

T.C.
İstanbul
YENİ YÜZYIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİYOMEDİKAL MÜHENDİSLİĞİ ANA BİLİM DALI



Yapay Zeka Algoritmaları ile Beyin Tümörlerinin Teşhisi

YÜKSEK LİSANS TEZİ

Omar KHALIL

Tez Danışmanı

Dr. Öğr. Üyesi Diyadin CAN

İSTANBUL

Ocak 2024

T.C.
İstanbul
YENİ YÜZYIL ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Kabul Onay

Biyomedikal Mühendisliği Ana Bilim Dalı Yüksek Lisans Programı çerçevesinde yürütölmüş olan bu çalışma aşağıdaki jüri tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 25.01.2024

Tez Danışmanı

Dr. Öğr. Üyesi Diyadin CAN
İstanbul Yeni Yüzyıl Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Diyadin CAN
İstanbul Yeni Yüzyıl Üniversitesi

Doç. Dr. Değır SOFUOĞLU
İstanbul Üniversitesi

Dr. Öğr. Üyesi Şakir ALTINSOY
İstanbul Yeni Yüzyıl Üniversitesi

ETİK BEYAN

İstanbul Yeni Yüzyıl Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

25 / 01 / 2024

Omar KHALIL

TEŞEKKÜR

Tırmanıp ağaçları bulutlara aştık....

Bu cümle zorlukların üstesinden gelme, sınırları aşma veya hayallerin gerçekleşmesi yönünde olumlu bir mesaj taşıyor. Umudumuzu kesmedik, çalışmaya devam ettik ve sonunda muradımıza ulaştık.

Bu çalışmayı yaparken yanımızda olan insanlar çok fazla. Aslında onların motive edici güzel sözleri olmasaydı buraya ulaşamazdık.

Bu çalışmada, bana ilk başladığımdan beri Türkçe öğrenmem konusunda destek ve emek veren hocam Öğr. Gör. Sümeyra Özkan'a özel bir teşekkürüm var. Tez danışmanım Dr. Öğr. Üyesi Diyadin Can'a sadece danışman hocam olduğu için değil büyük destek veren bir arkadaş olduğu için de teşekkür ederim. Ayrıca FEN BİLİMLERİ ENSTİTÜSÜ/ Biyomedikal mühendisliği bölümü yüksek lisans derslerinde kendilerinden çokça istifade ettiğim hocalarıma sonsuz teşekkür ediyorum. Çalışmasını uygulama süresinde yardımların esirgemeyen Diyadin Can, lisanstan tanıdığım arkadaşlarıma da teşekkür ederim.

Hayatım boyunca beni destekleyen, haklarını asla ödemeyeceğimi anneme ve babama teşekkürlerimi sunarım. Aslında bir insan ailenin desteği olmadan da bir şey yapamaz, abim, ablam, eniştem ve yengeme hayatımın son gününe kadar teşekkür etmeye devam edeceğim.

Son olarak ailemden uzak bu dönemde gurbeti kolaylaştıran insanlara saygım, sevgim ve teşekkürlerimi sunarım.

Sevgim ve Saygımla

Omar Khalil

İÇİNDEKİLER

Etik Beyan.....	ii
Teşekkür.....	iii
İÇİNDEKİLER	iv
Sembol ve Kısaltma Listesi	vi
Şekil ve Grafik Listesi	vii
ÖZET	ix
ABSTRACT.....	x
1. GİRİŞ.....	1
2. GENEL BİLGİLER	5
2.1 Beyin Anatomisi.....	5
2.1.1 Beynin Anatomisi	6
2.1.2 Beyin Tümörü	8
2.2 Manyetik Rezonans Görüntüleme.....	10
3. MALZEME VE YÖNTEMLER.....	14
3.1 Yapay Zeka	14
3.1.1 Yapay zeka ve tıp ilişkisi	16
3.2 Görüntü İşleme.....	18
3.2.1 Görüntü İşlemenin Kısa Tarihçesi	19
3.2.2 Görüntü işlemede kullanılan yapay zeka ve derin öğrenme algoritmaları	21
3.2.3 MATLAB ile Görüntü İşleme.....	23
3.2.4 Python ile görüntü işleme	24
4. BULGULAR.....	32
5. TARTIŞMA VE SONUÇ	36
5.1 Sonuç.....	36
5.2 Öneriler.....	36

EKLER.....	37
KAYNAKÇA.....	46
ÖZGEÇMİŞ	47



Sembol ve Kısaltma Listesi

CT veya BT	Bilgisayarlı Tomografi / Computational Tomography
MR, MRI veya NMR	Manyetik Rezonans veya Nükleer Manyetik Rezonans
AI veya YZ	Yapay Zeka / Artificial Intelligence
CNS	Merkezi sinir sistemi
ML	Makine Öğrenimi (Machine Learning)
MATLAB	MATrix LABoratory
DL	Derin Öğrenme (Deep Learning)



Şekil ve Grafik Listesi

Şekil 2-1: İnsan Vücudunun Sinir Sistemi [2].	5
Şekil 2-2: Beynin Parçaları [3]	6
Şekil 2-3: Beycik (Cerebellum) [7]	8
Şekil 2-4: Beyin Sapı [8]	8
Şekil 3-1: Görüntünün yüklenme, döndürülme ve kaydedilmesi	26
Şekil 3-2: Görüntünün yüklenmesi, dönüştürme ve kaydedilmesi	27
Şekil 3-3: Nesnelerin tespiti	28
Şekil 3-4: Nesneleri çerçevele	30
Şekil 3-5: Gri seviyeye dönüş	31
Şekil 4-1: Python ile yapılan analiz.	32
Şekil 4-2: Python ile yapılan analiz.	33
Şekil 4-3: Python ile yapılan Alzheimer analizi.	33
Şekil 4-4: Python ile yapılan ileri seviye Parkinson hastalığı analiz.	34
Şekil 4-5: Python ile yapılan analiz.	34
Şekil 4-6: Python ile yapılan analiz.	35

Tablo Listesi

Tablo 1: Yapay zekânın dilleri ve kütüphaneleri	18
---	----



ÖZET

Yapay Zeka Algoritmaları ile Beyin Tümörlerinin Teşhisi

Omar KHALIL

İstanbul Yeni Yüzyıl Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Biyomedikal Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Diyadin CAN

Ocak 2024, 47 sayfa

Görüntü işleme, çeşitli paket programlar ve/veya yazılım dilleri kullanılarak, genel olarak resimsel bilgilerin analizine ve bunlardan çeşitli sonuçlar elde etmeye yönelik bir yöntem olarak tanımlanabilir. Tezimizde bugüne kadar geliştirilmiş algoritmalar incelenecek ve bunlar geliştirilerek. Çeşitli görüntüleme cihazlarından alınan görselleri işlemek şeklinde olan hedefimize yönelik olarak pratik bir şekilde tümörlü bölgelerin, fikir edinilmesi amacıyla, ön teşhisi yapılması planlanmaktadır.

Tümör metastazının erken teşhisi klinik tedavi için çok önemlidir. Yapay zeka tıp alanında büyük umut vaat etmektedir. Bu nedenle, tezimizde, tıbbi radyoloji görüntüleme tekniklerini kullanarak tümör metastazının ön tespitini otomatik olarak yapmak amacıyla YZ algoritmalarının oluşturma ve kullanmayı amaçladık. Bu nedenle, tezimizde, tıbbi radyoloji görüntüleme tekniklerini kullanarak tümör metastazını tespit etmek amacıyla YZ algoritmalarının işlem tekniklerini, algoritmalarını ve elde ettiğimiz sonuçları sunmayı hedefliyoruz.

Anahtar Kelimeler: Görüntü İşleme, Yapay Zeka.

ABSTRACT

Detection of Brain Tumor Zones via Artificial Intelligence Algorithms

Omar KHALIL

Istanbul Yeni Yuzyil University, Science and Engineering Institute

Master Thesis, Biomedical Engineering Department

Supervisor: Assist. Prof. Dr. Diyadin CAN

January 2024, 47 pages

Image processing can be defined as a method for analyzing pictorial information in general and obtaining various results from it by using various package programs and/or software languages. In our thesis, algorithms that have been developed so far will be examined and developed. For our goal, which is to process images taken from various imaging devices, it is planned to make a preliminary diagnosis of tumor regions in a practical way in order to gain an idea.

Early diagnosis of tumor metastasis is very important for clinical treatment. Artificial intelligence holds great promise in the field of medicine. Therefore, in our thesis, we aimed to create and use AI algorithms to automatically make preliminary detection of tumor metastasis using medical radiology imaging techniques. Therefore, in our thesis, we aim to present the processing techniques, algorithms and results of AI algorithms to detect tumor metastasis using medical radiology imaging techniques.

Keywords: *Image Proseses, Artificial intelligence.*

1. GİRİŞ

Veri teknolojisi ve veri analizi süreçleri insan yaşamının her alanına yayılmış durumdadır. Son zamanlarda bu gelişmelerin sağlık alanındaki etkileri de büyük öneme sahiptir. Tıbbi görüntüler, herhangi bir tıbbi vakanın tıbbi değerlendirmesinin temellerini oluşturur. Bu görüntüler, vücudun biyolojik sistemini daha iyi anlamak için insan vücudunun anatomisini ve fizyolojisini karakterize etmek için kullanılmaktadır. Bu görüntülerden bilgi çıkarabilmek için çeşitli analiz ve değerlendirme teknikleri de geliştirilmektedir.

Normalde radyoloji veya klinik görüntüleme departmanı ile ilgili olan tıbbi görüntüleme, klinik analiz, tıbbi müdahale ve birkaç organ veya dokunun görsel fonksiyonel temsili için vücudun iç ve dış bölümlerinin görsel temsili yapma yoludur. Bilimsel bir araştırma alanı olarak tıbbi görüntüleme, tıbbi teşhisin kalitesini ve doğruluğunu artırmak amacıyla enstrümantasyon, veri toplama ve modelleme alanında araştırma ve geliştirme bağlamında biyomedikal mühendisliğinin, tıbbi fiziğin bir alt disiplinini oluşturur.

Tıbbi görüntüleme alanındaki birçok teknik ve buluş, bilimsel ve endüstriyel uygulamalarda kullanılabilir. Vücudun anatomisini gözlemlemeye yönelik invazif olmayan ve zararsız teknikler tıp alanında da korunmaktadır. Bu görüntülerin sonuçları, görüntü kullanılmadan önce düzgün bir şekilde analiz edilir ve verimli bir şekilde işlenir. Bu analizin uygulanabilir olabilmesi için görüntüden uygun verinin çıkarılmasını kolaylaştıracak bilgisayar yazılımları ve programları yazılmaktadır. Buna, karmaşık görüntü işlemeyi gerçekleştirmek için dijital analiz karmaşık algoritmalarının kullanıldığı Dijital Görüntü İşleme adı verilir.

İnsan vücudunun mekânsal özelliklerinin sınıflandırılması, tespiti ve bölümlendirilmesi, herhangi bir bilgisayarlı görüntünün/görüntünün temelini oluşturur. Dijital Sinyal İşlemenin bir alt alanı olarak görüntü işlemenin, girdi verilerine çok daha geniş bir algoritma aralığının uygulanmasına izin vermesi ve işleme sırasında gürültü oluşumu ve sinyal bozulması gibi sorunları önleyebilmesi açısından birçok avantajı vardır. Ancak insan vücudunun homojen olmaması ve özelliklerinin çeşitliliği nedeniyle görüntülerin analizi biraz karmaşık olabilir ve doğrusal filtreleme, temel bileşen analizi, bağımsız analiz gibi bazı gelişmiş görüntü işleme tekniklerine ek olarak duruma ilişkin belirli bir ön bilgi gerektirebilir. Bileşen analizi, gizli

Markov modelleri, kısmi diferansiyel denklemler, kendi kendini düzenleyen haritalar ve sinir ağları. Tüm bu teknikler, insan vücudundaki anormalliğin tanımlanması için istenen nesnenin segmentlere ayrılmasına olanak tanır. Tespit edilmesi gereken nesnenin belirlenmesi ve segmentlere ayrılabilmesi, diğerlerinden ayrı olarak tek bir küme halinde gruplanabilmesi tıbbi görüntüleme alanında büyük bir adımdır.

