu,  zellikle tıbbi g r nt leme gibi alanlarda,  zellik  ıkarmanın zorlu ve zaman alıcı olabileceğı durumlarda b y k bir avantaj sağılar (LeCun vd., 2015). Ayrıca, derin  ğrenme algoritmaları, genellikle b y k miktarda eğıtim verisi gerektiren karmařık g revlerde y ksek performans g sterebilir. Bu nedenle, b y k ve karmařık tıbbi g r nt  veri k melerinin analizinde,  zellikle de beyin t m r  tespitinde etkilidirler (Litjens vd., 2017).

3.1. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninin bilgi işleme yöntemini taklit etmeye çalışan bir tür makine öğrenmesi modelidir. YSA'lar, girdi ve çıktı arasında karmaşık ilişkileri modellemek ve öğrenmek için bir dizi bağlantılı düğüm veya nöron kullanır (Schmidhuber, 2015). Yapay sinir ağları, özellikle görüntü ve ses tanıma, doğal dil işleme ve tıbbi görüntü analizi gibi karmaşık görevlerde etkilidir. YSA'lar, girdi verilerinden çıktı verilerine olan karmaşık ilişkileri modellemek için "katmanlı" bir yapı kullanır. Bu katmanlar genellikle bir girdi katmanı, bir veya daha fazla gizli katman ve bir çıktı katmanı olmak üzere üç tipe ayrılır. Girdi katmanı, modelin girdi verilerini alır; gizli katmanlar, bu girdi verilerini işler ve çıktı katmanı, modelin sonucunu verir. Her katman, bir dizi bağlantılı düğüm veya nörondan oluşur ve bu nöronlar, belirli bir aktivasyon fonksiyonu kullanarak verileri işler (Schmidhuber, 2015).

Derin öğrenme, yapay sinir ağlarının bir alt kümesi olarak kabul edilir ve genellikle birden çok gizli katman içerir. Bu ek katmanlar, modelin karmaşık veri kümelerinden daha sofistike özellikler çıkarabilmesini sağlar. Bu özellik, derin öğrenme yöntemlerinin tıbbi görüntü analizi gibi karmaşık görevlerde özellikle değerli olmasını sağlar. Özellikle, derin öğrenme modelleri, MR görüntülerinde beyin tümörlerinin tespiti gibi karmaşık ve zorlu görevlerde etkileyici sonuçlar göstermiştir (Litjens vd., 2017).

3.2. Farklı Yapıdaki Derin Öğrenme Yöntemleri

Derin öğrenme, Yapay Sinir Ağları (YSA) temelli bir makine öğrenmesi alt dalıdır ve bir dizi farklı model ve yöntem içerir. Bu yöntemlerin her biri, belirli bir tür veriyi veya belirli bir tür görevi işlemek için optimize edilmiştir. Bu derin öğrenme modelleri arasında CNN, RNN ve Autoencoder Networks (AN) gibi farklı yapıda modeller bulunmaktadır.

Evrişimsel Sinir Ağları, genellikle görüntü sınıflandırma ve segmentasyon gibi görevlerde kullanılan bir derin öğrenme modelidir. CNN'ler, girdi verilerinin lokal özelliklerini öğrenmek ve korumak için konvolüsyonel katmanlar kullanır. Bu katmanlar, görüntü verilerinin geniş alanlarından özellikler çıkarır ve bu özellikler daha sonraki katmanlarda birleştirilir (LeCun vd., 2015). CNN'ler, tıbbi görüntü analizinde, özellikle MR görüntülerinde beyin tümörlerinin tespiti gibi görevlerde sıklıkla kullanılır.

RNN'ler, genellikle zaman serisi verileri veya doğal dil işleme gibi sıralı verileri işlemek için kullanılan bir derin öğrenme modelidir. RNN'ler, önceki girdilerden gelen bilgileri hatırlama yeteneğine sahiptir, bu da onları zamana bağlı veri kümeleriyle çalışmak için ideal hale getirir (Hochreiter ve Schmidhuber, 1997).

AN ise, genellikle büyük veri kümelerindeki karmaşık yapıları veya özellikleri öğrenmek için kullanılan bir derin öğrenme modelidir. AN'ler, genellikle denetimsiz öğrenme görevlerinde kullanılır ve çoğunlukla veri kümelerinden yüksek seviye özellikler çıkarmak için kullanılır (Hinton ve Salakhutdinov, 2006).

3.3. Derin Öğrenme Yöntemleri ile Beyin Tümörü Tespiti

Derin öğrenme yöntemleri, tıbbi görüntüleme ve özellikle beyin tümörü tespiti alanında etkileyici sonuçlar elde etmiştir. Beyin tümörleri, çeşitli MR görüntüleme teknikleri kullanılarak görüntülenebilir, ancak bu görüntülerin doğru bir şekilde analiz edilmesi ve tümörlerin belirlenmesi karmaşık ve zorlu bir görev olabilir. Derin öğrenme, bu zorluğu aşma potansiyeline sahip bir teknolojidir. Evrişimsel Sinir Ağları (CNN), MR görüntülerinin analizinde sıklıkla kullanılır. Bir CNN, MR görüntülerini doğrudan girdi olarak alabilir ve bu görüntüler üzerinde eğitilebilir. Bu, ağın tümörleri ve diğer önemli özellikleri otomatik olarak belirlemesini sağlar. CNN'ler, görüntü verilerinin geniş alanlarından özellikler çıkarır ve bu özellikler daha sonraki katmanlarda birleştirilir. Bu, ağın karmaşık özellikleri öğrenmesini ve tümörleri diğer doku türlerinden ayırt etmesini sağlar (Rajpurkar vd., 2017).

Son yıllarda, derin öğrenme tabanlı yöntemlerin beyin tümörü tespitinde etkileyici sonuçlar verdiği gösterilmiştir. Örneğin, bir çalışmada, derin öğrenme modeli, beyin tümörlerini tespit etmek ve sınıflandırmak için kullanılmıştır. Bu model, yüksek hassasiyet ve özgüllük değerleri elde etmiştir ve bu sonuçlar, modelin beyin tümörlerini etkin bir şekilde tespit edebileceğini göstermiştir (Wang vd., 2017).

4. BÖLÜM

4. BEYİN TÜMÖRLERİ HAKKINDA GENEL BİLGİ

Beyin tümörleri beyin veya beyin zarlarında oluşan anormal hücrelerin birikimiyle ortaya çıkan kanserli veya kanserli olmayan kitlelerdir (Türk Neroşirürji Derneği, 2023). Bu tümörler beyin fonksiyonlarını etkileyebilir ve ciddi sağlık sorunlarına yol açabilir. Beyin tümörleri genellikle başka organlardaki kanserin yayılması sonucunda oluşan sekonder tümörler (metastatik tümörler) veya doğrudan beyinde başlayan primer tümörler olabilir (Türk Neroşirürji Derneği, 2023).

Beyin tümörleri farklı tiplerde olabilir ve benign (iyi huylu) veya malign (kötü huylu) olarak sınıflandırılabilir. Benign tümörler çevre dokulara yayılmaz ve genellikle cerrahi müdahaleyle tedavi edilebilir (Abeloğlu vd., 2020). Bununla birlikte malign tümörler hızlı bir şekilde büyüyebilir, çevre dokulara sızayabilir ve invazif olabilir. Malign tümörler beyin dokusuna zarar verebilir ve tedavi edilmediğinde ciddi komplikasyonlara neden olabilir (Türk Neroşirürji Derneği, 2023).

Beyin tümörlerinin belirtileri tümörün boyutuna, konumuna ve büyümesine bağlı olarak değişebilir. Sıklıkla görülen belirtiler arasında baş ağrısı, epileptik nöbetler, görme sorunları, dengesizlik, kas güçsüzlüğü, hafıza problemleri ve davranış değişiklikleri yer alır (Türk Neroşirürji Derneği, 2023). Ancak belirtiler tümörün türüne ve yerine göre değişebilir ve bazen tümör erken aşamalarda belirti vermez.

Beyin tümörlerinin teşhisi tıbbi görüntüleme yöntemleri kullanılarak yapılır. Bunlar arasında MR ve bilgisayarlı tomografi (BT) gibi yöntemler bulunur (Türk Neroşirürji Derneği, 2023). Tanı aşamasında tümörün boyutu, konumu ve malignite derecesi belirlenir. Bu bilgiler uygun tedavi yaklaşımının belirlenmesine yardımcı olur.

4.1. Beyin Tümörü Türleri

Beyin tümörleri, Beyindeki hücrelerin anormal bir şekilde büyümesi sonucunda oluşurlar. Bu tümörler kaynaklandıkları hücre tipine, büyüme hızına, yayılma potansiyeline ve histopatolojik özelliklerine göre farklı türlerde sınıflandırılır (Johnson, 2018).

Beyin tümörlerinin benign ve malign olarak genel anlamda iki ana kategoriye ayrılrsa da kaynaklandıkları hücre tipine göre de farklı alt türlere ayrılır. Örneğin, glioma tümörleri, beyin dokusundaki glial hücrelerden kaynaklanır ve en sık görülen beyin tümörlerindendir (Jones vd., 2017). Meningioma tümörleri ise beyin zarlarından (meninksler) kaynaklanır ve genellikle iyi huylu bir karaktere sahiptirler (Miller vd., 2019). Diğer bir tür olan pituitar adenomlar, hipofiz bezinden kaynaklanan tümörlerdir ve hormonal bozukluklara neden olabilirler (Brown vd., 2018).

