**T.C.**

**MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLİLŞİM SİSTEMLERİ MÜHENDİSLİĞİ BÖLÜMÜ**

**LİSANS BİTİRME ÇALIŞMASI**

**Derin Öğrenme ve Görüntü İşleme Tekniklerini Kullanarak Covid-19 Röntgen Verilerinin Sınıflandırılması**

**Muhammed Furkan GÖKBEL**

**Osman ÖZKARACA**

**31 MAYIS 2019**

**MUĞLA**

**İçindekiler**

[1. GİRİŞ 4](#_Toc41058572)

[2. YAPAY ZEKA 5](#_Toc41058573)

[2.1 Makine Öğrenmesi 6](#_Toc41058574)

[2.1.1 Supervised(Gözetimli) Öğrenme 7](#_Toc41058575)

[2.1.2 Unsupervised(Gözetimsiz) Öğrenme 7](#_Toc41058576)

[2.1.3 Reinforcement(Takviyeli) Learning 7](#_Toc41058577)

[2.2 Derin Öğrenme 8](#_Toc41058578)

[2.2.1 Derin Öğrenme Mimarisi 9](#_Toc41058579)

[2.2.2 Derin Öğrenme Metotları 10](#_Toc41058580)

[2.2.3 Derin Öğrenme Kütüphaneleri 12](#_Toc41058581)

[2.2.4 Derin Öğrenmede Kullanılan Araçlar ve Diller 14](#_Toc41058582)

[2.3 Yapay Sinir Ağları 18](#_Toc41058583)

[3. EVRİŞİMSEL SİNİR AĞLARI 20](#_Toc41058584)

[3.1 Evrişim İşlemi Ve Evrişimsel Sinir Ağını Oluşturan Katmanlar 20](#_Toc41058585)

[3.1.1 Giriş Katmanı 20](#_Toc41058586)

[3.1.2 Konvolüsyon Katmanı 21](#_Toc41058587)

[3.1.3 Düzleştirilmiş Doğrusal Birim Katmanı(ReLu) 22](#_Toc41058588)

[3.1.4 Havuzlama Katmanı(Pooling Layer) 22](#_Toc41058589)

[3.1.5 Düzleştirme Katmanı(Flattening Layer) 23](#_Toc41058590)

[3.1.6 Tam Bağlantılı Katman(Full-Connection Layer) 23](#_Toc41058591)

[3.1.7 Dropout Katmanı 23](#_Toc41058592)

[3.1.8 Sınıflandırma Katmanı 23](#_Toc41058593)

[4. MATERYAL ve YÖNTEM 24](#_Toc41058594)

[4.1 Çalışma Veri Seti 24](#_Toc41058595)

[4.2 Eğitim ve Test Seti Oluşturma 24](#_Toc41058596)

[4.3 Model Katmanları Oluşturma 26](#_Toc41058597)

[4.4 Eğitilmiş Modeli Test Etme 28](#_Toc41058598)

[5. SONUÇ ve ÖNERİLER 31](#_Toc41058599)

[5.1 Sonuç 31](#_Toc41058600)

[5.2 Öneriler 32](#_Toc41058601)

[6. Kaynakça 33](#_Toc41058602)

**Özet**

Son zamanlarda sürekli artan internet kullanımı ve bunun sonucunda ortaya çıkan büyük veri dediğimiz kavramın çözümüne kolaylık getirmek için derin öğrenme algoritmaları geliştirilmiştir. Bu kavram yapay zekanın alt dalı olarak karşımıza çıkmaktadır. Görüntü işleme kavramı ise dijital ortamdaki görsellerin faydalı bilgiler edinmek için farklı yöntemlerle özdeşleştirebilme yöntemidir. Bu çalışmada mevcut veri setinde bulanan “Normal” ya da “Covid-19” xray görüntülerinin derin öğrenme yöntemi ve görüntü işleme teknikleri kullanılarak tespit etme amaçlanmıştır. Bu tahmin işleminde derin öğrenme kütüphanesi olan Keras ve derin öğrenme modellerinden Alex Net kullanılmıştır.

Uygulamada kullanılan veri seti, Kaggle sitesinden alınmıştır. Bu veri seti 400 eğitim verisi 63 test verisi içermektedir. Eğitim verilerinden bir kısmı validation ‘a ayrılmıştır.