Her görüntü, her pikselin bir kümeye ait olduğu piksel bazındadır. Görüntü segmentasyonu tıp alanının hayati bir varlığıdır. Çeşitli tıbbi verileri görselleştirerek ve çeşitli hastalıkların teşhisini kolaylaştırarak tıbbi sistemi destekler. Bu segmentasyon, karmaşık bir görüntüyü gruplayarak, bölerek ve yüksek ve düşük seviyeli görüntüler arasındaki farktan oluşan boşlukları kapatarak görüntüyü anlamsal anlamlı bölümlere ayırarak basitleştirir. Bu bölümlerin toplamı orijinal görüntü olacaktır. Tümörler, kırık kemikler ve diğer birçok hastalık görüntüleme yöntemiyle tespit edilir; bu yöntemde tümörün segmentlere ayrılması, klinik personelin tümörün sınırlarını tespit etmesine ve tümörün yalnızca boyutunu değil aynı zamanda tipini de belirlemesine yardımcı olur. Görüntüleme sırasında hastadan veya çevreden gelen gürültü, görüntünün kalitesini etkileyerek kötü veya zayıf tanıya neden olur. CT taraması, manyetik rezonans görüntüleme, X-Işınları ve benzeri görüntüleme makinelerinin görüntü kalitesini, teşhisini ve performansını artırmak için yıllar boyunca birçok teknik, algoritma ve yaklaşım geliştirilmiş ve uygulanmıştır; Görüntü ölçümleri ile tümör seviyesindeki herhangi bir değişiklik tespit edilmelidir. Bu yöntemler piksel doğruluk düzeyidir. Bu makinelerin elde edilen görüntüsünün filtrelenmesi ve bölümlere ayrılması, birçok hastanın hayatta kalma oranının artmasına olanak tanır; özellikle de belirli bir olayın gerçek nedeninin, bariz neden olması gerekmez. İnsan vücudunun en ölümcül nedenlerinden biri olan felç, kalbin fonksiyon bozukluğunu gösterir ancak bazen kafadaki bir tümör de felçle aynı belirtileri gösterebilir ve tıbbi kararlarda kafa karışıklığına neden olabilir. Bu tümörün erken evrelerde tespit edilmesi hayatta kalma yüzdesini artırır.

İnsan vücudunun homojen olmaması ve özelliklerinin çeşitliliği nedeniyle bazı segmentasyon teknikleri kullanılamamaktadır. Bu nedenle segmentasyon doğruluğu ve izlenmesi, tümör veya segmente edilmek istenen nesne etkilenir. Segmentasyon, 1970'lerden beri her vücut tipinde kullanılabilecek sağlam, eksiksiz yeni bir yöntemin bulunmadığı eski bir araştırma konusudur.

Kısıtlamalar ve Sınırlamalar:

Bu projeyi gerekleřtirme yolculuęu sırasında birok kısıtlama ve sınırlama ortaya ıktı:

1. Ekonomik Kısıtlamalar: Bu projenin tasarımı, zerinde MATLAB Yazılımının kurulu olduęu bir dizüstü bilgisayar gerektiriyordu.
2. evresel Kısıtlamalar: Bu tasarım zellikle dizüstü bilgisayarda alıřtırılan bir yazılım olduęu iin evreye zarar vermez. Hem evreyi hem de hastanın saęlığını etkilemez.
3. Saęlık ve Gvenlik Kısıtlamaları: Bu kısıtlamalar, bu tasarımın ne kadar gvenli olduęu ve hastanın gvenliğini saęladığı ile ele alınmaktadır. Tasarım daha nce de belirttiğimiz gibi hasta ile doęrudan teması olmayan bir yazılım kodu olduęundan hastanın saęlığına ve gvenliğine herhangi bir zararı bulunmamaktadır.
4. retilbilirlik Kısıtlamaları: Bu tasarımın bir dięer yn de herhangi bir imalat gerektirmemesidir.
5. Sosyal Kısıtlamalar: Bu kısıtlama, doktorun insan vcudundaki anormallięi tespit etme becerisini geliřtirme becerisini ele alır. Teknisyen, beyin tmrnn segmentasyonunun doęruluęunu artırarak tedavinin tmrn boyutuna gre etkinliğini izleyebilecektir.
6. Etik Kısıtlamalar: Tasarımın etik kısıtlamaları ortak deęerler gz nnde bulundurulurak dikkate alınmıştır. Algoritma yalnızca tmr hcrelerini normal hcrelerden kmeleme ve segmentlere ayırma grevlerini yerine getirir.

Tez alıřmamız ařağıdaki řekilde drt blm halinde dzenlenmiştir:

1. Tez, tezin kısaca anlatıldığı, gelecek veriler ve kısıtların anlatıldığı bir giriř ile bařlamaktadır.
2. Giriřin ardından literatr taraması, beyin anatomisi ve beyin tmr trlerinin yanı sıra segmentasyon alanında daha nce yapılmıř alıřmaların sunulmasına olanak saęlayacaktır.
3. Literatr taramasının ardından uygulanan yntem, yazılım tasarımının her adımının zellikleriyle birlikte tanıtılacaktır
4. Bulgular kısmında, yazılım tasarımının sonuları ve ıktıları verilip uygun řekilde tanıtımı yapılacaktır.

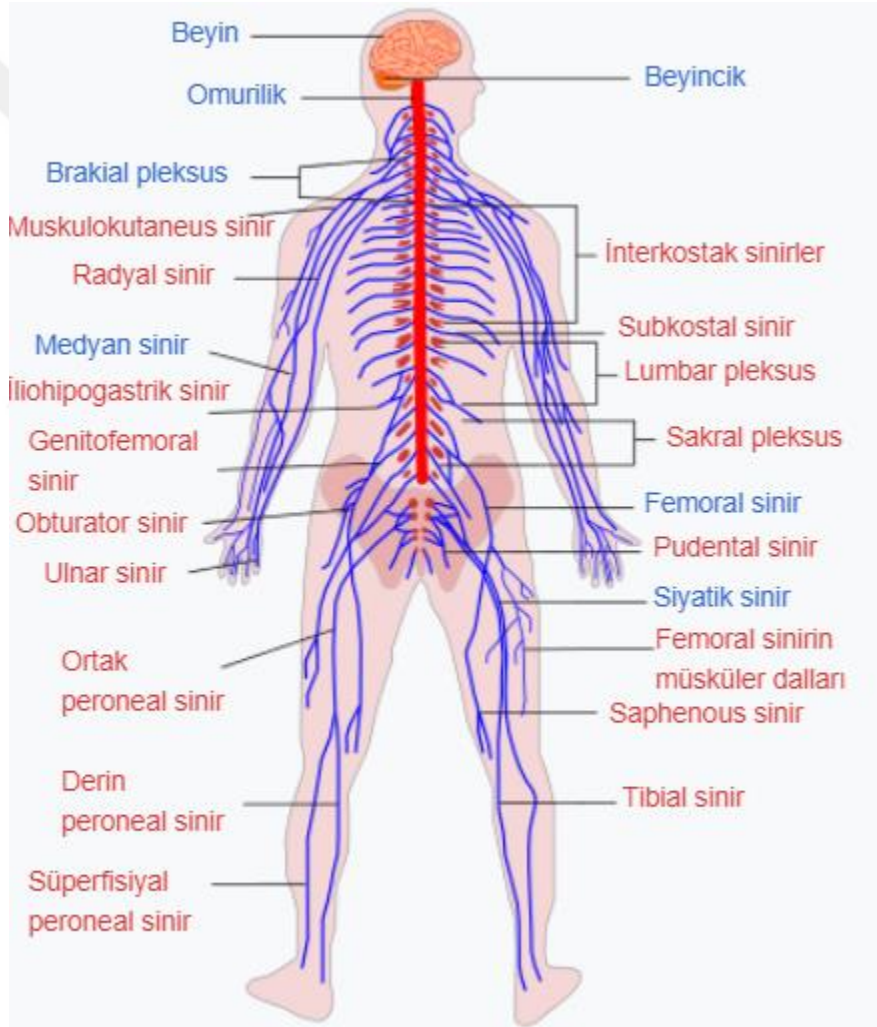
5. Son olarak sonuç bölümünde bu tasarımla ilgili gelecekte yapılabilecek olası çalışmalar ve tüm tezin bir özeti sunulacaktır.



2. GENEL BİLGİLER

2.1 Beyin Anatomisi

Sinirbilim, bilgiyi vücudun bir kısmından diğerine ileterek vücudun davranışını, hareketlerini ve duyuşal bilgilerini kontrol eden insan vücudundaki son derece karmaşık sistemin bilimsel olarak incelenmesidir. İnsan vücuduna yayılmış sensörler sayesinde sinir sistemi, vücudu etkileyen her türlü çevresel değişikliği algılayabilir, verileri CNS'ye göndererek bilgiyi analiz edebilir ve sonunda başka bir sistemin, yani endokrin sistemin savunmasına veya tepki vermesine olanak tanır. Motor fonksiyonu oluşturan sinir sistemi tarafından işlenen bilgilerdir [1].



Şekil 2-1: İnsan Vücudunun Sinir Sistemi [2].

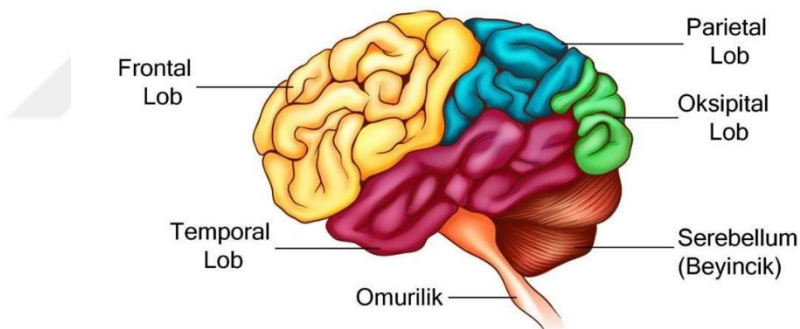
Omurgalıların sinir sistemi iki bölümden oluşur: Merkezi sinir sistemi (CNS) ve Periferik sinir sistemi (PNS). CNS beyin ve omurilikten oluşurken, PNS uzun liflerin veya aksonların kapalı

demetlerini oluşturan sinirlerden oluşur. Bu aksonlar CNS'yi vücudun her organına bağlar. Bu nedenle bu bölümde beyin anatomisi ve karmaşıklığı görüntüleme düzlemleri ve analiz prosedürü ile açıklanacaktır.

2.1.1 Beynin Anatomisi

İnsan vücudundaki en karmaşık organlardan biri beyindir. İnsan bilincinin mekânı, iradi hareketin koordinatörü ve bilinç dışı süreçlerin düzenleyicisi olarak her şey gerçekleştirilebilir. Beynin karmaşıklığına rağmen hakkında bilgi edinilmesi ilginç bir organdır.

İnsan vücudundaki sinir sisteminin bir parçası olan beyin, her biri vücudun belirli işlevlerinden sorumlu olan loblara bölünmüştür. Beyin, fiziksel işlevselliğin yanı sıra duygular, zeka, hafıza gibi birçok özellikten de sorumludur. Yapısal olarak beyin, kafatası tarafından korunan beyin sapı, beyincik ve beyin olmak üzere üç ana yapıya ayrılabilir.



Şekil 2-2: Beynin Parçaları [3]

2.1.1.1 Beyin

Beynin en büyük kısmı sol ve sağ yarım küreden oluşur. Duyusal sistemin daha yüksek fonksiyonunu korur. Sağ yarıküre vücudun sol tarafından, sağ yarıküre ise vücudun sol tarafından sorumludur [4]. Cerebrum beyaz ve gri maddeden oluşur ve dört farklı loba ayrılır: frontal, parietal, oksipital ve temporal loblar.

- Frontal lob:

Bu Lob beynin daha büyük bir kısmını kaplar ve başın ön kısmında bulunur. Kişilik özelliklerini, karar vermeyi ve hareketi içerir. Beynin motor bölümünü oluşturur. Frontal lobda Broca bölgesi olarak bilinen ve konuşma yeteneğiyle ilgili özel bir alan bulunur.

Broca bölgesinin hasar görmesi durumunda kişi artık iletişim kuramayacak, konuşarak kendini ifade edemeyecektir. Ancak konuşulan dili anlamaya ve metni okumaya dokunulmaz [4].