Beyin tümörlerinin histopatolojik özellikleri de sınıflandırmada önemli bir rol oynar. Örneğin, astrositomalar, oligodendrogliomalar ve ependimomalar gibi tümörler, hücrelerin morfolojik özelliklerine ve doku yapısına dayalı olarak sınıflandırılır (Louis vd., 2016). Bu histopatolojik özellikler, tümörlerin davranışını ve tedaviye yanıtını etkileyebilir.

Beyin tümörlerinin farklı tiplerinin tanımlanması, doğru teşhis ve tedavi planlaması için önemlidir. Bu nedenle, tümörlerin türlerine göre sınıflandırılması ve özelliklerinin anlaşılması büyük önem taşır. Tümör sınıflandırması, klinik uygulamalarda doğru tanı koyma ve tedavi seçeneklerinin belirlenmesi açısından kritik bir adımdır (Reifenberger vd., 2021).

4.2. Beyin Tümörlerinin Belirtileri

Beyin tümörleri, çeşitli belirtilerle kendini gösterebilen kompleks bir sağlık sorunudur. Beyin tümörlerinin belirtileri, tümörün büyüklüğü, yeri, türü ve etkilediği beyin bölgelerine bağlı olarak çeşitlilik gösterebilir. Bu belirtiler, tümörün etkisiyle ortaya çıkan beyin fonksiyonlarındaki bozulmalardan kaynaklanır.

Beyin tümörlerinin belirtileri genellikle şunları içerebilir:

1. Baş ağrısı: Beyin tümörleri sıklıkla baş ağrısıyla ilişkilendirilir. Baş ağrısı genellikle sık ve şiddetli olabilir. Ağrı, genellikle sabahları daha belirgin hale gelir ve aktivitelerle artabilir.
2. Bulantı ve kusma: Beyin tümörleri, beyin omurilik sıvısının normal dolaşımını etkileyebilir. Bu durum, beyin içinde artan basınçla sonuçlanabilir ve bulantı ile kusmaya yol açabilir.
3. Nörolojik belirtiler: Beyin tümörleri, sinir hücrelerinin normal işlevini bozabilir ve çeşitli nörolojik belirtilere neden olabilir. Bu belirtiler arasında denge kaybı, koordinasyon zorlukları, güçsüzlük, uyuşma veya karıncalanma hissi, konuşma bozuklukları ve görme problemleri yer alabilir.
4. Davranışsal ve bilişsel değişiklikler: Beyin tümörleri, kişilik değişiklikleri, duygusal dengesizlikler, hafıza sorunları ve konsantrasyon güçlüğü gibi davranışsal ve bilişsel değişikliklere neden olabilir.
5. İştah ve kilo değişiklikleri: Beyin tümörleri, iştahın azalması veya artmasıyla ilişkilendirilebilir. Bu durum, kilo kaybı veya kilo artışına yol açabilir.
6. Epileptik nöbetler: Bazı beyin tümörleri, epileptik nöbetlere neden olabilir. Bu nöbetler, aniden ortaya çıkan kas seğirmeleri, bilinç kaybı, hareket kaybı veya garip hislerle karakterizedir.

Beyin tümörlerinin belirtileri, bireyden bireye farklılık gösterebilir ve tümörün büyüklüğü, konumu ve etkilediği beyin bölgelerine bağlı olarak değişiklik gösterebilir. Bu nedenle, herhangi bir belirtiye sahipseniz veya endişeleriniz varsa, bir sağlık uzmanına danışmanız önemlidir.

4.3. Beyin Tümörlerinin Teşhisi

Beyin tümörlerinin doğru bir şekilde teşhis edilmesi, uygun tedavi seçeneklerinin belirlenmesi ve hastalığın yönetimi açısından önemlidir. Beyin tümörlerinin teşhisi genellikle bir dizi görüntüleme testi, nörolojik muayene ve laboratuvar testlerinden oluşur. Bu yöntemler tümörün varlığını doğrulamak, türünü belirlemek, tümörün büyüklüğünü ve yayılımını değerlendirmek ve diğer tedavi seçenekleri için temel bilgileri sağlamak amacıyla kullanılır (Stupp vd., 2005).

Beyin tümörlerinin teşhisinde yaygın olarak kullanılan yöntemler şunlardır:

1. **Beyin Görüntüleme:** Beyin tümörlerinin teşhisinde en sık kullanılan yöntemlerden biri görüntüleme testleridir. MR ve bilgisayarlı tomografi (BT) gibi yöntemler beyin dokusunu detaylı bir şekilde görüntülemek ve olası tümörleri tespit etmek için kullanılır (Weller vd., 2014)
2. **Biyopsi:** Görüntüleme testlerinin sonuçlarına dayanarak, tanıyı kesinleştirmek ve tümörün türünü belirlemek için bir biyopsi yapılabilir. Biyopsi, beyin dokusundan bir örnek alınması ve laboratuvar ortamında incelenmesi işlemidir.
3. **Nörolojik Muayene:** Beyin tümörlerinin teşhisinde nörolojik muayene önemli bir adımdır. Nörolojik muayene beyin fonksiyonlarını, refleksleri, dengeyi ve motor becerileri değerlendirmek için kullanılır. Bu muayene tümörün etkilediği beyin bölgelerine bağlı olarak ortaya çıkan belirtileri saptamak için yapılır.
4. **Kan Testleri:** Beyin tümörlerinin teşhisinde kan testleri destekleyici bir rol oynayabilir. Kan testleri tümör belirteçleri adı verilen özel proteinlerin veya moleküllerin düzeylerini değerlendirmek için kullanılır. Bu belirteçler bazı tümörlerin varlığını veya tedaviye yanıtı gösteren belirtileri ortaya çıkarabilir.

Beyin tümörlerinin teşhisi, bir veya daha fazla yöntemin kombinasyonu ile gerçekleştirilir. Bu yöntemler, uzman doktorlar tarafından değerlendirilir ve doğru bir teşhisin konulmasını sağlar. Doğru ve güncel teknolojik yöntemlerin kullanılması, hastaların tedavi planlarının belirlenmesi ve tedavi sürecinin yönetilmesi açısından büyük önem taşır.

5. BÖLÜM

5. DERİN ÖĞRENME YÖNTEMLERİ İLE BEYİN TÜMÖRÜ TESPİTİ

Derin öğrenme yöntemleri, beyin tümörlerinin tespitinde yüksek doğruluk oranlarıyla kullanılan etkili araçlardır. Bu başlık altında iki farklı derin öğrenme modeli olan CNN'ler ile yapılan beyin tümörü tespiti uygulamalarının tanıtımı yer alacaktır. İlk olarak, %77'lere varan kesinlik oranıyla sonuç veren bir CNN uygulaması, ardından %99'lara varan kesinlik oranıyla sonuç veren bir diğer CNN uygulaması tanıtılacaktır. Bu uygulamalar beyin tümörü tespitinde derin öğrenme yöntemlerinin potansiyelini göstermektedir.

Derin öğrenme yöntemleriyle yapılan beyin tümörü tespiti çalışmaları, bu alanın gelişimi için önemli bir adımdır. Derin öğrenme modellerinin doğruluk oranlarının artması tümör teşhisinde daha kesin ve güvenilir sonuçlar sağlayabilmektedir ve bu nedenle derin öğrenme yöntemlerinin görüntü sınıflandırma alanında daha da ilerlemesi ve yaygınlaşması beklenmektedir.

5.1. Derin Öğrenme Yöntemleri ile Beyin Tümörü Tespiti Süreci

Derin öğrenme yöntemleri beyin tümörü tespiti sürecinde önemli bir rol oynamaktadır. Bu başlık altında derin öğrenme yöntemleri kullanılarak gerçekleştirilen görüntü sınıflandırma süreci ayrıntılı bir şekilde incelenecektir.

Derin öğrenme tabanlı görüntü sınıflandırma süreci genellikle aşağıdaki adımlardan oluşmaktadır:

1. **Veri Toplama ve Hazırlama:** İlk adımımız; görüntü sınıflandırma uygulaması için, yani bu seminerin konusu itibarıyla Beyin MR görüntüleri üzerinde Tümör tespiti yapabilmek için uygun veri setinin toplanmasıdır. Bu veri seti beyin MR görüntülerinden oluşur. Toplanan veri seti, etiketlenmiş tümör ve tümörsüz görüntüler içermelidir. Veri seti daha sonra işlenerek modelin eğitimi için uygun hale getirilir.

2. **Veri Ön İşleme:** Veri ön işleme adımında toplanan beyin MR görüntüleri üzerinde çeşitli işlemler uygulanır. Bu işlemler arasında görüntü boyutunu yeniden şekillendirme, normalize etme, kontrast düzenleme ve gürültü azaltma gibi adımlar bulunabilir. Veri ön işleme adımı, görüntülerin daha iyi özellikler çıkarılabilmesi ve modelin daha etkili çalışabilmesi için önemlidir.
3. **Derin Öğrenme Modelinin Oluşturulması:** Beyin tümörü tespiti için bir derin öğrenme modeli oluşturulur. Bu genellikle bir CNN modelidir. CNN, görüntü verilerini analiz etmek için özel olarak tasarlanmış bir sinir ağı türüdür. Modelin mimarisi genellikle birkaç evrişim, havuzlama ve tam bağlantılı katmandan oluşur.
4. **Model Eğitimi:** Oluşturulan derin öğrenme modeli eğitim veri seti üzerinde eğitilir. Bu adımda modelin ağırlıkları ve parametreleri, eğitim veri setindeki görüntülerin özelliklerini ve tümör sınıflandırmalarını öğrenmek için güncellenir. Eğitim sürecinde kayıp fonksiyonu ve optimizasyon algoritması kullanılarak modelin performansı iyileştirilmeye çalışılır.
5. **Model Doğrulama ve Ayarlaması:** Eğitim sürecinin ardından modelin performansı doğrulama veri seti üzerinde değerlendirilir. Bu adımda modelin doğruluğunu, hassasiyetini, özgüllüğünü ve diğer performans metriklerini hesaplamak için test veri seti kullanılır. Modelin performansı değerlendirilerek, hiperparametrelerin (örneğin, öğrenme hızı, batch boyutu) ayarlanması ve modelin iyileştirilmesi gerçekleştirilir.
6. **Test ve Değerlendirme:** Son adımda eğitilmiş model gerçek dünya verileri üzerinde test edilir ve beyin tümörü tespiti yapılır. Modelin doğruluğu, tespit başarısı ve yanlış pozitif/negatif oranı gibi performans metrikleri kullanılarak değerlendirilir. Modelin gerçek dünya verilerinde yüksek doğruluk oranı elde etmesi hedeflenir.