**Anahtar Kelimeler:** Derin Öğrenme, Görüntü İşleme, Keras, İkili Sınıflandırma

# **GİRİŞ**

Teknolojinin gelişimiyle yapay zeka uygulamaları yaşamımızın birçok yerinde kullanılmaya başladı. Sağlık, savunma sanayii, istihbarat birimleri, uzay bilimleri ve siber güvenlik gibi alanlara önem veren devlet kurumları biriken verilerin işlenmesinde ve analizinde fazla zaman harcamadan ve zorlanmadan rapor almak istemektedir. Makine öğrenmesi yöntemlerini kullanarak bu veri yığınları anlamlı hale gelmekte ve gerekli sınıflandırmalar yapılabilmektedir. Son zamanlarda derin öğrenme yöntemlerinin gelişmesiyle beraber yapay zeka; görüntü, gen ve ses analizleri, robotik, otonom sistemler, tıp alanındaki hastalık teşhisleri gibi birçok alanda kullanılmaya başlanmıştır. Bu tür problemlerin çözümünde yüksek doğruluk elde etmesi yaygın bir kullanım alanı oluşmasını sağlamıştır [1]. Bu tezin amacı, derin öğrenme algoritmaları kullanarak “Covid 19 hastası bireyler” ve “normal bireyler” göğüs röntgenlerini kullanarak sınıflandırma işlemi yapmaktır. Veriler Kaggle sitesinden paylaşılan ve sınıflandırma için sunulan verilerden rastgele seçilmiştir. Veri setinin yaklaşık %75 i eğitim için kalan %25 i doğrulama seti olarak kullanılmıştır. Eğitimi gerçekleştirmek için açık kaynak kodlu Derin öğrenme kütüphanesi olan Keras kullanılmıştır. Araştırma konusu olan Korona Virüs ise (COVID-19), 13 Ocak 2020'de keşfedilen bir virüstür. Virüs, Aralık ayının sonlarında Wuhan 'da bir grup insanda ortaya çıkan bir grup solunum semptomlarıdır (ateş, öksürük, nefes darlığı). Salgın başlangıçta bölgedeki deniz ürünleri ve hayvan pazarlarında bulunan insanlarda keşfedilmiştir. Daha sonra insandan insana yayılmaya başlayan virüs başta Wuhan şehirinden, Hubei eyaletindeki diğer şehirlere ve Çin Halk Cumhuriyeti’nin diğer eyaletlerine ve diğer dünya ülkelerine yayılmıştır. En çok karşılaşılan belirtiler ateş, öksürük ve nefes darlığıdır. Şiddetli olgularda zatürre, ağır solunum yetmezliği, böbrek yetmezliği ve ölüm gelişebilmektedir. Hiçbir olgu görülmeden virüs tespit edilen bireylerde mevcuttur. Hastalar genelde 60 yaş üstü olup bağışıklık sistemi güçlü olan bireylerde hastalık hafif geçmektedir [2].

# **YAPAY ZEKA**

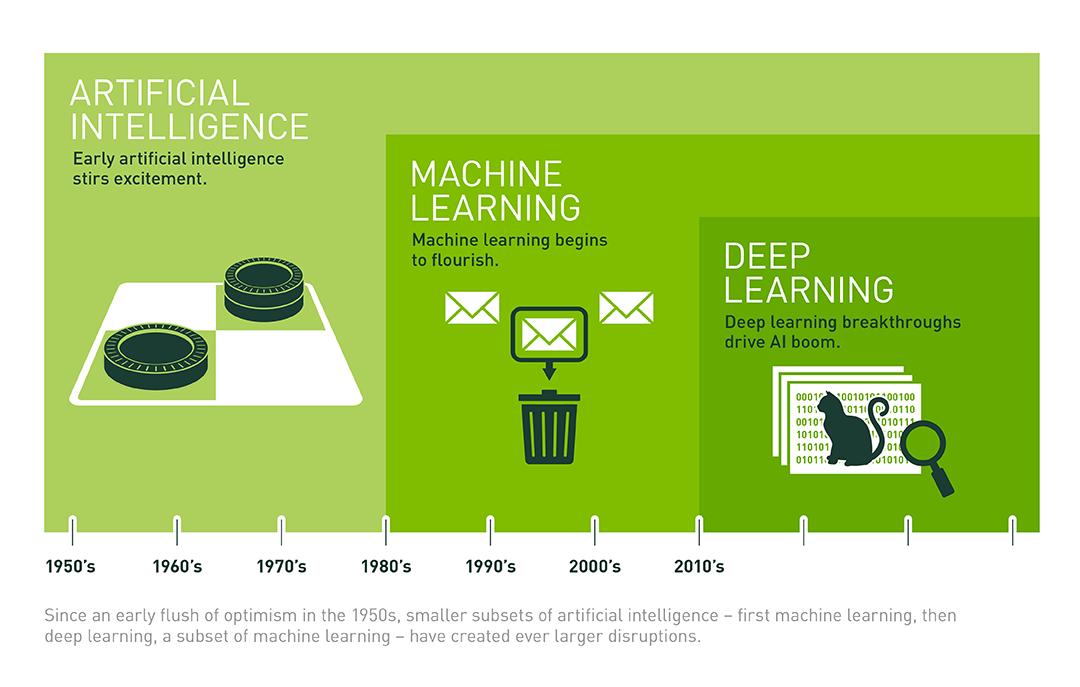
Geçmişi 1940’lı yıllarda Turing Makinelerine kadar giden bu buluş, insanın düşünme yöntemlerini analiz ederek oluşturulmuştur ve yine bu yeteneği analiz ederek geliştirilmektedir. Yapay zeka, bir makinenin veya bir bilgisayar programının insan gibi düşünüp karar verebilme yeteneğine denir. Yapay zeka bugünlerde finans ve ekonomi, askeriye, video oyunları, tıp gibi birçok alanda kullanılıyor.