- Parietal Lob:

Beynin orta kısımları dokunma, ağrı ve sıcaklık algılamasından sorumludur [4]. Nesnelerin tanımlanmasına ve özel ilişkilerin anlaşılmasına ek olarak [5].

- Oksipital lob:

Başın arka kısmında bulunur ve görmeden sorumludur [5].

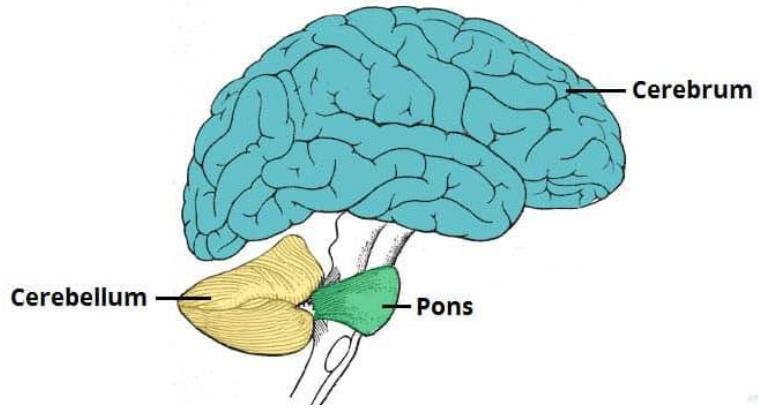
- Temporal lob:

Beynin yan tarafında yer alan, kısa süreli hafızadan sorumlu olan işitmedir [6]. Temporal lobda dilin anlaşılmasından sorumlu olan Wernicke alanı adı verilen özel bir alan bulunmaktadır [5].

Sol temporal lobda yer alan Wernicke alanı bireyin ifadelerinin anlam kazanmasını sağlar. Bu bölgenin hasar görmesi Wernicke afazisine neden olur. Bu durumda hasta, anlamı olmayan uzun cümleler, gereksiz kelimeler veya yeni oluşturulmuş kelimeler konuşabilir [4].

2.1.1.2 Beyincik

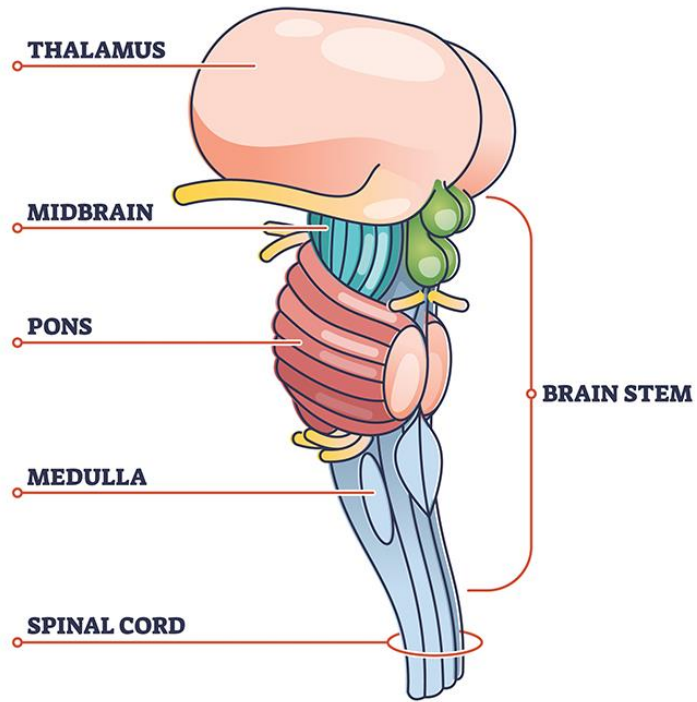
Başın arka kısmında, elin yumruk büyüklüğünde, beynin şakak ve oksipital lobunun altında ve beyin sapının üzerinde yer alan beyincik, korteks gibi iki loba sahiptir. Serebellumun dış kısmı nöronları içerirken, serebellumun iç kısmı serebral korteks ile iletişim kurar. Görevi istemli kas hareketlerini koordine etmek ve postürü, dengeyi ve dengeyi sağlamaktır [5].



Şekil 2-3: Beycik (Cerebellum) [7]

2.1.1.3 Beyin Sapı

Beyin sapı, beyni omuriliğe bağlar [5]. Pons, medulla ve orta beyinden oluşur.



Şekil 2-4: Beyin Sapı [8]

2.1.2 Beyin Tümörü

Her organ gibi beyin de bir hastalık ya da tümör nedeniyle tehlikeye girmeye mahkumdur. Birçok kuruluş bu konuyla ilgili farkındalık geliştirmeye başladı. İntrakraniyal tümör olarak bilinen beyin tümörü, öngörülemeyen ve kontrol edilemeyen bir şekilde gelişmeye ve

büyümeye başlayan, istenmeyen bir kitledir. Nörolojik cerrahlar derneğine göre, birincil ve metastatik olmak üzere iki kategoriye ayrılabilir yüz elliden fazla beyin tümörü türü mevcuttur [9]. Birincil beyin tümörü, beyin dokularından veya beyin çevresinden kaynaklanan bir tümör grubudur; birincil beyin tümörleri Glial veya Glial olmayan ve iyi huylu veya kötü huylu tümörler olarak kabul edilirken, metastatik beyin tümörleri göç eden bir tümör grubudur. Bu grup beyin tümörleri vücudun farklı bir bölgesinde başlar ve beyne göç eder, kan dolaşımı yoluyla göç eder ve kanserli ve kötü huylu olarak kabul edilir [9.].

2.1.2.1 Tümör Tipleri

DSÖ Düşük Derece WHO Derece I

- En az kötü huylu (iyi huylu)
- Muhtemelen yalnızca ameliyatla tedavi edilebilir
- Süzücü değil
- Uzun vadede hayatta kalma
- Yavaş büyüyor
- Pilositik astrositom
- Kraniyofarenjiyom
- Gangliositoma
- Ganglioglioma

DSÖ Derece II

- Nispeten yavaş büyüyor
- Biraz sızıcı
- Daha yüksek dereceli olarak tekrarlayabilir
- "Diffüz" Astrositom
- Pineositoma
- Saf oligodendroglioma

Yüksek Dereceli WHO Derece III

- Malign
- Sızıcı
- Daha yüksek dereceli olarak tekrarlama eğilimi gösterir

- Anaplastik astrositom
- Anaplastik ependimom
- Anaplastik oligodendroglioma

DSÖ Derece IV

- En kötü huylu
- Baskın büyüme, agresif
- Yaygın olarak infiltre edici
- Hızlı tekrarlama
- Nekroz eğilimli
- Glioblastoma multiforme (GBM)
- Pineoblastom
- Medulloblastom
- Ependimoblastom

2.2 Manyetik Rezonans Görüntüleme

Manyetik Rezonans Görüntüleme (MRI), iyonlaştırıcı radyasyon kullanmadan insan vücudunun içinin ayrıntılı görüntülerini sağlayan güçlü bir tıbbi görüntüleme tekniğidir. Bir manyetik alan içindeki belirli atom çekirdekleri elektromanyetik radyasyonu emip yaydığında ortaya çıkan bir fenomen olan nükleer manyetik rezonans (NMR) ilkelerine dayanır. MRG, nörolojik, kas-iskelet sistemi ve kardiyovasküler bozukluklar dahil olmak üzere çok çeşitli tıbbi durumların teşhisinde kullanılır.

MRI ayrıntılı olarak şu şekilde çalışır:

1. NMR'nin Temel İlkeleri:

MRG'nin kalbinde nükleer manyetik rezonans ilkesi yatar. Güçlü bir manyetik alana yerleştirildiğinde, hidrojen çekirdekleri (protonlar) gibi bazı atom çekirdekleri kendilerini manyetik alanla hizalar. Ancak, hizalanmaları mükemmel değildir; ayrıca manyetik alanın eksenini etrafında dönen topaçlar gibi hareket ederler.

2. Radyofrekans (RF) Nabız ve Uyarma:

Bir MRI görüntüsü oluşturmak için ilk adım, hizalanmış çekirdeklerin dengesini geçici olarak bozmaktır. Bu, manyetik alana dik olarak uygulanan bir elektromanyetik darbe olan bir radyo frekansı (RF) darbesi kullanılarak elde edilir. RF darbesi, protonlara enerji sağlayarak onların düşük enerjili durumlarından daha yüksek enerjili bir duruma geçmelerine neden olur. Bu süreç uyarılma olarak bilinir.

3. Gevşeme:

RF darbesi kapatıldıktan sonra, uyarılmış protonlar daha düşük enerji durumlarına geri dönmeye başlarlar. Bu gevşeme sırasında iki önemli özellik ölçülür: T1 (uzunlamasına gevşeme süresi) ve T2 (enine gevşeme süresi).

T1 Gevşeme:

Bu süreç, protonların ana manyetik alanla denge hizalanmalarına geri dönmesini içerir. T1 gevşemesi doku tipine göre değişir ve doku kompozisyonu hakkında bilgi verir. Farklı dokuların farklı T1 gevşeme süreleri vardır ve bu da son görüntüde zıt sinyallere yol açar.

T2 Gevşemesi:

T2 gevşemesi, uyarılmış protonların komşu protonlarla etkileşimleri nedeniyle faz tutarlılıklarını kaybetmesiyle gerçekleşir. Bu faz tutarlılığı kaybı, enine mıknatıslanmada bir azalmaya neden olur. T2 gevşeme süreleri de dokular arasında farklılık gösterir ve görüntü kontrastına katkıda bulunur.

4. Sinyal Tespiti:

Gevşeme işlemi sırasında, protonlar elektromanyetik sinyaller şeklinde enerji yayarlar. Bu sinyaller, MRI makinesinin, görüntülenen vücut kısmının etrafına stratejik olarak yerleştirilmiş alıcı bobinleri tarafından algılanır. Sinyaller son derece zayıftır ve amplifikasyon ve işleme gerektirir.

5. Uzamsal Kodlama:

Bir görüntü oluşturmak için MRI makinesinin vücudun farklı yerlerinden gelen sinyalleri ayırt etmesi gerekir. Uzamsal kodlama, manyetik alan gradyanları kullanılarak elde

edilir. Bu gradyanlar, ana manyetik alanın üzerine bindirilerek, protonların konumlarına bağı olarak biraz farklı frekanslarda rezonansa girmesine neden olur. Bu gradyanların gücünü ve zamanlamasını değiştirerek, MRI makinesi uzamsal bilgileri kodlayabilir.

6. Görüntü Yeniden Oluşturma:

Tespit edilen sinyaller, uyarılmış protonların hem uzamsal hem de frekans bilgilerini içerir. Bu verileri ayrıntılı bir görüntüye dönüştürmek için Fourier dönüşümü adı verilen matematiksel bir işlem kullanılır. Fourier dönüşümü, hidrojen çekirdeklerinin dağılımının uzamsal bir haritasını oluşturmak için sinyallerin frekans bileşenlerini analiz eder.

7. Görüntü Kontrastı:

MRG'deki görüntü kontrastı, farklı dokuların gevşeme süreleri (T1 ve T2) ile belirlenir. T1 gevşeme süresi daha uzun olan dokular T1 ağırlıklı görüntülerde daha parlak görünürken, T2 gevşeme süresi daha uzun olan dokular T2 ağırlıklı görüntülerde daha parlak görünür. Radyologlar, RF darbelerinin ve yankı sinyallerinin zamanlamasını ayarlayarak, belirli doku özelliklerini vurgulamak için MRI görüntülerindeki kontrastı manipüle edebilir.

8. MRI Dizisi Türleri:

Spesifik doku özelliklerini veya özelliklerini vurgulamak için farklı MRI dizileri kullanılır. Bazı yaygın diziler şunları içerir:

T1-Ağırlıklı: Anatomik detayları görselleştirmek ve farklı T1 gevşeme sürelerine sahip dokuları ayırt etmek için en iyisidir.

T2 Ağırlıklı: T2 gevşeme sürelerindeki farklılıklar nedeniyle farklı yumuşak dokular arasında mükemmel kontrast sağlar.

Proton Yoğunluk Ağırlıklı: Proton dağılımını vurgular ve dengeli doku kontrastı sağlar.

Sıvı-Atenuated Inversion Recovery (FLAIR): Beyin omurilik sıvısından gelen sinyalleri baskılayarak beyin lezyonlarının görünürlüğünü artırır.