5.2. Nitelikli Beyin MR Görüntülerinin Elde Edilmesi

Veri setleri, doğru ve güvenilir sonuçlar elde etmek için büyük önem taşır. Özellikle İnsan Sağlığının söz konusu olduğu bu seminerin merkezinde bulunan organımızın asgari düzeyde hatalı, mümkünse hatasız ve gürültüsüz ve yüksek çözünürlükte görüntülenebilmesi, uygulamaların sağlıklı çalışabilmesi, doğru tanı ve tespit için son derece önemlidir.

Bu başlık altında, iyi beyin MR görüntülerinin elde edilmesi için dikkate alınması gereken önemli faktörler açıklanacaktır.

1. **Manyetik Alan Gücü:** Beyin MR görüntülerinin kalitesi ve ayrıntı düzeyi, kullanılan manyetik alan gücüyle doğrudan ilişkilidir. Yüksek manyetik alan gücüne sahip MR cihazları daha yüksek çözünürlüklü ve daha net görüntüler sağlar. Bu nedenle iyi kalitede beyin MR görüntülerinin elde edilmesi için yüksek manyetik alan gücü tercih edilmelidir.
2. **Görüntüleme Parametreleri:** Beyin MR görüntülerinin elde edilmesi için kullanılan görüntüleme parametreleri, görüntü kalitesi üzerinde önemli bir etkiye sahiptir. Kesit kalınlığı, tekrarlanma süresi (TR), ekol zamanı (TE), sinyal/alım oranı (SNR), ters iyileştirme süresi (TI), paralelleştirme faktörü ve bobin seçimi gibi parametreler, görüntülerin kontrastı, gürültü düzeyi ve çözünürlüğü üzerinde etkili olabilir. Bu parametrelerin doğru bir şekilde belirlenmesi, iyi kalitede beyin MR görüntülerinin elde edilmesi için önemlidir (Bakas vd., 2017).
3. **Hasta Hazırlığı:** İyi kalitede beyin MR görüntülerinin elde edilmesi için hasta hazırlığı da önem taşır. Hasta rahat bir pozisyonda ve hareketsiz bir şekilde taranmalıdır. Hasta hareketi görüntü kalitesini olumsuz yönde etkileyebilir. Ayrıca uygun rahatlama teknikleri uygulanarak hasta rahatlatılmalıdır (Isensee vd., 2017).
4. **Görüntü Kalibrasyonu:** Beyin MR görüntülerinin elde edilmesi sürecinde cihazın doğru bir şekilde kalibre edilmesi önemlidir. Manyetik alan homojenliği, bobinlerin doğru yerleştirilmesi ve sinyal alım süresinin optimize edilmesi gibi faktörler göz önünde bulundurularak görüntü kalibrasyonu yapılmalıdır. Bu, daha iyi kontrast ve daha net görüntülerin elde edilmesine yardımcı olur (Menze vd., 2014).
5. **Hasta İşbirliği:** İyi kalitede beyin MR görüntülerinin elde edilmesi için hasta işbirliği de önemlidir. Hasta, talimatları doğru bir şekilde takip etmeli ve sakin bir şekilde hareketsiz durmalıdır. Ayrıca solunum veya kalp atışı gibi hareketlerin etkisini azaltmak için uygun teknikler kullanılabilir.

Bu faktörler, iyi kalitede beyin MR görüntülerinin elde edilmesi için dikkate alınması gereken önemli etmenlerdir. Doğru manyetik alan gücü seçimi, uygun görüntüleme parametrelerinin belirlenmesi, hasta hazırlığı, görüntü kalibrasyonu ve hasta işbirliği, başarılı bir şekilde beyin MR görüntülerinin elde edilmesini sağlayacaktır.

5.3. Uygulama 1: %77'ye Varan Kesinlik

1. Kullanılacak Kütüphanelerin Import Edilmesi: Kodun başında, kullanılacak olan kütüphaneler import edilir, yani projeye dahil edilir. Bu kütüphaneler gerekli fonksiyonları ve araçları sağlamak için kullanılır. Örneğin, os kütüphanesi dosya ve dizin işlemleri için kullanılırken, numpy kütüphanesi sayısal işlemler için kullanılır. keras kütüphanesi ise derin öğrenme modelleri oluşturmak ve eğitmek için kullanılır. Her kütüphanenin kendine has yapıları ve kullanım alanları bulunmaktadır.

```
# Kullanılacak kütüphaneler
import os
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping
from sklearn.metrics import classification_report, confusion_matrix
from keras.preprocessing import image
```

2. Veri Setinin Yüklenmesi: `load_data()` fonksiyonu, belirli bir dizindeki veri setini yüklemek için kullanılır. Bu fonksiyon, eğitim ve test verilerini içeren klasörleri okur ve görüntüleri yeniden boyutlandırarak ön işleme yapar. Son olarak, `ImageDataGenerator` kullanarak görüntüleri batch'lere bölen generator'ları döndürür. Bu proje, KAGGLE platformunun Notebook kısmı üzerinde yazılmış ve yürütülmüştür. Veri seti de KAGGLE üzerinden alınacaktır.

```
# Veriyi yükleme fonksiyonu
def load_data(train_dir, test_dir):
    # Görüntüleri yeniden ölçeklendirmek için bir ImageDataGenerator oluşturuyoruz
    datagen = ImageDataGenerator(rescale=1./255)

    # Eğitim verilerimizi yüklemek ve ön işlemleri gerçekleştirmek için bir generator oluşturuyoruz
    train_generator = datagen.flow_from_directory(
        train_dir,
        target_size=(150, 150),
        batch_size=32,
        class_mode='categorical')

    # Test verilerimizi yüklemek ve ön işlemleri gerçekleştirmek için bir generator oluşturuyoruz
    test_generator = datagen.flow_from_directory(
        test_dir,
        target_size=(150, 150),
        batch_size=32,
        class_mode='categorical')

    # Generatorları döndürüyoruz
    return train_generator, test_generator
```

Veri seti konumu: `/kaggle/input/brain-tumor-classification-mri/`

```
# Veriyi yükleme
train_dir = '/kaggle/input/brain-tumor-classification-mri/Training'
test_dir = '/kaggle/input/brain-tumor-classification-mri/Testing'
train_gen, test_gen = load_data(train_dir, test_dir)
```

```
Found 2870 images belonging to 4 classes.
Found 394 images belonging to 4 classes.
```

Sınıflar: `class_indices` fonksiyonu ile Sınıfları görüntülüyoruz.

```
test_gen.class_indices
```

```
{'glioma_tumor': 0, 'meningioma_tumor': 1, 'no_tumor': 2, 'pituitary_tumor': 3}
```

3. Modelin Oluşturulması: create_model() fonksiyonu derin öğrenme modelinin mimarisini oluşturur. Bu örnekte CNN kullanılmıştır. Model, çeşitli evrişimli ve pooling katmanlarını içerir. Flatten ve Dense katmanları ile çıktıya yönlendirilir ve softmax aktivasyon fonksiyonu ile sınıflandırma yapılır.

```
# Model oluşturma fonksiyonu
def create_model():
    model = Sequential()

    model.add(Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(150, 150, 3)))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Conv2D(128, (3, 3), padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))

    model.add(Dense(4, activation='softmax'))

    # Modeli döndürüyoruz
    return model
```

```
# Model oluşturma
model = create_model()
```

4. Modelin Derlenmesi: compile() fonksiyonu modelin derlenmesi için kullanılır. Bu adımda kayıp fonksiyonu (loss), optimize edici (optimizer) ve metrikler (metrics) belirlenir. Kayıp fonksiyonu olarak "categorical_crossentropy" kullanılmıştır çünkü çoklu sınıflandırma problemi söz konusudur. Optimizasyon algoritması olarak "Adam" seçilmiştir. Ayrıca, doğruluk metriği kullanılmaktadır.

```
# Modeli compile ediyoruz. Loss fonksiyonu olarak categorical_crossentropy kullanıyoruz çünkü çoklu sınıflandırma problemimiz var.
# Optimizasyon algoritması olarak Adam'ı kullanıyoruz. Metrik olarak accuracy'yi kullanıyoruz.
model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.001), metrics=['accuracy'])
```

5. Callbacks Tanımlanması: checkpoint ve early gibi callbacks, modelin eğitimi sırasında belirli durumlarda gerçekleşecek işlemleri belirler. ModelCheckpoint callback'i, her epoch sonunda modelin ağırlıklarını kaydederek en iyi modelin kaybolmasını önler. EarlyStopping callback'i ise belirli bir sayıda epoch boyunca modelin performansının artmaması durumunda eğitimi durdurur.

```
# Callbacks tanımlıyoruz. Bunlar modelin eğitimi sırasında belirli durumlarda gerçekleşecek işlemleri belirler
# ModelCheckpoint, her epoch sonunda modelin ağırlıklarını kaydeder. Böylece en iyi modeli kaybetmeyiz
# EarlyStopping, belirli bir sayıda epoch boyunca modelin performansının artmaması durumunda eğitimi durdurur

checkpoint = ModelCheckpoint("brain_tumor_detector.h5", monitor='val_accuracy', verbose=1, save_best_only=True, save_weights_only=False, mode='auto', period=1)
early = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=100, verbose=1, mode='auto')
```