3. MALZEME VE YÖNTEMLER

3.1 Yapay Zeka

Yapay zekâ, bilgisayarların insanlardan daha iyi bir şekilde düşünebilme ve problem çözebilme yeteneklerine sahip olabilmelerini sağlamaya yönelik bir teknolojidir. Bu yetenekler sayesinde bilgisayarlar insanlar gibi düşünebilir, konuşabilir ve çeşitli görevleri yerine getirebilir. Yapay zekâ, insanların gerçekleştiremediği veya zorluk çektiği birçok görevi kolaylıkla yerine getirebilir ve bu sayede insanların işlerini kolaylaştırabilir. Özet olarak, yapay zekâ insanlardan daha iyi bir şekilde düşünebilen ve problem çözebilen bilgisayarlar yaratmaya yönelik bir teknolojidir. Örneğin, yapay zekâ sayesinde bir bilgisayar müzik dinleyebilir, resimleri tanıyabilir veya insanların konuşmalarını anlayabilir. Bu teknolojinin kullanım alanları oldukça geniştir ve bu alanlar gün geçtikçe daha da genişlemektedir. Yapay zekâ, insanların hayatını kolaylaştıran ve onların işlerini daha verimli hale getiren bir teknolojidir.

1940 ile 1960 arasındaki dönem, teknolojik gelişmelerin (İkinci Dünya Savaşı'nın bir hızlandırıcı olarak rol oynadığı) ve makinelerin işleyişini organik varlıkların işleyişiyle nasıl bir araya getirebileceğini anlama isteğinin birleşimiyle güçlü bir şekilde belirlendi. Norbert Wiener, kibernetikte öncü bir isim olarak matematiksel teoriyi, elektroniği ve otomasyonu "hayvanlarda ve makinelerde kontrol ve iletişimin bütünlük bir teorisi" olarak birleştirmeyi amaçladı. Bu süreçten hemen önce, 1943 yılında Warren McCulloch ve Walter Pitts, biyolojik nöronun (formal nöron) ilk matematiksel ve bilgisayar modelini geliştirmişlerdi

1950'lerin başlarında, John Von Neumann ve Alan Turing, yapay zeka terimini ortaya çıkarmamış olabilirler, ancak onlar arkasındaki teknolojinin kurucu babalarıydı: Bilgisayarları 19. yüzyıl ondalık mantığından (bu, 0 ila 9 arasındaki değerlerle uğraşır) ikilik mantığına (Boolean cebiri temelinde çalışan ve 0 veya 1'in zincirlerine dayanan mantık) geçirdiler. Bu geçiş, çağdaş bilgisayarların mimarisini şekillendirmelerine ve programlanmış görevleri gerçekleştirebilen evrensel makineler olduğunu göstermelerine olanak sağladı. Turing ise 1950 yılında ünlü "Bilgisayar Makineleri ve Zeka" makalesinde makinenin olası zekâsını ilk kez sorguladı. Bu makalesinde "taklit oyunu" kavramını tanıttı: Bir insan hakem, teletip ile hem bir insanla hem de bir makineyle etkileşime giriyor ve hangisinin insan hangisinin makine olduğunu anlamaya çalışıyordu. Bu "Turing testinin geçerliliği üzerine yapılan tartışmalara

rağmen, insan ve makine arasındaki sınırı sorgulamanın başlangıcına kaynaklık ettiği sıkça söylenir.

"Yapay Zeka" terimi, John McCarthy'a atfedilebilir. Marvin Minsky, Yapay zekâyı, "şu anda insanlar tarafından daha iyi gerçekleştirilen görevleri yerine getiren bilgisayar programlarının geliştirilmesi" olarak tanımlar; bu görevler yüksek düzeyli bilişsel süreçleri içerir, örneğin algı, bellek düzenleme ve eleştirel akıl yürütme. Yapay zeka disiplininin kurucusu olarak kabul edilen an, 1956 yazında Rockefeller Enstitüsü tarafından finanse edilen Dartmouth College konferansıdır. İlginç bir şekilde, bir konferans olarak başlamamış, daha çok bir atölye şeklinde ilerlemişti. Sadece McCarthy ve Minsky gibi altı kişi, bu çalışmanın tamamında sürekli olarak yer almıştı; bu çalışma, esasen formal mantığa dayalı gelişmelere odaklanmıştı.

Yapay zekâ kavramı 1950'lerin ortalarında ortaya çıkmıştır. Bu dönemde bilgisayarların işlem yetenekleri hala oldukça sınırlıydı ve bu nedenle yapay zekâ kavramının gerçekleştirilmesi de oldukça zordu. Ancak ilerleyen yıllarda bilgisayarların işlem yetenekleri hızla gelişmeye başladı ve bu sayede yapay zekâ teknolojisi de ilerleme kaydetmeye başladı. 1960'larda yapay zekâ, ilk kez gerçek bir problemi çözmeyi başardı ve bu da yapay zekâ teknolojisinin gelişimine büyük bir ivme kazandırdı. 1970'lerde ise yapay zekâ teknolojisi, birçok alanda kullanılmaya başlandı ve bu sayede yapay zekâ, insanların hayatını kolaylaştıran ve onların işlerini daha verimli hale getiren bir teknoloji haline geldi. Günümüzde ise yapay zekâ teknolojisi hızla gelişmekte ve bu teknolojinin kullanım alanları da gün geçtikçe genişlemektedir.

Yapay zekâ teknolojisi, günümüzde birçok alanda kullanılmaktadır. Örneğin, yapay zekâ sayesinde bir bilgisayar müzik dinleyebilir, resimleri tanıyabilir veya insanların konuşmalarını anlayabilir. Ayrıca yapay zekâ, sağlık sektöründe hastalıkların tanısında ve tedavisinde kullanılmaktadır. Yapay zekâ ayrıca otomobil ve uzay teknolojisinde de kullanılmaktadır. Örneğin, yapay zekâ sayesinde bir otomobil kendi kendine sürüşe veya bir uzay aracının gezegenler arasında seyahat edebilmesine olanak sağlamaktadır. Yapay zekâ teknolojisi, birçok alanda insanların işlerini kolaylaştırmakta ve onların hayatını kolaylaştırmaktadır. Bu teknolojinin kullanım alanları da gün geçtikçe daha da genişlemektedir.

Yapay zekâ uygulamaları genellikle açık kaynak ekosisteminde sürdürülmektedir. Bundan dolayı birçok programlama dili, kütüphane ve uygulama ortamları mevcuttur [10].

3.1.1 Yapay zeka ve tıp ilişkisi

En son teknolojilerin yakınsaması, özellikle de makine öğrenimi (ML) tekniklerinin ilerlemesi, tıpta dönüştürücü bir çağa öncülük ediyor. Bu evrim, geleneksel istatistiksel yaklaşımların sınırlamalarını aşma ve eyleme dönüştürülebilir içgörüler için karmaşık ve çok yönlü tıbbi verilerin kodunu çözme potansiyelinin kilidini açma vaadini taşıyor. Tıbbi teşhis alanında, yapay zekâ ve tıbbın birleşimi ivme kazanıyor ve bu hiçbir yerde MRI teşhisi alanında olduğu kadar belirgin değil. Sağlık hizmetleri dijital dönüşümü benimsedikçe, MRI gibi tıbbi görüntüleme yöntemlerinden elde edilen bol miktarda veri, yapay zekâ uygulamalarının odak noktası haline geliyor. Yapay Zekanın (AI), özellikle ML'nin, öngörücü radyoloji alanında önemli bir rol oynamak için devreye girdiği yer burasıdır. Hem klinik teşhisleri hem de operasyonel kolaylaştırmayı kapsayan tıbbi alandaki yapay zeka potansiyelini ortaya koymaktadır.

Düzenlenmiş süreç, özellikle MRI taramaları bağlamında kapsamlı tıbbi verilerin işlenmesi ve yönetilmesi için özel olarak tasarlanmış karmaşık bir çerçevenin oluşturulmasıyla başlar. Daha sonra, karmaşık kalıpları, ince sapmaları ve gizli içgörülerini ortaya çıkarmak için bu veri hazinelerinin derinliklerine inerek en ileri analitik teknikler devreye sokulur. Sonuç olarak, ısmarlama yazılım araçları titizlikle hazırlanmış olup, geniş yelpazedeki sağlık hizmeti uygulayıcılarına artırılmış klinik karar destek sisteminden yararlanma olanağı sağlanmaktadır. Sistemin performansının titiz bir şekilde değerlendirilmesi son aşamayı işaret eder. Bu bütünsel yaklaşım, yalnızca tıp uzmanlarına karmaşık MRI verilerini çözme yetkisi vermekle kalmıyor, aynı zamanda hastaların görüntüleme verilerindeki potansiyel anormallikleri veya komplikasyonları öngörebilen öngörücü modellerin geliştirilmesine de zemin hazırlıyor. Tıpkı gerçek zamanlı risk modellerinin cerrahi bakımı dönüştürdüğü gibi, tahmine dayalı yapay zekâ modellerinin MRI teşhisine entegrasyonu, sorunları ortaya çıkmadan önce önleyici olarak tanımlama vaadini taşıyor.

Meyer ve arkadaşlarının çığır açan çalışması. Derin öğrenmeden doğan tahmine dayalı modellerin gücünün bir kanıtıdır. Çabaları, yapay zekâ destekli modellerin MRI taramalarını nasıl titizlikle analiz edebildiğini ve insan gözleminden kaçabilecek ciddi anormalliklerin ince göstergelerini nasıl ortaya çıkarabildiğini ortaya çıkardı. Bu, ML'nin radyolojik yorumlamada paha biçilmez rehberlik sağlama potansiyelinin altını çizmektedir. Önemli ilerlemeler

kaydedilmiş olsa da yapay zeka destekli tahmine dayalı modellerin MRI teşhis iş akışına tam entegrasyonu, kısmen tıbbi görüntülemenin karmaşık doğası ve özel uzmanlık ihtiyacı nedeniyle devam eden bir çaba olmaya devam ediyor. Bununla birlikte, potansiyel etki derindir. Yapay zekanın incelikli analizine dayalı gerçek zamanlı uyarılar üretme yeteneği, radyologların karmaşık vakalara yaklaşımını yeniden şekillendirebilir, potansiyel olarak teşhis doğruluğunu artırabilir ve sonuçta hasta sonuçlarını iyileştirebilir.

MRI teşhisi alanında, özellikle görüntü alma sonrası aşamada yapay zeka, dönüştürücü bir rol oynamaya hazırlanıyor. MRI taramalarının ardından tıbbi veriler, AI algoritmalarının görüntüleri titizlikle incelediği, altta yatan sağlık koşullarını gösterebilecek kalıpları ve sapmaları tanımladığı analiz aşamasına girer. Yapay zekâ destekli bu analiz, radyologların yeteneklerini artırarak sağlık sorunlarına işaret edebilecek incelikli göstergeleri tespit etmelerine olanak tanıyor. Dahası, yapay zekanın öngörü yetenekleri, yalnızca anormallik tespitinin ötesine geçerek, yaklaşan komplikasyonlara ilişkin, görüntülerde ortaya çıkmadan önce potansiyel olarak içgörüler sağlıyor.

Sonuç olarak, yapay zekâ ve tıbbin kesişmesi, tıbbi teşhiste bir paradigma değişiminin önünü açıyor; yapay zekâ odaklı tahmine dayalı modeller, teşhis hassasiyetini artırma, operasyonel verimliliği optimize etme ve hasta koşullarına ilişkin erken bilgiler sunma potansiyeline sahip. Sorunsuz entegrasyona doğru yolculuk devam ederken, yapay zekâ destekli MRI teşhislerinin geleceği parlak bir şekilde parlıyor ve kişiselleştirilmiş ve proaktif hasta bakımında yeni bir çağın sinyalini veriyor. Yapay zekanın tıbbi uygulamalar üzerindeki dönüştürücü etkisine tanık olmaya devam ettikçe, yapay zekâ ile tıp arasındaki simbiyotik ilişki, hasta sonuçlarını ve tıbbi ortamları yeniden şekillendirmeye hazırlanıyor [11].

Tablo 1: Yapay zekânın dilleri ve kütüphaneleri

Programlama Dilleri	Kütüphaneler
Python	TensorFlow
R	PyTorch
C++	Knet
Java	Apache Spark
Lisp	Scikit-learn
Prolog	PyCaret
Julia	OpenCV
C	H2O vb.
	Keras

Tüm dünya ülkeleri, yapay zekâ ve ilişkili hususlarda kendi ulusal politika ve stratejilerini oluşturmaktadır. Ayrıca, kanun yapıcı mekanizmalarla mevzuata uygun düzenlemeleri de yürütmektedirler. Bu politikalar, ülkelerin oluşturdukları stratejiler doğrultusunda güncellenmekte ve takibi yapılmaktadır.

3.2 Görüntü İşleme

Görüntü işleme, görüntülerin özelliklerini belirleme, düzenleme veya değiştirme işlemidir. Bu işlemler, görüntülerin işlenmesi veya özel amaçlar için kullanılması gerektiğinde yapılır. Görüntü işleme, birçok alanda kullanılır ve çeşitli amaçlar için yapılır. Örneğin, görüntü işleme kullanılarak görüntülerin içerisindeki nesneler tanımlanabilir, görüntüler arasında benzerlik aranabilir ve görüntüler sınıflandırılabilir.

Görüntü işleme, bilgisayar programları ve algoritmaları kullanılarak yapılır. Bu programlar ve algoritmalar, görüntüler üzerinde istenen işlemleri gerçekleştirirler. Örneğin, bir görüntü üzerinde resim düzenleme işlemleri yapmak için resim düzenleme programları kullanılabilir veya bir görüntü üzerinde nesnelerin tespiti için nesne tanıma algoritmaları kullanılabilir.

Görüntü işleme, birçok farklı dil ve kütüphane kullanılarak yapılabilir. Örneğin, Python dilinde NumPy, OpenCV, Pillow ve Scikit-image gibi kütüphaneler kullanılarak görüntü işleme işlemleri yapılabilir. Bu kütüphaneler, görüntülerin yüklenmesi, düzenlenmesi ve işlenmesi gibi işlemler için kullanılabilir. Ayrıca, C++ ve Java gibi diller de görüntü işleme için kullanılabilir.

Görüntü işleme, bir görüntüyü inceleyip, değiştirme, işleme veya yorumlama işlemlerini içeren bir bilim dalıdır. Bu işlemler, birçok alanda kullanılabilir. Örnekler:

1. Medikal görüntüleme: Röntgen, tomografi, MR gibi görüntüleme teknikleri kullanılarak hastalıkların tespiti ve tedavi planlarının yapılmasında görüntü işleme yöntemleri kullanılır.
2. Bilgisayar görüşü: Bilgisayarların dünyayı nasıl algıladığını anlamaya yönelik araştırmalar yapılırken, görüntü işleme yöntemleri kullanılır. Bu alanda, örneğin, nesnelerin tanınması, yol takip etme gibi problemler çözülür.
3. Çevre bilimleri: Örneğin, hava fotoğraflarının incelenmesi ve coğrafi bilgi sistemleri (CBS) gibi uygulamalar, görüntü işleme tekniklerini kullanır.
4. Endüstriyel otomasyon: Örneğin, bir fabrikada üretim hatlarının otomatikleştirilmesi için, üretim sırasında üretilen ürünlerin tanınması ve sayımı için görüntü işleme yöntemleri kullanılır.
5. Bilim ve mühendislik: Örneğin, malzeme mikroskobik görüntülerinin incelenmesi ve malzeme özelliklerinin belirlenmesi için görüntü işleme yöntemleri kullanılır.

Bu örnekler sadece birkaç örnektir, görüntü işleme alanı daha geniş bir yelpazede kullanılabilir.

3.2.1 Görüntü İşlemenin Kısa Tarihçesi

Görüntü işleme, insanların ilk kez görüntülerini inceleyip değiştirme işlemini yapmaya çalıştıkları zamandan beri var olmuş bir alandır. Ancak, görüntü işleme tekniklerinin bugünkü anlamda kullanılmaya başlaması, bilgisayarların gelişmesiyle birlikte olmuştur.

Görüntü işleme tarihi, çeşitli aşamalar ve gelişmelerle doludur. Bu gelişmelerin bazıları şunlardır:

1. 1800'ler: İlk kez, fotoğraf makineleri icat edildi. Bu makineler, ışık etkisiyle çekilen fotoğrafları üretebiliyordu. Bu fotoğraflar, ilk kez görüntülerin çekilip saklanması ve kopyalanması işlemini gerçekleştiriyordu.
2. 1900'ler: Görüntü işleme teknikleri, ilk kez elektronik cihazların kullanılmasıyla gelişmeye başladı. Örneğin, CRT (katot ışın tüpleri / katod ray tube) ekranları, görüntülerin ekrana yansıtılmasını sağlıyordu.

3. 1950'ler: İlk kez, elektronik cihazlarla görüntülerin işlenmesi işlemi yapıldı. Örneğin, Harvard Tavır Analizi Projesi, görüntülerin işlenmesi işlemini ilk kez gerçekleştirdi. Bu proje, görüntülerin elektronik cihazlar aracılığıyla taranması ve önceden tanımlanmış özelliklerine göre sınıflandırılması işlemini gerçekleştirdi.
4. 1960'lar: İlk kez, bilgisayarlar kullanılarak görüntülerin işlenmesi işlemi yapıldı. Örneğin, MIT'in Lincoln Laboratuvarı, ilk kez bilgisayarlar aracılığıyla görüntülerin işlenmesi işlemini gerçekleştirdi. Bu çalışmalar, görüntülerin elektronik cihazlar aracılığıyla taranması ve bilgisayarlar aracılığıyla işlenmesi işlemini gerçekleştirdi.
5. 1970'ler: Görüntü işleme teknikleri, bilgisayarların hızının artmasıyla birlikte hızla gelişmeye başladı. Bu dönemde, ilk kez görüntülerin işlenmesi işlemini gerçekleştiren yazılımlar geliştirildi. Örneğin, Adobe Photoshop gibi yazılımlar, görüntülerin işlenmesi işlemini kolaylaştıran araçları içeriyordu.
6. 1980'ler: İlk kez, görüntülerin işlenmesi işlemini gerçekleştiren cihazların satışı yapıldı. Örneğin, Apple Macintosh bilgisayarları, görüntülerin işlenmesi işlemini gerçekleştiren yazılımları içeriyordu.
7. 1990'lar: İnternetin yaygınlaşmasıyla birlikte, görüntülerin dünya üzerinde hızla paylaşılması mümkün hale geldi. Bu dönemde, görüntülerin işlenmesi işlemini gerçekleştiren yazılımların sayısı hızla artmaya başladı.
8. 2000'ler: İlk kez, görüntülerin işlenmesi işlemini gerçekleştiren yazılımların çevrimiçi sürümleri piyasaya sürüldü. Bu sayede, görüntülerin işlenmesi işlemini gerçekleştiren yazılımlar, dünya üzerinde herhangi bir yerden erişilebilir hale geldi.
9. 2010'lar: İlk kez, görüntülerin işlenmesi işlemini gerçekleştiren yazılımların mobil cihazlar için uyarlanmış sürümleri piyasaya sürüldü. Bu sayede, görüntülerin işlenmesi işlemini gerçekleştiren yazılımlar, herhangi bir yerde ve herhangi bir cihazdan erişilebilir hale geldi.

Bu örnekler, görüntü işleme tarihine ilişkin özet bilgi vermektedir. Görüntü işleme alanında, bu dönemlerde gerçekleşen çok sayıda gelişme ve yenilik mevcuttur.

Görüntü işleme alanında, son yılların en önemli gelişmelerinden biri, yapay zeka ve derin öğrenme algoritmalarının kullanılmaya başlanmasıdır. Bu algoritmalar, özellikle görüntü tanıma, sınıflandırma ve nesne tespiti gibi görevlerde çok başarılı sonuçlar vermektedir.

Diğer önemli bir gelişme ise, görüntülerin çok daha yüksek çözünürlüklerde işlenmesine olanak sağlayan cihazların geliştirilmesidir. Örneğin, 4K ve 8K gibi yüksek çözünürlüklü ekranlar, görüntülerin çok daha detaylı bir şekilde işlenmesine olanak sağlamaktadır.

Son yıllarda, görüntü işleme alanında, Internet of Things (IoT) ve Nesnelerin Interneti (NI) gibi kavramların da önem kazandığı görülmektedir. Bu kavramlar, fiziksel nesnelerin internete bağlanarak birbirleriyle iletişim kurmasını sağlar. Bu sayede, örneğin, ev otomasyonu gibi uygulamaların gerçekleştirilebilmesine olanak sağlar. Bu alanlarda da görüntü işleme teknikleri kullanılmaktadır.

Bu örnekler, görüntü işleme alanında son yıllarda gerçekleşen önemli gelişmeleri göstermektedir. Bu alanda, gelecekte de çok sayıda yenilik ve gelişme beklenmektedir.

3.2.2 Görüntü işlemede kullanılan yapay zeka ve derin öğrenme algoritmaları

Görüntü işleme alanında kullanılan Yapay Zeka (YZ / AI) ve Derin Öğrenme (Deep Learning) algoritmaları, birçok farklı tipte olabilir. Örnekler:

1. Kümeleme algoritmaları: Bu algoritmalar, verilen görüntüleri benzer özellikleri olan gruplara ayırmayı amaçlar. Örneğin, görüntülerin renklerine göre ayrılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
2. Sınıflandırma algoritmaları: Bu algoritmalar, verilen görüntüleri önceden tanımlanmış sınıflara ayırmayı amaçlar. Örneğin, görüntülerin insan yüzleri olup olmadığının tespiti işlemi bu algoritmalar tarafından gerçekleştirilebilir.
3. Nesne tespiti algoritmaları: Bu algoritmalar, verilen görüntülerde belirli nesnelerin tespit edilmesini amaçlar. Örneğin, görüntülerde araba, insan yüzü gibi nesnelerin tespiti işlemi bu algoritmalar tarafından gerçekleştirilebilir.
4. Özelleştirme algoritmaları: Bu algoritmalar, verilen görüntüleri özelleştirme işlemini gerçekleştirir. Örneğin, görüntülerin renklerinin değiştirilmesi işlemi bu algoritmalar tarafından gerçekleştirilebilir.

5. Özellik çıkarımı algoritmaları: Bu algoritmalar, verilen görüntülerden özellikler çıkarır. Örneğin, görüntülerden renklerin çıkarılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
6. Konvolüsyonel sinir ağları (CNN): Bu algoritmalar, görüntülerin işlenmesi işlemini gerçekleştirir. Özellikle, görüntü tanıma ve sınıflandırma gibi görevlerde çok başarılı sonuçlar verir.
7. Öğrenme tabanlı algoritmalar: Bu algoritmalar, önceden işlenmiş görüntülerden öğrenir ve bu öğrenmeyi kullanarak yeni görüntüleri işler. Örneğin, görüntülerin renklerinin değiştirilmesi işlemini gerçekleştiren algoritmalar bu kategoride yer alabilir.
8. Öğrenme tabanlı çıkarım algoritmaları: Bu algoritmalar, verilen görüntüler üzerinde öğrenme tabanlı çıkarım işlemleri gerçekleştirir. Örneğin, görüntüler üzerinde özelliklerin çıkarılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
9. Büyük veri işleme algoritmaları: Bu algoritmalar, görüntülerin işlenmesi işlemini gerçekleştirirken büyük veri setlerini kullanır. Örneğin, görüntülerin sınıflandırılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
10. Öğrenme tabanlı çıkarım algoritmaları: Bu algoritmalar, verilen görüntüler üzerinde öğrenme tabanlı çıkarım işlemleri gerçekleştirir. Örneğin, görüntüler üzerinde özelliklerin çıkarılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
11. Öğrenme tabanlı denetim algoritmaları: Bu algoritmalar, verilen görüntüler üzerinde öğrenme tabanlı denetim işlemleri gerçekleştirir. Örneğin, görüntüler üzerinde düzenlemeler yapılması işlemi bu algoritmalar tarafından gerçekleştirilebilir.
12. Öğrenme tabanlı optimizasyon algoritmaları: Bu algoritmalar, verilen görüntüler üzerinde öğrenme tabanlı optimizasyon işlemleri gerçekleştirir. Örneğin, görüntüler üzerinde renklerin optimize edilmesi işlemi bu algoritmalar tarafından gerçekleştirilebilir.