6. Modelin Eğitimi: fit() fonksiyonu, modelin eğitimini gerçekleştirir. Bu adımda, eğitim ve doğrulama verileri, belirtilen epoch sayısı ve callbacks'ler kullanılarak model üzerinde eğitilir. Eğitim süreci boyunca, epoch başına düşen kayıp (loss) ve doğruluk (accuracy) değerleri kaydedilir.

```
# Modeli fit ediyoruz. Yani modeli eğitiyoruz. Eğitim verilerini ve doğrulama verilerini belirtiyoruz. Epoch sayısını ve callbacks'leri belirtiyoruz.
history = model.fit(train_gen, epochs=100, validation_data=test_gen, callbacks=[checkpoint,early])
```

```
Epoch 1/100
90/90 [=====] - ETA: 0s - loss: 0.9557 - accuracy: 0.6073
Epoch 1: val_accuracy improved from -inf to 0.41117, saving model to brain_tumor_detector.h5
90/90 [=====] - 28s 230ms/step - loss: 0.9557 - accuracy: 0.6073 - val_loss: 2.2181 - val_accuracy: 0.4112
Epoch 2/100
90/90 [=====] - ETA: 0s - loss: 0.5847 - accuracy: 0.7533
Epoch 2: val_accuracy improved from 0.41117 to 0.57614, saving model to brain_tumor_detector.h5
90/90 [=====] - 9s 99ms/step - loss: 0.5847 - accuracy: 0.7533 - val_loss: 1.7691 - val_accuracy: 0.5761
Epoch 3/100
90/90 [=====] - ETA: 0s - loss: 0.4305 - accuracy: 0.8408
Epoch 3: val_accuracy improved from 0.57614 to 0.59137, saving model to brain_tumor_detector.h5
90/90 [=====] - 9s 100ms/step - loss: 0.4305 - accuracy: 0.8408 - val_loss: 2.5350 - val_accuracy: 0.5914
Epoch 4/100
89/90 [=====>.] - ETA: 0s - loss: 0.2825 - accuracy: 0.8904
Epoch 4: val_accuracy improved from 0.59137 to 0.63198, saving model to brain_tumor_detector.h5
90/90 [=====] - 10s 113ms/step - loss: 0.2815 - accuracy: 0.8909 - val_loss: 2.0145 - val_accuracy: 0.6320
Epoch 5/100
90/90 [=====] - ETA: 0s - loss: 0.1867 - accuracy: 0.9307
Epoch 5: val_accuracy improved from 0.63198 to 0.65736, saving model to brain_tumor_detector.h5
90/90 [=====] - 10s 108ms/step - loss: 0.1867 - accuracy: 0.9307 - val_loss: 2.8939 - val_accuracy: 0.6574
Epoch 6/100
90/90 [=====] - ETA: 0s - loss: 0.1425 - accuracy: 0.9495
Epoch 6: val_accuracy improved from 0.65736 to 0.67766, saving model to brain_tumor_detector.h5
90/90 [=====] - 9s 100ms/step - loss: 0.1425 - accuracy: 0.9495 - val_loss: 3.8825 - val_accuracy: 0.6777
Epoch 7/100
90/90 [=====] - ETA: 0s - loss: 0.0922 - accuracy: 0.9690
Epoch 7: val_accuracy did not improve from 0.67766
90/90 [=====] - 8s 86ms/step - loss: 0.0922 - accuracy: 0.9690 - val_loss: 4.2958 - val_accuracy: 0.6675
Epoch 8/100
90/90 [=====] - ETA: 0s - loss: 0.0661 - accuracy: 0.9756
Epoch 8: val_accuracy improved from 0.67766 to 0.72589, saving model to brain_tumor_detector.h5
90/90 [=====] - 9s 103ms/step - loss: 0.0661 - accuracy: 0.9756 - val_loss: 3.9844 - val_accuracy: 0.7259
Epoch 9/100
90/90 [=====] - ETA: 0s - loss: 0.0716 - accuracy: 0.9735
Epoch 9: val_accuracy did not improve from 0.72589
90/90 [=====] - 8s 89ms/step - loss: 0.0716 - accuracy: 0.9735 - val_loss: 4.0255 - val_accuracy: 0.6929
Epoch 10/100
90/90 [=====] - ETA: 0s - loss: 0.0354 - accuracy: 0.9906
Epoch 10: val_accuracy did not improve from 0.72589
90/90 [=====] - 8s 92ms/step - loss: 0.0354 - accuracy: 0.9906 - val_loss: 4.4096 - val_accuracy: 0.7183
```

```

Epoch 70/100
90/90 [=====] - ETA: 0s - loss: 0.0163 - accuracy: 0.9948
Epoch 70: val_accuracy improved from 0.75635 to 0.76904, saving model to brain_tumor_detector.h5
90/90 [=====] - 9s 98ms/step - loss: 0.0163 - accuracy: 0.9948 - val_loss: 4.6999 - val_accuracy: 0.7690
Epoch 71/100
90/90 [=====] - ETA: 0s - loss: 0.0129 - accuracy: 0.9958
Epoch 71: val_accuracy did not improve from 0.76904
90/90 [=====] - 8s 90ms/step - loss: 0.0129 - accuracy: 0.9958 - val_loss: 4.9400 - val_accuracy: 0.7335
Epoch 72/100
90/90 [=====] - ETA: 0s - loss: 0.0135 - accuracy: 0.9965
Epoch 72: val_accuracy improved from 0.76904 to 0.77919, saving model to brain_tumor_detector.h5
90/90 [=====] - 10s 107ms/step - loss: 0.0135 - accuracy: 0.9965 - val_loss: 5.6965 - val_accuracy: 0.7792
Epoch 73/100
70/90 [=====] - ETA: 1s - loss: 0.0686 - accuracy: 0.9839

90/90 [=====] - 8s 84ms/step - loss: 2.0338e-04 - accuracy: 1.0000 - val_loss: 7.6472 - val_accuracy: 0.7766
Epoch 98/100
90/90 [=====] - ETA: 0s - loss: 2.3748e-04 - accuracy: 1.0000
Epoch 98: val_accuracy did not improve from 0.78680
90/90 [=====] - 8s 85ms/step - loss: 2.3748e-04 - accuracy: 1.0000 - val_loss: 7.0063 - val_accuracy: 0.7843
Epoch 99/100
90/90 [=====] - ETA: 0s - loss: 4.7975e-05 - accuracy: 1.0000
Epoch 99: val_accuracy did not improve from 0.78680
90/90 [=====] - 9s 101ms/step - loss: 4.7975e-05 - accuracy: 1.0000 - val_loss: 7.3622 - val_accuracy: 0.7766
Epoch 100/100
90/90 [=====] - ETA: 0s - loss: 3.6225e-05 - accuracy: 1.0000
Epoch 100: val_accuracy did not improve from 0.78680
90/90 [=====] - 8s 93ms/step - loss: 3.6225e-05 - accuracy: 1.0000 - val_loss: 7.4854 - val_accuracy: 0.7766

```

7. Modelin Değerlendirilmesi: evaluate() fonksiyonu, modelin test veri kümesi üzerinde değerlendirilmesini sağlar. Bu adımda, modelin kayıp ve doğruluk değerleri hesaplanır. Ayrıca, classification_report ve confusion_matrix fonksiyonları kullanılarak sınıflandırma sonuçları gösterilir.

```

12/12 [=====] - 1s 88ms/step - loss: 7.5829 - accuracy: 0.7760
Test accuracy: 0.7760416865348816
13/13 [=====] - 1s 78ms/step
Confusion Matrix
[[ 3 45 35 17]
 [10 44 43 18]
 [ 3 38 43 21]
 [ 2 24 29 19]]
Classification Report

```

	precision	recall	f1-score	support
glioma_tumor	0.17	0.03	0.05	100
meningioma_tumor	0.29	0.38	0.33	115
no_tumor	0.29	0.41	0.34	105
pituitary_tumor	0.25	0.26	0.26	74
accuracy			0.28	394
macro avg	0.25	0.27	0.24	394
weighted avg	0.25	0.28	0.25	394

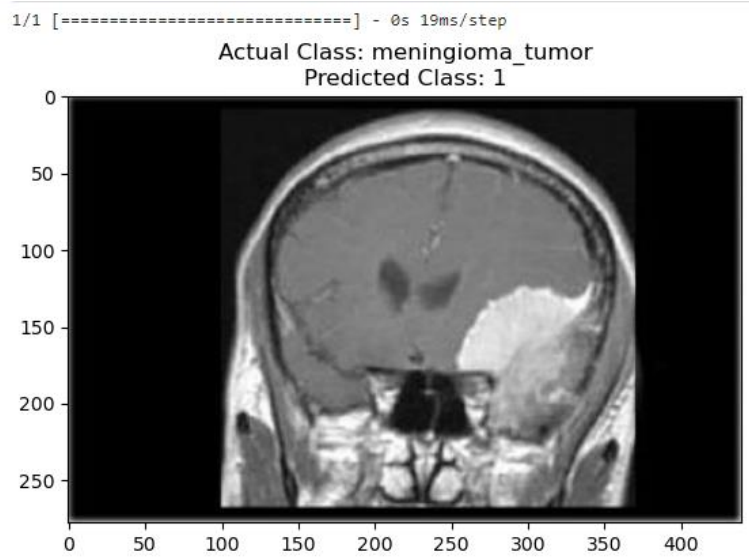
Karışıklık matrisi, bir sınıflandırma modelinin doğru ve yanlış tahminlerini görmenin bir yoludur. Her satır gerçek sınıfı temsil ederken, her sütun tahmin edilen sınıfı temsil eder. Matristeki değerler belirli bir gerçek sınıfın belirli bir tahmin edilen sınıfa ne kadar sık düştüğünü gösterir.