Bu örnekler sadece birkaç örnektir, görüntü işleme alanında kullanılan yapay zeka ve derin öğrenme algoritmaları daha geniş bir yelpazede bulunabilir.

3.2.3 MATLAB ile Görüntü İşleme

MATLAB (MATrix LABoratory), MathWorks tarafından geliştirilen bir programlama dilidir. Bu dil, matematiksel işlemler ve hesaplamalar yapmak için tasarlanmıştır. MATLAB, kullanıcılarına birçok matematiksel işlemi kolaylıkla gerçekleştirmelerini sağlar. MATLAB, birçok matematiksel fonksiyon ve kütüphane içermektedir ve bu fonksiyonlar ve kütüphaneler sayesinde kullanıcılarına matematiksel işlemleri kolaylıkla gerçekleştirmelerini sağlar. MATLAB ayrıca, grafikler çizme, veri analizi ve çeşitli hesaplamalar yapma gibi özelliklere de sahiptir. Bu özellikler sayesinde MATLAB, birçok alanda kullanılmaktadır.

MATLAB, önceleri sadece matematiksel işlemler yapmak için kullanılmaktaydı. Ancak zaman içinde bu dilin kullanım alanı da genişlemiş ve MATLAB, birçok alanda kullanılmaya başlanmıştır. Örneğin, MATLAB sayesinde birçok matematiksel model oluşturulabilir ve bu modeller sayesinde birçok problem çözülebilir. MATLAB ayrıca, birçok veri analizi ve hesaplama işlemini de kolaylıkla gerçekleştirebilir. Bu özellikler sayesinde MATLAB, birçok alanda kullanılmaktadır. Örneğin, MATLAB sayesinde birçok matematiksel model oluşturulabilir ve bu modeller sayesinde birçok problem çözülebilir.

MATLAB ile yapay zeka uygulamaları yapılabilir. MATLAB, bir programlama dilidir ve bu dil sayesinde çeşitli matematiksel işlemler ve hesaplamalar yapılabilir. Ayrıca, MATLAB ile yapay zeka uygulamaları da yazılabilir. Örneğin, MATLAB ile bir sinir ağı modeli oluşturarak bu modeli eğitip, verilen bir görev için çözüm bulabilirsiniz. Bu sayede MATLAB ile yapay zeka uygulamaları yaparak bu alanda kendinizi geliştirebilirsiniz.

MATLAB kullanarak görüntü işlemek için, bir görüntü dosyasını okumak için “imread”, görüntüyü görüntülemek için “imshow” ve görüntüyü yeniden boyutlandırmak için “imresize” gibi yerleşik görüntü işleme işlevlerini kullanabilirsiniz. Görüntü filtreleme, görüntü segmentasyonu ve görüntü tanıma gibi daha gelişmiş işlemleri gerçekleştirmek için Görüntü İşleme Araç Kutusu gibi görüntü işleme araçlarını da kullanabilirsiniz. GIMP, ImageJ ve Adobe Photoshop gibi birçok başka görüntü işleme yazılımı seçeneği mevcuttur. Bu programlar, görüntü işleme için çok çeşitli özellikler ve araçlar sunar ve birçoğunu indirmek ve kullanmak ücretsizdir. Kendi özel görüntü işleme algoritmalarınızı ve işlevlerinizi geliştirmek için Python veya C++ gibi programlama dillerini de kullanabilirsiniz.

MATLAB, bir programlama dilidir ve genellikle matematik, bilim, mühendislik ve finans alanlarında kullanılır. Kanserli doku tespiti, bir sağlık uygulamasıdır ve bu konuda uzmanlık gerektirir. Ancak, genel olarak, MATLAB kullanılarak kanserli doku tespiti için aşağıdaki adımların izlenebileceği söylenebilir:

1. Öncelikle, kanserli doku tespit için gerekli görüntü verileri toplanmalıdır. Bu veriler, örneğin, radyografi görüntüleri ya da tomografi görüntüleri olabilir.
2. Daha sonra, bu görüntü verileri MATLAB kullanılarak işlenmelidir. Bu işlemler, görüntülerin netleştirilmesi, kontrast ayarlanması ve görüntülerin ön işleme aşamasında işlenmesi gibi adımları içerebilir.
3. Ardından, kanserli doku tespiti için bir tanıma modeli oluşturulmalıdır. Bu model, verilen görüntü verilerinden öğrenilen kanserli doku özelliklerini kullanarak yeni görüntülerde kanserli doku tespiti yapabilmelidir.
4. Modelin doğruluğu, eğitim verileri üzerinde test edilerek değerlendirilmelidir. Modelin doğruluğu, doğru tahmin yüzdesi veya başka bir performans ölçütü kullanılarak değerlendirilebilir.
5. Eğer model yeterince doğru sonuçlar veriyorsa, kanserli doku tespiti için kullanıma hazır hale gelmiş olacaktır. Bu model, yeni görüntü verilerine uygulandığında, kanserli doku tespitini otomatik olarak yapabilir.

Bu adımlar, MATLAB kullanılarak kanserli doku tespiti için genel bir yol haritası olarak düşünülebilir. Ancak unutulmamalıdır ki, bu konuda uzmanlık gerektiren bir alandır ve bu adımların detayları ve uygulama yöntemleri çok daha kapsamlı bir şekilde açıklanmalıdır.

3.2.4 Python ile görüntü işleme

Görüntü işleme, bir görüntünün özelliklerini belirleme, düzenleme veya değiştirme işlemidir. Bu işlemler, görüntülerin işlenmesi veya özel amaçlar için kullanılması gerektiğinde yapılır. Örneğin, bir görüntüdeki nesneleri tanıma, görüntüler arasında benzerlik arama veya görüntüleri sınıflandırma gibi işlemler görüntü işleme uygulamalarının önemli örnekleridir.

Python dilinde, görüntü işleme için birçok araç ve kütüphane mevcuttur. Bu kütüphaneler, görüntülerin yüklenmesi, düzenlenmesi ve işlenmesi gibi işlemler için kullanılabilir. Örneğin, Python'da en yaygın olarak kullanılan görüntü işleme kütüphanelerinden bazıları şunlardır:

- NumPy: Bu kütüphane, Python'da sayısal verilerin işlenmesi için kullanılır. NumPy, görüntülerin yüklenmesi, düzenlenmesi ve işlenmesi gibi işlemler için çok yararlıdır. Örneğin, NumPy kullanılarak görüntülerin pixel değerleri okunabilir ve değiştirilebilir.
- OpenCV: Bu açık kaynaklı kütüphane, görüntü işleme, bilgisayar görüşü ve video işleme gibi alanlarda çok yaygın olarak kullanılır. OpenCV, görüntülerin yüklenmesi, düzenlenmesi ve işlenmesi gibi birçok işlem için çok yararlıdır. Örneğin, OpenCV kullanılarak görüntüler üzerinde filtre uygulanabilir, nesneler tespit edilebilir ve görüntüler arasında benzerlik aranabilir.
- Pillow: Bu kütüphane, Python'da görüntülerin yüklenmesi, düzenlenmesi ve işlenmesi gibi işlemler için kullanılır. Örneğin, Pillow kullanılarak görüntüler üzerinde resim düzenleme işlemleri yapılabilir ve görüntü formatları arasında dönüştürme yapılabilir. Ayrıca, Pillow kütüphanesi, görüntüler üzerinde farklı çizimler ve yazılar ekleyebilme, görüntüleri döndürme ve ölçeklendirme gibi işlemleri de destekler.
- Scikit-image: Bu kütüphane, Python'da görüntülerin özelliklerini belirleme, sınıflandırma ve eşleştirme gibi işlemler için kullanılır. Scikit-image, görüntüler üzerinde farklı filtreler uygulayabilme, nesnelerin çerçevelenmesi gibi işlemleri de destekler.

Görüntü işleme işlemlerini gerçekleştirirken, öncelikle görüntüyü yüklemek gerekir. Bu işlem, Python'da OpenCV veya Pillow gibi kütüphaneler aracılığıyla yapılabilir. Daha sonra, görüntü üzerinde istenen işlemler gerçekleştirilebilir. Örneğin, görüntü üzerinde resim düzenleme işlemleri yapmak için Pillow kütüphanesi kullanılabilir veya görüntü üzerinde nesnelerin tespiti için OpenCV kullanılabilir.

Görüntü işleme işlemlerini gerçekleştirirken, Python dilinde yapılan çeşitli kodlamalar ile görüntüler üzerinde istenen işlemler yapılabilir. Örneğin, bir görüntü üzerinde resim düzenleme işlemleri yapmak için aşağıdaki kod bloğu kullanılabilir:

```
from PIL import Image

# Görüntüyü yükle
image = Image.open('image.jpg')

# Görüntüyü döndür
image = image.rotate(45)

# Görüntüyü kaydet
image.save('image_modified.jpg')
```

Şekil 3-1: Görüntünün yüklenme, döndürülme ve kaydedilmesi

```
from PIL import Image
# Görüntüyü yükle
image = Image.open('image.jpg')
# Görüntüyü döndür
image = image.rotate(45)
# Görüntüyü kaydet
image.save('image_modified.jpg')
```

Bu örnekte, bir görüntüyü yüklüyor, döndürüyor ve döndürülmüş görüntüyü kaydediyoruz. Bu şekilde, Python dilinde görüntü işleme işlemleri yapılabilir.

Görüntü işleme işlemlerinde, görüntüler üzerinde farklı filtreler uygulanabilir. Örneğin, bir görüntüyü gri seviyesine dönüştürmek için aşağıdaki kod bloğu kullanılabilir:

```
import cv2
import numpy as np

# Görüntüyü yükle
image = cv2.imread('image.jpg')

# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Gri seviyeli görüntüyü kaydet
cv2.imwrite('gray_image.jpg', gray_image)
```

Şekil 3-2: Görüntünün yüklenmesi, dönüştürme ve kaydedilmesi

```
import cv2
import numpy as np
# Görüntüyü yükle
image = cv2.imread('image.jpg')
# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Gri seviyeli görüntüyü kaydet
cv2.imwrite('gray_image.jpg', gray_image)
```

Bu örnekte, OpenCV kütüphanesi kullanılarak bir görüntüyü yüklüyor ve “cv2.cvtColor” fonksiyonu ile görüntüyü gri seviyesine dönüştürüyoruz. Daha sonra, gri seviyeli görüntüyü kaydediyoruz. Bu şekilde, Python dilinde görüntüler üzerinde farklı filtreler uygulanabilir.

Görüntü işleme işlemlerinde, görüntüler üzerinde nesnelerin tespiti de yapılabilir. Örneğin, bir görüntü üzerinde el yüzüklerinin tespiti için aşağıdaki kod bloğu kullanılabilir:

```

import cv2
import numpy as np

# Görüntüyü yükle
image = cv2.imread('image.jpg')

# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# El yüzükleri için özel bir cascades dosyası kullan
cascade_file = 'hand_cascade.xml'
hand_cascade = cv2.CascadeClassifier(cascade_file)

# Nesneleri tespit et
hands = hand_cascade.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5)

# Tespit edilen nesneleri çerçevele
for (x, y, w, h) in hands:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

# Görüntüyü göster
cv2.imshow('Detected hands', image)
cv2.waitKey()

```

Şekil 3-3: Nesnelerin tespiti

```

import cv2

import numpy as np

# Görüntüyü yükle
image = cv2.imread('image.jpg')

# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# El yüzükleri için özel bir cascades dosyası kullan
cascade_file = 'hand_cascade.xml'

hand_cascade = cv2.CascadeClassifier(cascade_file)

# Nesneleri tespit et

```

```

hands      =      hand_cascade.detectMultiScale(gray_image,      scaleFactor=1.1,
minNeighbors=5)
# Tespit edilen nesneleri çerçevele
for (x, y, w, h) in hands:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
# Görüntüyü göster
cv2.imshow('Detected hands', image)
cv2.waitKey()

```

Görüntü işleme işlemlerinde, görüntüler arasında benzerlik araması da yapılabilir. Örneğin, bir görüntü üzerinde el yüzüklerinin tespiti için aşağıdaki kod bloğu kullanılabilir:

```

import cv2
import numpy as np

# Görüntüyü yükle
image = cv2.imread('image.jpg')

# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# El yüzükleri için özel bir cascades dosyası kullan
cascade_file = 'hand_cascade.xml'
hand_cascade = cv2.CascadeClassifier(cascade_file)

# Nesneleri tespit et
hands = hand_cascade.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5)

# Tespit edilen nesneleri çerçevele
for (x, y, w, h) in hands:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

```

```
# Görüntüyü göster
cv2.imshow('Detected hands', image)
cv2.waitKey()
```

Şekil 3-4: Nesneleri çerçevele

```
import cv2
import numpy as np
# Görüntüyü yükle
image = cv2.imread('image.jpg')
# Görüntüyü gri seviyesine dönüştür
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# El yüzükleri için özel bir cascades dosyası kullan
cascade_file = 'hand_cascade.xml'
hand_cascade = cv2.CascadeClassifier(cascade_file)
# Nesneleri tespit et
hands = hand_cascade.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5)
# Tespit edilen nesneleri çerçevele
for (x, y, w, h) in hands:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
# Görüntüyü göster
cv2.imshow('Detected hands', image)
cv2.waitKey()
```

Bu örnekte, OpenCV kütüphanesi kullanılarak bir görüntüyü yüklüyor ve “cv2.CascadeClassifier” fonksiyonu ile el yüzüklerini tespit etme işlemi gerçekleştiriyoruz. Daha sonra, tespit edilen nesneleri çerçeveleyip görüntüyü gösteriyoruz. Bu şekilde, Python dilinde görüntüler üzerinde nesnelerin tespiti yapılabilir.