Sınıflandırma raporu, her sınıf için sınıflandırma metriklerini (precision, recall, f1-score) ve modelin genel doğruluk oranını içerir.

- **Precision**, modelinizin bir sınıfı doğru bir şekilde tahmin etme oranını gösterir. Yani, bir örneği belirli bir sınıf olarak tahmin ettiğinde, bu tahminin ne kadar sık doğru olduğunu gösterir.
- **Recall** (geri çağırma oranı), modelinizin belirli bir sınıfı ne kadar sık doğru bir şekilde tahmin ettiğini gösterir. Yani, bir sınıftaki tüm örnekleri ne kadar sık doğru bir şekilde tahmin ettiğini gösterir.
- **F1-Score**, precision ve recall'ın harmonik ortalamasıdır ve bir sınıflandırma modelinin genel performansını ölçer. Yüksek bir F1 skoru, modelin hem iyi precision hem de iyi recall'a sahip olduğunu gösterir.
- **Support**, her sınıftaki gerçek örneklerin sayısını gösterir.

Modelin doğruluğu eğitim setinde yaklaşık %77.6 iken, test setinde %28'dir. Bu, modelin overfitting (aşırı uyum) olduğunu gösterir. Yani model, eğitim verilerini çok iyi öğrenmiş ancak yeni verilere (test seti) genelleme yapma konusunda başarısız olmuştur. Sınıflandırma Raporunda istenilen değerler elde edilememiştir.

8. Tek Bir Görüntünün Test Edilmesi: Bu başlıkta da tek bir görüntü üzerinden bir test yapılacaktır. `test_image()` fonksiyonu, modelin tek bir görüntü üzerinde nasıl çalıştığını gösterir. Bu fonksiyon, belirli bir görüntüyü yükler, ön işleme yapar ve modeli kullanarak sınıflandırma yapar. Sonuçlar, gerçek sınıf ve tahmin edilen sınıf şeklinde ekrana yazdırılır.



Bu örnekte Model, Meningioma Tümör sınıfındaki bir görüntüyü = 1, yani Meningioma Tümör olarak tespit etmiş ve doğru sınıflandırmıştır.

5.4. Uygulama 2: %99'a Varan Kesinlik

1. Veri Ön İşleme: İlk olarak TensorFlow ve Keras kullanılarak bir ImageDataGenerator oluşturulmuştur. Bu nesne, görüntülerin ölçeklendirilmesi ve veri genişletme için kullanılır.

Rescale=1/255, görüntü piksellerini 0-1 aralığına ölçeklendirmek için kullanılır.

```
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
import cv2 as cv
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from datetime import datetime
from sklearn.model_selection import train_test_split
```

flow_from_directory metodu, belirli bir dizinden görüntüleri yükler ve otomatik olarak etiketler. Veri setindeki her alt dizin, bir sınıfı temsil eder:

```
imgd=ImageDataGenerator(rescale=1/255)
```

Veri seti dört sınıfa ayrılmıştır ve her bir sınıfın kaç tane örneği olduğu belirlenir:

```
tumor_dataset=imgd.flow_from_directory('../input/brain-tumor-classification-mri/Training')
```

Found 2870 images belonging to 4 classes.

class_indices fonksiyonu ile Sınıfları görüntülüyoruz.

```
tumor_dataset.class_indices
```

```
{'glioma_tumor': 0, 'meningioma_tumor': 1, 'no_tumor': 2, 'pituitary_tumor': 3}
```

2. Model Oluşturma: CNN kullanılarak bir model oluşturulur. Model, birkaç katmandan oluşur: Conv2D, Activation, MaxPool2D, Flatten, Dense. Her bir katmanın belirli bir işlevi vardır.

```
path_train = '../input/brain-tumor-classification-mri/Training'
path_test = '../input/brain-tumor-classification-mri/Testing'

import tensorflow as tf
import tensorflow.keras.layers as tfl

train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_data = train_datagen.flow_from_directory(
    path_train,
    subset='training',
    target_size=(200 , 200),
    batch_size=32)

val_data = train_datagen.flow_from_directory(
    path_train,
    subset='validation',
    target_size=(200 , 200),
    batch_size=32 )
test_data = train_datagen.flow_from_directory(
    path_test,
    target_size=(200 , 200),
    batch_size=32 )
```

Found 2297 images belonging to 4 classes.
Found 573 images belonging to 4 classes.
Found 394 images belonging to 4 classes.

```
from keras.models import Sequential
from keras.layers import Flatten, Activation, Dense, Dropout, Conv2D, MaxPool2D
```

```
model =Sequential()

#convolution and maxpooling layer
model.add(Conv2D(filters=25, kernel_size=3,
                  strides=2, padding='valid', input_shape=(200,200,3)))
model.add(Activation('relu'))
model.add(MaxPool2D(pool_size=2))

#flatten layer
model.add(Flatten())

#hidden layer
model.add(Dense(16))
model.add(Activation('relu'))

#output layer
model.add(Dense(4))
model.add(Activation('sigmoid'))

model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 99, 99, 25)	700
activation (Activation)	(None, 99, 99, 25)	0
max_pooling2d (MaxPooling2D)	(None, 49, 49, 25)	0
flatten (Flatten)	(None, 60025)	0
dense (Dense)	(None, 16)	960416
activation_1 (Activation)	(None, 16)	0
dense_1 (Dense)	(None, 4)	68
activation_2 (Activation)	(None, 4)	0
Total params: 961,184		
Trainable params: 961,184		
Non-trainable params: 0		

Sequential, Keras kütüphanesinde bulunan bir sınıftır. Bu sınıf, derin öğrenme modellerini oluşturmak için kullanılır. Sequential sınıfı, ardışık bir model oluşturmayı sağlar, yani katmanları sırayla eklemek ve birbirine bağlamak için kullanılır. Sequential modelde katmanlar tek bir girişten tek bir çıkışa doğru bir akışta birleştirilir. Bu nedenle, bu tür modeller genellikle sınıflandırma veya regresyon gibi basit görevler için kullanılır.

Sequential modelin kullanımı oldukça kolaydır. Modeli oluşturmak için sırayla katmanları ekleyebilirsiniz. Örneğin, Convolutional, Pooling ve Dense (tam bağlantılı) gibi çeşitli katmanları ekleyerek modelinizi oluşturabilirsiniz. Modeli derledikten sonra, verileri modele besleyerek eğitim ve tahmin işlemlerini gerçekleştirebilirsiniz. Sequential modelin avantajlarından biri, basit ve sezgisel bir kullanıma sahip olmasıdır. Ayrıca, tek giriş-çıkış akışı sayesinde basit modellerin oluşturulmasını sağlar. Ancak, bu model tipi, daha karmaşık yapılar veya çoklu giriş-çıkış akışları gerektiren problemler için uygun olmayabilir.

Sequential model, genellikle basit CNN (Convolutional Neural Network) veya MLP (Multi-Layer Perceptron) gibi yapıları uygulamak için tercih edilir. Daha karmaşık modeller oluşturmak için Functional API veya Subclassing gibi diğer Keras API'lerine başvurmak gerekebilir.

3. Model Compile: model.compile ile modelin öğrenme süreci yapılandırılır. Bu durumda, optimizasyon algoritması olarak 'adam' kullanılır ve kayıp fonksiyonu olarak 'categorical_crossentropy' kullanılır.

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

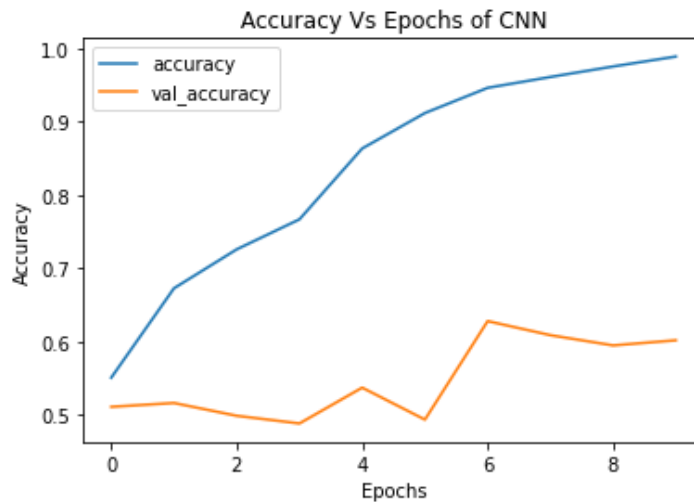
4. Model Eğitimi: Model model.fit metodu ile eğitilir. Bu aşamada model, eğitim veri setini kullanarak öğrenir ve doğrulama veri setini kullanarak tahminlerin doğruluğunu değerlendirir. Eğitim belirtilen sayıda epoch boyunca gerçekleştirilir.

```
history=model.fit (train_data,epochs=10, validation_data=val_data)
```

Epoch 1/10
72/72 [=====] - 31s 334ms/step - loss: 1.0899 - accuracy: 0.5512 - val_loss: 1.0773 - val_accuracy: 0.5113
Epoch 2/10
72/72 [=====] - 12s 169ms/step - loss: 0.7889 - accuracy: 0.6731 - val_loss: 1.0709 - val_accuracy: 0.5166
Epoch 3/10
72/72 [=====] - 13s 182ms/step - loss: 0.6857 - accuracy: 0.7262 - val_loss: 1.0715 - val_accuracy: 0.4991
Epoch 4/10
72/72 [=====] - 12s 170ms/step - loss: 0.5750 - accuracy: 0.7671 - val_loss: 1.2135 - val_accuracy: 0.4887
Epoch 5/10
72/72 [=====] - 13s 179ms/step - loss: 0.3842 - accuracy: 0.8637 - val_loss: 1.0759 - val_accuracy: 0.5375
Epoch 6/10
72/72 [=====] - 12s 168ms/step - loss: 0.2582 - accuracy: 0.9121 - val_loss: 1.2947 - val_accuracy: 0.4939
Epoch 7/10
72/72 [=====] - 12s 160ms/step - loss: 0.1859 - accuracy: 0.9465 - val_loss: 1.0900 - val_accuracy: 0.6283
Epoch 8/10
72/72 [=====] - 13s 178ms/step - loss: 0.1515 - accuracy: 0.9613 - val_loss: 1.2331 - val_accuracy: 0.6091
Epoch 9/10
72/72 [=====] - 12s 171ms/step - loss: 0.1066 - accuracy: 0.9756 - val_loss: 1.2737 - val_accuracy: 0.5951
Epoch 10/10
72/72 [=====] - 13s 177ms/step - loss: 0.0686 - accuracy: 0.9891 - val_loss: 1.4060 - val_accuracy: 0.6021

Eğitim sonrası, modelin eğitim ve doğrulama doğrulukları grafiği çizilir.

```
plt.plot(history.history['accuracy'],label='accuracy')  
plt.plot(history.history['val_accuracy'],label='val_accuracy')  
plt.legend()  
plt.title('Accuracy vs CNN Epochs')  
plt.xlabel('Epochs')  
plt.ylabel('Accuracy');
```



5. Modelin Değerlendirilmesi `evaluate()` fonksiyonu, modelin test veri kümesi üzerinde değerlendirilmesini sağlar. Bu adımda, modelin kayıp ve doğruluk değerleri hesaplanır. Ayrıca, `classification_report` ve `confusion_matrix` fonksiyonları kullanılarak sınıflandırma sonuçları gösterilir.

```
model.evaluate(train_data)
model.evaluate(test_data)
```

```
72/72 [=====] - 12s 171ms/step - loss: 0.0470 - accuracy: 0.9943
13/13 [=====] - 2s 143ms/step - loss: 3.3311 - accuracy: 0.6675
[3.3310999870300293, 0.6675127148628235]
```

Modelin doğruluğu eğitim setinde yaklaşık %99.4 iken, test setinde yaklaşık %66.8'dir. Bu, modelin overfitting (aşırı uyum) olduğunu gösterir. Yani model, eğitim verilerini çok iyi öğrenmiş ancak yeni verilere (test seti) genelleme yapma konusunda başarısız olmuştur.

Modelin tahminlerinin ne olduğunu görmek için `model.predict` kullanılır. Bu, her bir sınıf için olasılıklar döndürür.

```
y_predicte=model.predict(test_data)
y_predicte
```

```
array([[9.21069145e-01, 3.99352551e-01, 5.41547287e-08, 2.65669886e-09],
       [4.68248397e-01, 9.99999881e-01, 1.00000000e+00, 8.93124416e-02],
       [7.60874569e-01, 9.99438226e-01, 9.53092992e-01, 5.85444068e-05],
       ...,
       [6.41239703e-01, 9.42773461e-01, 9.87101197e-01, 2.12087482e-01],
       [9.55228731e-02, 9.99228835e-01, 9.99158978e-01, 8.74878764e-01],
       [9.69159782e-01, 3.10083311e-02, 4.16957562e-08, 1.43488625e-08]],
      dtype=float32)
```

6. Tek Bir Görüntünün Test Edilmesi: Bu başlıkta da tek bir görüntü üzerinden bir test yapılacaktır. `test_image()` fonksiyonu, modelin tek bir görüntü üzerinde nasıl çalıştığını gösterir. Bu fonksiyon, belirli bir görüntüyü yükler, ön işleme yapar ve modeli kullanarak sınıflandırma yapar. Sonuçlar, gerçek sınıf ve tahmin edilen sınıf şeklinde ekrana yazdırılır.

```
import os
from PIL import Image

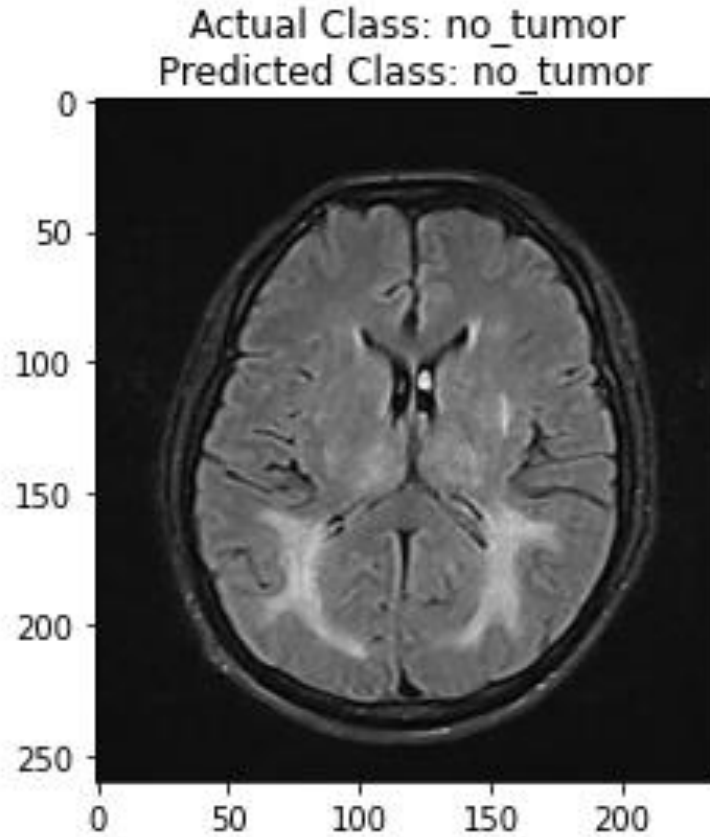
def test_image(file_path, model):
    img = Image.open(file_path).resize((200, 200)) # adjust the image size to match with the model's input shape
    img_array = np.array(img)/255.0 # normalize the image as we did before
    img_array = np.expand_dims(img_array, axis=0)

    prediction = model.predict(img_array)
    predicted_class = np.argmax(prediction)

    actual_class = os.path.basename(os.path.dirname(file_path))

    img = Image.open(file_path)
    plt.imshow(img)
    plt.title(f"Actual Class: {actual_class}\nPredicted Class: {tumor_classes[predicted_class]}")
    plt.show()