Görüntü işleme işlemlerinde, görüntüler arasında benzerlik araması da yapılabilir. Bu işlem, görüntüler arasında benzer özellikleri bulma ve bu benzer özellikleri kullanarak görüntüler

arasında eşleştirme işlemidir. Örneğin, bir görüntü üzerinde el yüzüklerinin tespiti için aşağıdaki kod bloğu kullanılabilir:

```
import cv2
import numpy as np

# Görüntüyü yükle
image = cv2.imread('image.jpg')

# Görüntüyü gri seviyesine dönüştür
```

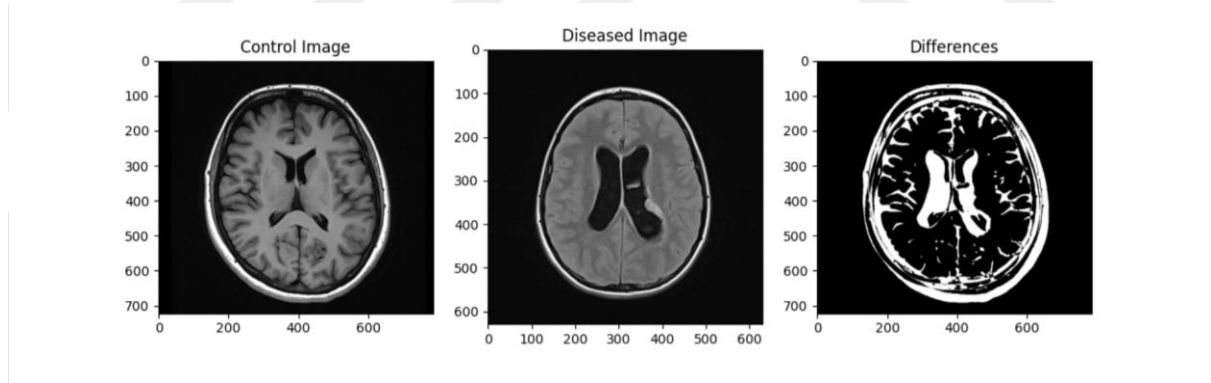
Şekil 3-5: Gri seviyeye dönüş

```
import cv2
import numpy as np
# Görüntüyü yükle
image = cv2.imread('image.jpg')
# Görüntüyü gri seviyesine dönüştür
```

4. BULGULAR

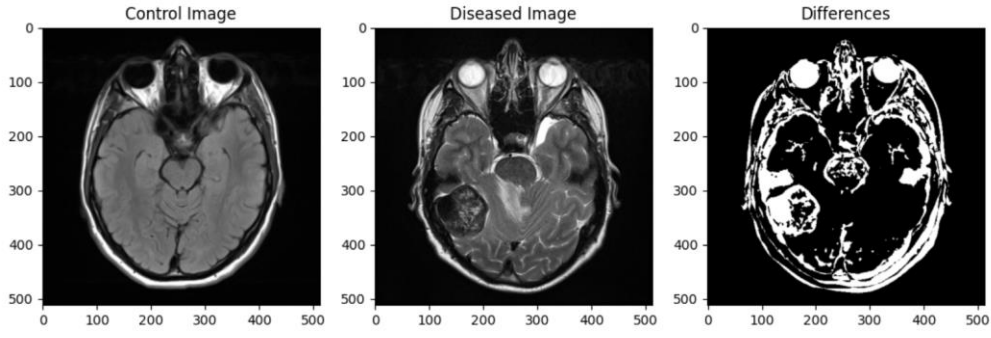
Öncelikle MATLAB ile yapılan çalışmalarımız ardından da Python ile tekrarlanmıştır. Python ile yapılan çalışmalar birkaç ayrı sebepten tercih nedenidir. Söz konusu sebepler; stabil bir yazılım dili olması, tezimizdeki çalışma konusu dışında pek çok alan için de uygulamalarının mevcudiyeti, pek çok kütüphane içeriyor olması hasebiyle kullanıcı için kolaylaştırıcı karakteri, kütüphane oluşturma avantajı ve bunu gibi pek çok gerekçe sıralanabilir.

Yapılan iş, çok basit bir şekilde şöyle özetlenebilir: Sağlıklı bir bireye ait bir kontrol görüntüsü ve hasta bir bireye ait bir görüntü veri olarak alınır. Hazırlanan kod temelde bu iki görüntüyü birbiri ile mukayese eder. Hasta veya hasta olduğundan şüphelenilen bireye ait görüntü sağlıklı bireyin görüntüsü ile piksellere ayrılarak detaylıca karşılaştırılır. Aşağıdaki ekran görüntülerinde bu verile görülmektedir. Sol yanda sağlıklı bireye ait görüntü verisi ve ortada da hasta bireyin görüntüsü mevcuttur. Her bir pikselde yoğunluk analizi sonucu, elde edilen fark da sağ tarafta yer almaktadır. Bu üçüncü görüntü bir girdi değil de bir çıktıdır. Şüpheli bölgeler bu şekilde bir tarama ile ön plana çıkarılmaktadır.



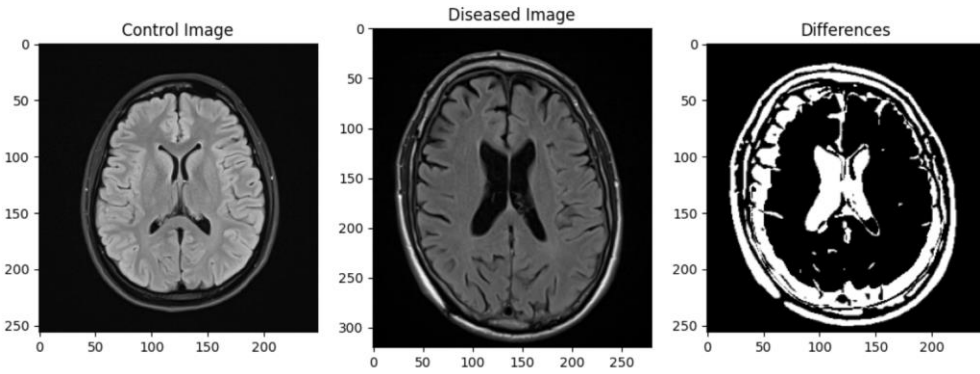
Şekil 4-1: Python ile yapılan analiz.

Ortada bulunan ve hastaya ait görüntü için kontrol görüntüsü ile mukayese edildiğinde otomatik olarak çıkan farklar sol görüntüde görünmektedir.



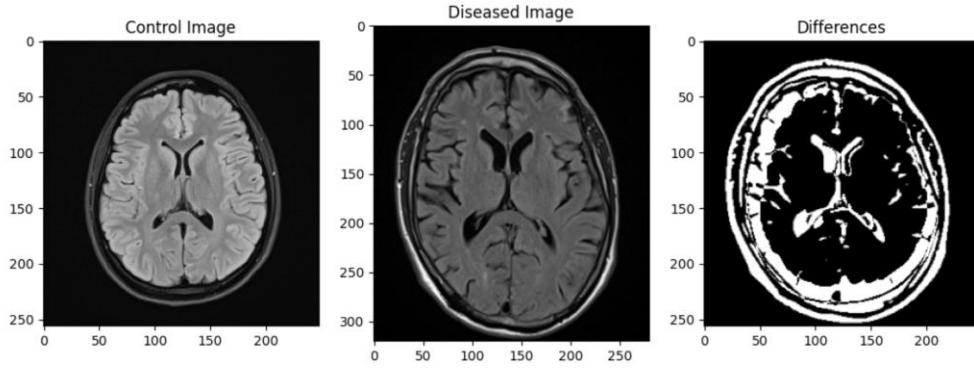
Şekil 4-2: Python ile yapılan analiz.

Bu görüntü tümör tespit etmiştir. Öte yandan görüntülerdeki beyin kısımları aynı boyutta olmadıkları için beyaz kısımlar fazla görünmektedir.



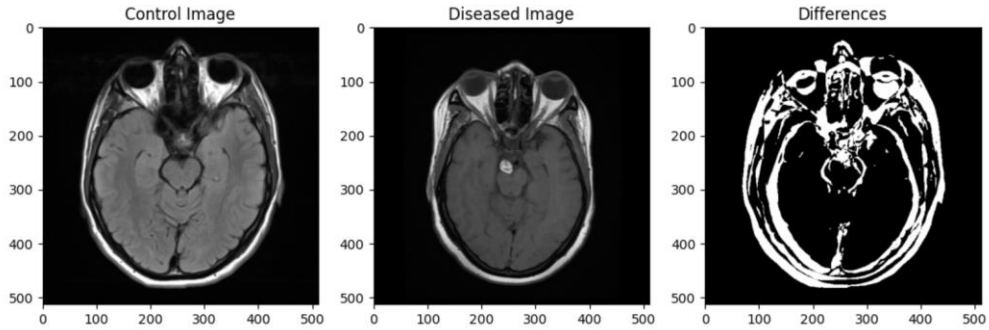
Şekil 4-3: Python ile yapılan Alzheimer analizi.

Bu görüntüde serebrospinal bölgenin nisbeten büyüklüğü görülmektedir. Bir Parkinson hastasına ait olan görüntünün kontrol grubu ile mukayese edilmesi sonucunda elde edilen görüntü görünmektedir. Kontrol görüntüsü ile hastaya ait görüntü karşılaştırılırsa aslında ton farkı olduğu da görünmektedir. Bu ton farkı bariz şekilde fark ettiğinde sağ görüntüde bu durumun ortaya çıktığı görünmektedir.



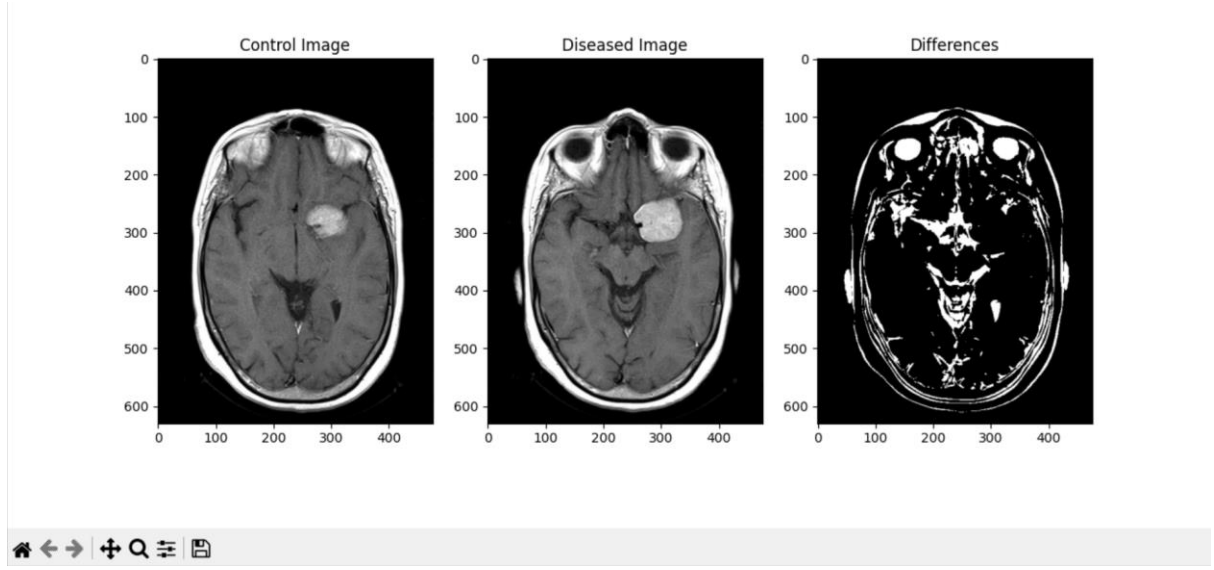
Şekil 4-4: Python ile yapılan ileri seviye Parkinson hastalığı analiz.