test_image("../input/brain-tumor-classification-mri/Testing/no_tumor/image(10).jpg", model)
```



Bu örnekte Model, No-Tumor sınıfındaki bir görüntüyü doğru sınıflandırmıştır.

6. BÖLÜM

6. GÖRÜNTÜ SINIFLANDIRMADA EN ÇOK KULLANILAN TEKNOLOJİLER

Görüntü sınıflandırma, görüntü verilerini farklı sınıflara ayırarak nesneleri tanıma ve sınıflandırma işlemidir. Bu alanda birçok farklı algoritma, programlama dili ve kütüphane kullanılmaktadır. Görüntü sınıflandırmada en yaygın olarak kullanılan teknolojiler, algoritmalar, programlama dilleri, modeller ve kütüphaneler ilerleyen kısımlarda anlatılacaktır.

6.1. Görüntü Sınıflandırmada En Çok Kullanılan Algoritmalar

Görüntü sınıflandırma bilgisayarlı görü tarafından elde edilen görüntülerin belirli sınıflara atanmasıyla ilgilenen bir alandır. Bu alanda en çok kullanılan algoritmalar görüntü özelliklerini yakalamak ve sınıflandırmak için güçlü matematiksel ve istatistiksel modeller kullanır. Aşağıda en çok kullanılan görüntü sınıflandırma algoritmalarını detaylı bir şekilde tanıtılacaktır.

- **Evrşimli Sinir Ağları (Convolutional Neural Networks - CNN):** Görüntü işlemede büyük bir başarı elde eden derin öğrenme algoritmasıdır. CNN çok katmanlı yapısı ve evrşim, pooling ve tam bağlantılı katmanlar gibi özel tasarlanmış bileşenleri sayesinde görüntü verilerini etkili bir şekilde işleyebilir ve sınıflandırabilir. AlexNet, VGGNet, ResNet ve Inception gibi ünlü CNN modelleri, görüntü sınıflandırmada büyük başarılar elde etmiştir (Krizhevsky vd., 2017; Simonyan ve Zisserman, 2014; He vd., 2016; Szegedy vd., 2015).
- **Destek Vektör Makineleri (Support Vector Machines - SVM):** SVM, doğrusal ve doğrusal olmayan sınıflandırma problemlerinde etkili olan bir makine öğrenmesi algoritmasıdır. Görüntü sınıflandırmada SVM, görüntü özelliklerini vektörler olarak temsil ederek sınıflandırma yapar. SVM'nin avantajlarından biri yüksek boyutlu verilerin işlenebilmesidir. SVM, margin maksimizasyonu prensibi üzerine kuruludur ve karar sınırlarını belirlemek için kernel fonksiyonları kullanır (Vapnik, 1999).

- **Karar Ağaçları (Decision Trees):** Karar ağaçları, görüntü özelliklerini kullanarak ağaç yapısı oluşturan bir sınıflandırma algoritmasıdır. Her iç düğümde bir özellik testi yapılır ve yaprak düğümlerde sınıflandırma yapılır. Karar ağaçları basitlik ve yorumlanabilirlik avantajlarına sahiptir. Random Forest ve Gradient Boosting gibi yöntemlerle birleştirildiğinde daha güçlü bir sınıflandırma performansı elde edilebilir.

Bu algoritmaların her biri, görüntü sınıflandırma alanında farklı avantajlara sahiptir. Bazıları derin öğrenme ile yüksek hassasiyet ve doğruluk sağlarken, diğerleri basitlik ve yorumlanabilirlik açısından daha ön plana çıkar. Bu nedenle uygulama ihtiyaçlarına bağlı olarak en uygun algoritmanın seçilmesi önemlidir.

6.2. Görüntü İşlemede En Çok Kullanılan Programlama Dilleri

Görüntü işleme alanında, programlama dili seçimi büyük bir öneme sahiptir. Hem hızlı prototipleme hem de performans açısından uygun bir dil seçmek önemlidir. Aşağıda görüntü işlemede en çok kullanılan programlama dillerini tanıtacağım.

- **Python:** Python, görüntü işleme topluluğu tarafından yaygın olarak kullanılan bir programlama dilidir. Geniş bir kütüphane ekosistemi ve kolay kullanımı ile öne çıkar. OpenCV ve TensorFlow gibi kütüphaneler Python üzerinde görüntü işleme için güçlü araçlar sağlar.
- **C++:** Görüntü işlemede performans gereksinimleri yüksek olduğunda, C++ tercih edilen bir dildir. C++'ın düşük seviye programlama yetenekleri ve verimli bellek yönetimi, hızlı ve etkili görüntü işleme uygulamalarının geliştirilmesine olanak sağlar (Bradski ve Kaehler, 2008).
- **MATLAB:** MATLAB, matriks tabanlı hesaplama yetenekleri ile görüntü işleme için yaygın olarak kullanılan bir programlama dili ve ortamıdır. MATLAB zengin bir görüntü işleme kütüphanesi içerir ve kullanıcı dostu bir arayüze sahiptir. Özellikle akademik ve araştırma ortamlarında sıkça tercih edilir (Gonzalez vd., 2004).

Bu diller, farklı avantajlara ve kullanım senaryolarına sahiptir. Python, hızlı prototipleme ve geniş bir kütüphane desteği sağlarken, C++ performans gereksinimlerini karşılamak için tercih edilir. MATLAB ise matriks hesaplama yetenekleri ve kullanıcı dostu arayüzü ile öne çıkar.

6.3. Görüntü İşlemede En Çok Kullanılan Kütüphaneler

Görüntü işleme alanında kütüphaneler, görüntü verilerini işlemek, analiz etmek ve sınıflandırmak için önemli araçlardır. Aşağıda görüntü işlemede en çok kullanılan kütüphaneler tanıtılacaktır.

- **OpenCV:** OpenCV, açık kaynaklı bir görüntü işleme kütüphanesidir ve çok geniş bir işlevsellik sunar. Hem Python hem de C++ için kullanılabilir. Görüntü işleme operasyonları, nesne algılama, yüz tanıma, görüntü iyileştirme gibi birçok işlemi destekler (Bradski ve Kaehler, 2008).
- **scikit-image:** scikit-image, Python tabanlı bir görüntü işleme kütüphanesidir. Yüksek düzeyde işlevsellik sunar ve farklı görüntü işleme operasyonlarını gerçekleştirmek için çeşitli fonksiyonlar içerir. Görüntü filtreleme, kenar tespiti, morfolojik operasyonlar gibi işlemleri kolayca gerçekleştirebiliriz.
- **TensorFlow:** TensorFlow, derin öğrenme ve makine öğrenme modellerinin geliştirilmesi için yaygın olarak kullanılan bir kütüphanedir. Görüntü sınıflandırma, nesne algılama, segmentasyon gibi görüntü işleme problemlerini çözmek için kullanılabilir. Python ve diğer programlama dilleri için kullanılabilir.

Bu kütüphaneler, görüntü işleme alanında birçok işlevi yerine getirmek için kullanılır. OpenCV, temel işlemleri ve görüntü analizini desteklerken, scikit-image daha yüksek seviye işlemler için kullanılır. TensorFlow ise derin öğrenme ve karmaşık model oluşturma için tercih edilen bir kütüphanedir.

6.4. Görüntü Sınıflandırmada En Çok Kullanılan Modeller

Görüntü sınıflandırma, bir görüntünün bir veya daha fazla sınıfa ait olduğunu belirlemek için kullanılan bir işlemdir. Bu alanda birçok başarılı model ve mimari geliştirilmiştir. Aşağıda, görüntü sınıflandırmada en çok kullanılan ve popüler olan 10 model tanıtılacaktır.

1. **AlexNet:** AlexNet, 2012 yılında Alex Krizhevsky tarafından geliştirilen derin bir sinir ağı modelidir. Görüntü sınıflandırma alanında çığır açan bu model, ImageNet veri kümesi üzerinde o dönemdeki diğer modellere göre çok daha iyi performans gösterdi. AlexNet'in en dikkat çeken özelliği, 5 konvolüsyon ve 3 tam bağlantılı katmandan oluşmasıdır. Model, ReLU aktivasyon fonksiyonu, max pooling ve dropout gibi öğrenmeyi iyileştiren teknikleri kullanır. AlexNet, özellikle derin öğrenme alanında popüler hale gelmesine katkıda bulunmuştur (Krizhevsky vd., 2017)
2. **VGG16:** VGG16, Oxford Üniversitesi'nden bir araştırma grubu tarafından geliştirilen bir modeldir. Modelin ismi 16 adet konvolüsyon katmanına sahip olmasından gelir. VGG16, çok derin bir yapıya sahip olmasına rağmen, basit bir mimariye sahiptir. Tüm konvolüsyon katmanlarında 3x3 boyutunda filtreler kullanılır. Bu sayede daha küçük boyutlu filtrelerin kullanılmasıyla daha derin özelliklerin öğrenilmesi sağlanır. VGG16 özellikle görüntü sınıflandırmada yüksek başarı elde etmesiyle tanınır (Simonyan ve Zisserman, 2014).
3. **ResNet50:** ResNet50, Microsoft Research tarafından geliştirilen bir modeldir. Residual ağlarının (Residual Networks) temsilcisi olan bu model aşırı öğrenmeyi azaltmak için "skip connection" adı verilen bir yapı kullanır. Bu yapı önceki katmanın çıktısını bir sonraki katmana direkt olarak aktarmak suretiyle gradient kaybını önler ve daha derin ağların eğitilebilir olmasını sağlar. ResNet50, 50 adet konvolüsyon katmanından oluşur ve ImageNet veri kümesindeki çeşitli sınıfları tanımada yüksek başarı elde eder (He vd., 2016).
4. **InceptionV3:** InceptionV3, Google tarafından geliştirilen bir modeldir. İlk olarak GoogLeNet olarak bilinen Inception mimarisine dayanır. InceptionV3, 48 konvolüsyon katmanından oluşan karmaşık bir yapıya sahiptir. Özellikle 1x1, 3x3 ve 5x5 boyutunda filtrelerin paralel olarak kullanılması ve daha sonra bu çıktıların birleştirilmesiyle özellik çıkarımı gerçekleştirir. Bu mimari hem hesaplama kaynağını etkin bir şekilde kullanır hem de daha geniş bir özellik yelpazesi sağlar (Szegedy vd., 2015).

5. **MobileNetV2:** MobileNetV2, Google tarafından geliştirilen hafif ve taşınabilir bir modeldir. Mobil cihazlarda kullanılmak üzere tasarlanmıştır ve düşük hesaplama gücüne sahip cihazlarda bile yüksek performans sunar. MobileNetV2, özellikle derinlik-ayrıştırma (depthwise separable) konvolüsyonları kullanır. Bu yapı hesaplama maliyetini azaltırken modelin etkinliğini korumasını sağlar. Aynı zamanda genişletme katmanları da kullanarak modelin öğrenme kapasitesini artırır (Sandler vd., 2018).
6. **DenseNet:** Bu modelde, her katman, önceki tüm katmanlardan gelen çıktıları alır ve bu çıktıları kendi çıktısıyla birleştirir. Bu yoğun bağlantılar, bilgi akışını artırır ve modelin öğrenme kapasitesini artırır. Ayrıca, parametre sayısını da azaltır. DenseNet, özellikle sınırlı veri setleriyle çalışırken etkili sonuçlar verir (Huang vd., 2017).
7. **NASNet:** NASNet, Google Brain tarafından geliştirilen bir modeldir. Modelin özelliği otomatik makine öğrenimi (AutoML) tekniklerinin kullanılarak özelleştirilmiş bir yapı oluşturulmasıdır. Bu yöntemle birçok farklı model yapılandırması deneyerek en iyi yapıyı otomatik olarak belirler. NASNet, özellikle hesaplama maliyetini azaltarak ve daha yüksek doğruluk sağlayarak etkili sonuçlar elde eder (Chollet vd., 2017).