Bu görüntüde hastaya ait bazı görüntünün bazı bölgelerinde çevresine göre ton farkı gösterdiği görülmektedir. Bu tespit de Parkinson hastalığında nöronların ölümleri ile de uyumlu olduğu düşünülebilir.



Şekil 4-5: Python ile yapılan analiz.

Bu görüntüler farklı katmanlardan alınan görüntüler olduğundan çok fazla fark olduğu belgindir. Beyne ait aynı bir bölgenin farklı katmanlarından alınan görüntülerin birbirleri ile karşılaştırması yapılmıştır. Böylece bir tümör için adeta bir kontrol grubu oluşturulmuştur.



Şekil 4-6: Python ile yapılan analiz.

Yine, önceki şekildeki motivasyon ile aynı nedenle, farklı seviyelerden alınan görüntüler birbirleri ile incelenmiştir.

5. TARTIŞMA VE SONUÇ

5.1 Sonuç

Çalışmamızın başında elde mevcut veriler varsa bunları değerlendirmeye alacak bir algoritma oluşturulması hedeflenmekteydi. Önce MATLAB ile yaptığımız çalışma istediğimiz sonucu verdi. Bununla beraber Python ile de benzer bir çalışma yapıldığında, uygun şekilde bir algoritma oluşturulduğunda yine sistemin çalıştığı görülmüştür.

Çalışma esnasında beyin görüntülerinin aynı ebatlarda olmaması sorunu ile karşılaştık. Bu problemi görüntü işleme teknikleri ile çözdük.

Tömür için yapılan bu çalışmanın ayrıca; alzaymır, epilepsi, parkinson için de tekrarlanıp tekrarlanamayacağı bir araştırma konusu olarak bulunmaktadır.

5.2 Öneriler

Hazırladığımız algoritmada kullanıcı açısından birtakım iyileştirmeler ve filtrelemeler yapılması hedeflerimiz arasındadır. Örneğin filtrelemeleri cinsiyet ve yaş aralıklarına göre otomatik olarak yaptırmak öncelikli hedefimizdir.

EKLER

MATLAB:

```
% MATLAB code for
% Separate the brain part from MRI image.

% read the mri image.
k=imread("mrii.jpg");

% display the image.
imshow(k,[]);

% convert it into binary image.
k1=im2bw(k,graythresh(k));

% display the binary image.
imshow(k1);

% Make the brain largest connected component.
% We need to apply opening operation.
% define the structuring element.
SE=strel('disk',7,4);

% apply the opening operation.
k2=imopen(k1,SE);

% display the image now.
imshow(k2);

% apply connected component analysis.
b=bwlabel(k2);
```

```
% display the colored map image.
imshow(b,[]);

% brain is component labeled as 9.
% set all other component as 0 except brain.
b(b~=9)=0;

% display the brain part.
imshow(b);

% inside the brain part, black portion is there.
% close the black pixels inside brain part.
k3=imclose(b,strel('disk',18));

% display the brain part.
imshow(k3);

% extract the brain from original image.
k4=k3.*double(k);

% display the real brain from original image.
imshow(k4,[]);
```

```

% display the image.
imshow(k,[]);

% convert it into binary image.
k1=im2bw(k,graythresh(k));
|
% display the binary image.
imshow(k1);

% Make the brain largest connected component.
% We need to apply opening operation.
% define the structuring element.
SE=strel('disk',7,4);

% apply the opening operation.
k2=imopen(k1,SE);

% display the image now.
imshow(k2);

% apply connected component analysis.
b=bwlabel(k2);

% display the colored map image.
imshow(b,[]);

% brain is component labeled as 9.
% set all other component as 0 except brain.
b(b~=9)=0;

% display the brain part.

```

```

% display the image now.
imshow(k2);

% apply connected component analysis.
b=bwlabel(k2);

% display the colored map image.
imshow(b,[]);

% brain is component labeled as 9.
% set all other component as 0 except brain.
b(b~=9)=0;

% display the brain part.
imshow(b);

% inside the brain part, black portion is there.
% close the black pixels inside brain part.
k3=imclose(b,strel('disk',18));

% display the brain part.
imshow(k3);

% extract the brain from original image.
k4=k3.*double(k);

% display the real brain from original image.
imshow(k4,[]);

```



PYTHON

```
import cv2
import numpy as np
import os
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

# Define the directory containing the MRI images
directory = 'mri_images_directory'

# Get a list of all image filenames in the directory
image_files = os.listdir(directory)

# Initialize an empty 3D volume
volume = None

# Iterate over the image files
for filename in image_files:
    # Load the MRI image
    image_path = os.path.join(directory, filename)
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Apply thresholding to extract white zones
    _, white_zones = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY)

    # Invert the image
    white_zones = cv2.bitwise_not(white_zones)

    # Stack the white zones onto the volume
    if volume is None:
        volume = white_zones
```

```

else:
    volume = np.dstack((volume, white_zones))

# Generate coordinates for the volume
z, y, x = volume.shape
x_coords, y_coords, z_coords = np.meshgrid(np.arange(x), np.arange(y), np.arange(z))

# Create a 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Scatter plot the white zone coordinates
ax.scatter(x_coords[volume > 0], y_coords[volume > 0], z_coords[volume > 0], c='white',
marker='.')

# Set plot limits and labels
ax.set_xlim(0, x-1)
ax.set_ylim(0, y-1)
ax.set_zlim(0, z-1)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# Show the 3D plot
plt.show()

```

```
import cv2
import numpy as np
from tkinter import filedialog
import tkinter as tk
import matplotlib.pyplot as plt

# Function to enhance the image
""" Codeium: Refactor Explain Docstring """
def enhance_image(image):
    # Apply Gaussian blur to reduce noise and enhance features
    blurred = cv2.GaussianBlur(image, (5, 5), 0)

    # Apply adaptive thresholding to segment the image
    _, thresholded = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

    return thresholded

# Function to detect and draw circles around white zones
""" Codeium: Refactor Explain Docstring """
def draw_circles(image, regions):
    result_image = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
    for region in regions:
        cv2.circle(result_image, (region[1], region[0]), 10, (0, 0, 255), 2) # Draw a red circle
    return result_image
```

```
def compare_images(control_path, diseased_path):
    # Load control and diseased images
    control_image = cv2.imread(control_path, cv2.IMREAD_GRAYSCALE)
    diseased_image = cv2.imread(diseased_path, cv2.IMREAD_GRAYSCALE)

    # Enhance the images
    enhanced_control = enhance_image(control_image)
    enhanced_diseased = enhance_image(diseased_image)

    # Find the differences between the two images
    differences = cv2.absdiff(enhanced_control, enhanced_diseased)

    # Display the original, enhanced, and difference images side by side
    plt.figure(figsize=(15, 5))

    plt.subplot(1, 3, 1)
    plt.title("Control Image")
    plt.imshow(control_image, cmap='gray')

    plt.subplot(1, 3, 2)
    plt.title("Diseased Image")
    plt.imshow(diseased_image, cmap='gray')
```

```

def region_growing(image_path):
    # Load the MRI image
    mri_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Enhance the image
    enhanced_image = enhance_image(mri_image)

    # Define the initial region of the spot
    initial_region = np.array([[100, 100]])

    # Get the shape of the image
    image_shape = enhanced_image.shape

    # Create a mask to keep track of visited pixels
    visited = np.zeros(image_shape, dtype=bool)

    # Grow the region by adding neighboring pixels that have similar colors
    region = initial_region
    for _ in range(10): # Adjust the number of iterations as needed
        new_region = []
        for pixel in region:
            neighbors = np.array([[1, 0], [0, 1], [-1, 0], [0, -1]])
            for neighbor in neighbors:
                new_pixel = pixel + neighbor

                if (
                    (new_pixel >= 0).all()
                    and (new_pixel < image_shape).all()
                    and not visited[new_pixel[0], new_pixel[1]]
                ):
                    current_color = enhanced_image[new_pixel[0], new_pixel[1]]
                    if current_color == 255: # Check if the pixel is part of the white region
                        new_region.append(new_pixel)
                        visited[new_pixel[0], new_pixel[1]] = True
        region = np.array(new_region)

    # Draw circles around white zones
    result_image = draw_circles(mri_image, region)

    # Display the original and enhanced images with circles
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.title("Original MRI Image")
    plt.imshow(mri_image, cmap='gray')

    plt.subplot(1, 2, 2)
    plt.title("Enhanced Image with Detected Spots")
    plt.imshow(result_image)
    plt.show()

```

```

# Display the original and enhanced images with circles
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.title("Original MRI Image")
plt.imshow(mri_image, cmap='gray')

plt.subplot(1, 2, 2)
plt.title("Enhanced Image with Detected Spots")
plt.imshow(result_image)
plt.show()

# Feature extraction: Create an image with white zones
white_zones = np.zeros(image_shape, dtype=np.uint8)
white_zones[enhanced_image == 255] = 255

# Display the image with white zones
plt.figure()
plt.title("White Zones")
plt.imshow(white_zones, cmap='gray')
plt.show()

```

```

def browse_images():
    control_path = filedialog.askopenfilename(title="Select Control Image")
    if control_path:
        diseased_path = filedialog.askopenfilename(title="Select Diseased Image")
        if diseased_path:
            compare_images(control_path, diseased_path)
            region_growing(diseased_path)

# Create the main application window
app = tk.Tk()
app.title("MRI Image Analysis")

# Create a button to browse for control and diseased images
browse_button = tk.Button(app, text="Browse Images", command=browse_images)
browse_button.pack(pady=20)

# Run the main application loop
app.mainloop()

```

KAYNAKÇA

- [1]. Tortora, G.J., Derrickson, B. (2016). Principles of Anatomy and Physiology (15th ed.). J. Wiley. ISBN 978-1-119-34373-8
- [2]. Merkezî sinir sistemi, İnternet Adresi: https://tr.wikipedia.org/wiki/Merkez%C3%AE_sinir_sistemi, Son Erişim Tarihi: 25.12.2023.
- [3]. Beyin bölgeleri, İnternet Adresi: <https://www.dunyadanismanlikmerkezi.com/beyin-bolgeleri/>, Son Erişim Tarihi: 25.12.2023.
- [4] "Anatomy of the brain". (2018). Mayfield Brain and Spine.
- [5] "Brain Anatomy and how brain works" (2021). John Hopkins Medicine. (3&4).
- [6] "Anatomy of the brain". (2018). Mayfield Brain and Spine
- [7] <https://teachmeanatomy.info/neuroanatomy/structures/cerebellum/>
- [8] <https://www.lybrate.com/topic/brainstem-image>
- [9] Anshu Gupta and Tanimu Dwivedi . (2017). "A Simplified Overview of World Health Organization Classification Update of Central Nervous System Tumors 2016" Journal of Neurosciences in Rural Practices. 8(4): 629–641. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5709890/?report=printable>.
- [10] <https://www.coe.int/en/web/artificial-intelligence/history-of-ai>
- [11] Using artificial intelligence techniques to support clinical decisions in perioperative medicine

ÖZGEÇMİŞ

Adı Soyadı : Omar KHALİL
Doğum Yeri :
Doğum Tarihi :
e-posta adresi :
Adresi :
Eğitim Bilgileri
Lise : Amqa Lisesi (Lübnan)
Lisans : Lübnan Ulusal Arası Üniversitesi
Yabancı Dil ve Düzeyi Arapça – İngilizce – Türkçe