8. **EfficientNet:** EfficientNet, Google Brain tarafından geliştirilen bir modeldir. Bu model genişlik, derinlik ve çözünürlük gibi faktörleri dikkate alarak ölçeklendirme katsayıları kullanır. Bu sayede modelin boyutu ve hesaplama kaynağına ihtiyaç duyan derinlik ayarlanabilir. EfficientNet özellikle hesaplama maliyetini azaltırken yüksek doğruluk elde etme yeteneğiyle bilinir (Xie vd., 2017).
9. **Xception:** Xception, François Chollet tarafından geliştirilen bir modeldir. Xception, Inception mimarisine dayanır ancak konvolüsyon katmanlarında "depthwise separable convolution" adı verilen özel bir yapı kullanır. Bu yapı hesaplama kaynaklarını daha etkin bir şekilde kullanır ve daha az parametreye ihtiyaç duyar. Xception, özellikle görüntü sınıflandırmada yüksek performans gösterir (Tan vd., 2019).
10. **SqueezeNet:** SqueezeNet, Berkeley Üniversitesi tarafından geliştirilen hafif bir modeldir. Model derin ağlardan daha küçük bir model boyutu ve daha az hesaplama kaynağı gerektiren bir yapıya sahiptir. SqueezeNet, "squeeze" ve "expand" adı verilen özel konvolüsyon katmanları kullanarak sıkıştırma ve genişletme işlemlerini gerçekleştirir. Bu sayede, daha küçük model boyutlarıyla bile etkili sonuçlar elde eder (Iandola vd., 2016).

Bu modeller, görüntü sınıflandırmada geniş bir kullanıma sahiptir ve birçok görüntü işleme uygulamasında başarılı sonuçlar vermiştir. Her bir model, farklı mimari yapılar ve özellikler sunarak çeşitli problemlere uyum sağlamaktadır.

7. SONUÇ VE ÖNERİLER

7. SONUÇ VE ÖNERİLER

7.1. Derin Öğrenme Yöntemlerinin Beyin Tümörü Tespitindeki Başarıları

Bu çalışma, derin öğrenme yöntemlerinin beyin tümörü tespiti üzerindeki başarılarını incelemek ve değerlendirmek amacıyla gerçekleştirilmiştir. İki farklı derin öğrenme modeli kullanılarak gerçekleştirilen uygulamaların sonuçları analiz edilmiştir. Ancak elde edilen sonuçlar bazı zorluklar ve sınırlamalarla karşılaşmıştır.

Model 1, eğitim setinde yaklaşık %77.6 kesinlik elde etmiştir. Ancak test setindeki doğruluk oranı %28 olarak belirlenmiştir. Bu sonuçlar, modelin eğitim verilerine aşırı uyum sağladığını ve genelleme yapma yeteneğinin düşük olduğunu göstermektedir. Ayrıca, sınıflandırma raporunda istenilen değerler elde edilememiştir. Bu durum, modelin niteliksiz ve yetersiz veriyle eğitildiğini ve daha iyi bir veri toplama sürecinin gerekliliğini göstermektedir.

Model 2, eğitim setinde yaklaşık %99.4 kesinlik elde etmiştir. Ancak test setindeki doğruluk oranı yaklaşık %66.8 olarak belirlenmiştir. Bu da modelin aşırı uyum sorunu yaşadığını ve genelleme yeteneğinin düşük olduğunu göstermektedir. Niteliksiz ve yetersiz verinin bu sonuçlara yol açtığı görülmektedir.

Her iki uygulama da bir adet fotoğraf ile nöronlar denenmiş ve fotoğrafın hangi sınıfa ait olduğu doğru tahmin edilebilmiştir. Bu, modelin bazı durumlarda doğru sonuçlar verebildiğini göstermektedir.

Sonuç olarak, bu çalışmada kullanılan modellerin doğruluk oranları düşük olmuş ve aşırı uyum sorunu yaşanmıştır. Bu durum, niteliksiz ve yetersiz verinin model performansını etkilediğini göstermektedir. Görüntü sınıflandırma problemlerinde daha yüksek doğruluk elde etmek için daha fazla veri toplanması ve daha dikkatli bir veri ön işleme süreci uygulanması gerekmektedir.

7.2. Öneriler

Bu çalışmanın sonuçları ve analizlerine dayanarak, derin öğrenme yöntemlerinin beyin tümörü tespitindeki başarılarını artırmak için aşağıdaki öneriler sunulmuştur:

1. **Daha Büyük ve Çeşitli Veri Kümesi:** Modelin daha iyi genelleme yapabilmesi için daha büyük ve çeşitli bir veri kümesi kullanılmalıdır. Farklı hastanelerden ve farklı MR cihazlarından elde edilen görüntülerin dahil edilmesi, modelin daha çeşitli örnekleri tanımayı öğrenmesini sağlayabilir.
2. **Veri Ön İşleme ve Veri Dengeleme:** Veri ön işleme aşamasında daha fazla dikkat gösterilmelidir. Görüntülerin düzgün bir şekilde yeniden boyutlandırılması, normalize edilmesi ve gürültüden arındırılması önemlidir. Ayrıca, sınıf dengesizliği gibi sorunlarla da ilgilenilmelidir.
3. **Model Optimizasyonu:** Modelin daha iyi performans göstermesi için hiperparametrelerin iyileştirilmesi gerekmektedir. Farklı aktivasyon fonksiyonları, optimizasyon algoritmaları ve derinlik/darlık ayarlamaları denenmelidir.
4. **Transfer Öğrenme:** Mevcut başarılı görüntü sınıflandırma modellerinden öğrenme transferi yapılabilir. Bu yöntem, daha az veriyle daha iyi sonuçlar elde etmeyi sağlayabilir.
5. **Diğer Görüntü İşleme Yöntemleri:** Görüntü sınıflandırmada derin öğrenme dışında diğer geleneksel görüntü işleme yöntemlerinin de kullanılması düşünülmelidir. Özellik çıkarma ve sınıflandırma algoritmaları gibi yöntemler değerlendirilebilir.

Bu çalışma, derin öğrenme yöntemlerinin beyin tümörü tespiti gibi önemli bir uygulama alanında nasıl kullanılabileceğini göstermiştir. Ancak, daha fazla çalışma ve iyileştirme yapılması gerekmektedir. Önerilen yöntemlerin uygulanması, daha iyi sonuçlar elde etmek ve klinik ortamlarda kullanılabilirliği artırmak açısından önemlidir.

KAYNAKÇA

- Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2016). An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE journal of biomedical and health informatics*, 21(1), 31-40.
- Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. (2018). H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE transactions on medical imaging*, 37(12), 2663-2674.
- Maxwell, A., Li, R., Yang, B., Weng, H., Ou, A., Hong, H., ... & Zhang, C. (2017). Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC bioinformatics*, 18, 121-131.
- LeCun, Y., Bengio, Y., ve Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Louis, D. N., Perry, A., Reifenberger, G., Von Deimling, A., Figarella-Branger, D., Cavenee, W. K., ... & Ellison, D. W. (2016). The 2016 World Health Organization classification of tumors of the central nervous system: a summary. *Acta neuropathologica*, 131, 803-820.
- Ostrom, Q. T., Gittleman, H., Truitt, G., Boscia, A., Kruchko, C., & Barnholtz-Sloan, J. S. (2018). CBTRUS statistical report: primary brain and other central nervous system tumors diagnosed in the United States in 2011–2015. *Neuro-oncology*, 20(suppl_4), iv1-iv86.
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.
- Smith-Bindman, R., Kwan, M. L., Marlow, E. C., Theis, M. K., Bolch, W., Cheng, S. Y., ... & Miglioretti, D. L. (2019). Trends in use of medical imaging in US health care systems and in Ontario, Canada, 2000-2016. *Jama*, 322(9), 843-856.

- Mayo Clinic. (2023). Brain tumor. <https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084> .19 Mayıs 2023 tarihi saat 01:00'de ilgili adresten alınmıştır.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4* (pp. 3-11). Springer International Publishing.
- National Institute of Biomedical Imaging and Bioengineering. (2023). Magnetic Resonance Imaging (MRI). <https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri> .19 Mayıs 2023 tarihi saat 01:30'da ilgili adresten alınmıştır.
- Mayo Clinic. (2023). Brain tumor: Diagnosis & treatment. <https://www.mayoclinic.org/diseases-conditions/brain-tumor/diagnosis-treatment/drc-20350088> .19 Mayıs 2023 tarihi saat 02:00'de ilgili adresten alınmıştır.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
- Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.
- Wang, S., Zhou, M., Liu, Z., Liu, Z., Gu, D., Zang, Y., ... & Tian, J. (2017). Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation. *Medical image analysis*, 40, 172-183.

- Türk Nöroşirürji Derneği. (2023). Beyin Tümörleri. <https://turknorosirurji.org.tr/menu/68/beyin-tumorleri> .10 Haziran 2023 tarihi saat 01:00’de ilgili adresten alınmıştır.
- Stupp, R., Mason, W. P., Van Den Bent, M. J., Weller, M., Fisher, B., Taphoorn, M. J., ... & Mirimanoff, R. O. (2005). Radiotherapy plus concomitant and adjuvant temozolomide for glioblastoma. *New England journal of medicine*, 352(10), 987-996.
- Weller, M., Van Den Bent, M., Hopkins, K., Tonn, J. C., Stupp, R., Falini, A., ... & Wick, W. (2014). EANO guideline for the diagnosis and treatment of anaplastic gliomas and glioblastoma. *The lancet oncology*, 15(9), e395-e403.
- Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J. S., ... & Davatzikos, C. (2017). Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific data*, 4(1), 1-13.
- Isensee, F., Kickingreder, P., Wick, W., Bendszus, M., & Maier-Hein, K. H. (2018). Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3* (pp. 287-297). Springer International Publishing.
- Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., ... & Van Leemput, K. (2014). The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE transactions on medical imaging*, 34(10), 1993-2024.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.
- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc."
- Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). Digital image using Matlab processing. *Person Prentice Hall, Lexington*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
- Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).
- Tan, M., & Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv 2019. arXiv preprint arXiv:1905.11946.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv preprint arXiv:1602.07360.