Yapay Zeka birçok alt dallara ayrılmıştır. Bunlar;

* Makine Öğrenmesi
* Yapay Sinir Ağları
* Deep Learning(Derin Öğrenme)
* Doğal Dil işleme (Dil ile düşünme)
* Konuşma Sentezi (Yapay Konuşma)
* Konuşma Anlama (Konuşma Analizi)
* Uzman sistemler
* Örüntü Tanıma
* Genetik Algoritmalar
* Genetik Programlama
* Bulanık Mantık
* Çoklu Örnekle Öğrenme(Multiple Instance Learning) [3]

## Makine Öğrenmesi

Matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntem paradigmasıdır [4].



Şekil 1:Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme Karşılaştırması [5]

Makine öğrenimi 1990 yılında tekrar ayrı bir alan olarak gelişmeye başladı. Alan değişikliğinin amacı, insan problemlerini gerçek hayatta ele geçirmektir. Makine öğrenimi ve veri madenciliği genellikle aynı yöntemleri kullanır ve bu yöntemlerin çok fazla çakışması vardır. Bu yöntemler aşağıdaki gibi özetlenebilir ve ayırt edilebilir: Makine öğrenmesinin ana noktası, bilinen özelliklerden öğrenilen verilere dayalı tahminler yapmaktır [6].

Makine öğrenmesi algoritmaları üç gruba ayrılır: **gözetimli öğrenme** ve **gözetimsiz öğrenme ve takviyeli öğrenme.**

### **Supervised(Gözetimli) Öğrenme**

Bu öğrenme tekniğinde giriş değerleri ile istenen çıkış değerleri arasında eşleme yapan bir fonksiyon oluşturulur.

Eğitim verisi hem girdilerden hem de çıktılardan oluşur. Bu fonksiyon, classification (sınıflandırma) veya regresyon (eğri uydurma) algoritmaları ile belirlenebilir.

Veri setindeki çıkışlar kategorik ise classification(sınıflandırma), nümerik ise regresyon (eğri uydurma) algoritmaları kullanılır [7].

### **Unsupervised(Gözetimsiz) Öğrenme**

Bu yöntemde **işaretlenmemiş** veri üzerinden bilinmeyen bir yapıyı tahmin etmek için bir algoritma kullanan makine öğrenmesi tekniğidir. Burada giriş verisinin hangi sınıfa ait olduğu **belirsizdir** [7]**.**

### **Reinforcement(Takviyeli) Learning**

Amaç odaklı bir yöntem olduğu için diğer iki öğrenme yöntemine göre biraz farklılıklar içermektedir. Aslında davranış psikolojisinden yola çıkan bir öğrenme yöntemidir.

Asıl amaç Agent’ın (öğrenen etken) çevreyle etkileşerek çevreden geri bildirim (ödül), bu ödülleri maksimuma çıkartarak optimum policy’i(hareket tarzı) bulmasıdır [7].

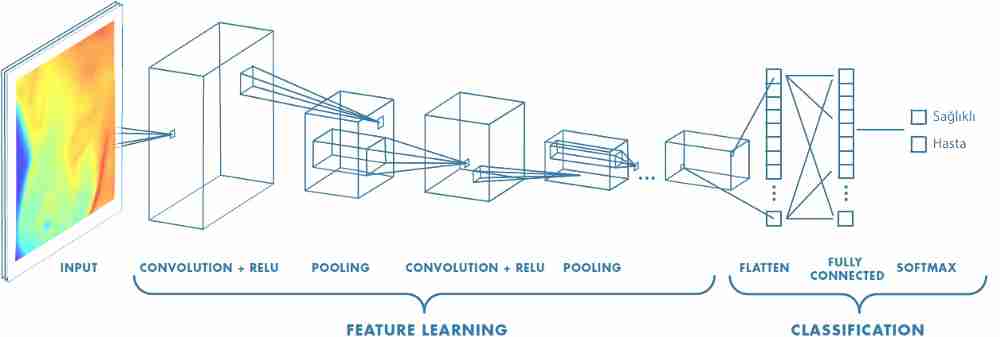
## Derin Öğrenme

Derin öğrenme, Makine öğrenmesi teknikleri arasında yer alan bir teknolojidir. Derin öğrenme terimi ile gündelik yaşantımızda insanoğlunun öğrenme yetisi bilgisayarlara aktarılarak veriler üzerinden istenilen işlemlerin gerçekleştirilmesi amaçlanmaktadır.

Derin öğrenme konusunu incelemeden önce, veri kavramını ayrıntılı olarak incelenmelidir. İnsan öğrenmesindeki gibi derin öğrenme teknolojisi de veriler aracılığıyla yapılır. Bilgisayarların ve mobil cihazların yaygın kullanımı ile her gün büyük miktarda veri üretilir. Bu kullanım nedeniyle büyük veri kavramı ortaya çıktı.

Büyük veri; sosyal medya, blog, fotoğraf, video ve loglardan toplanan verilerin anlamlı ve işlenebilir şekildeki halidir. Büyük veri kavramında büyük kelimesinin sayılarla ifade etmek gerekirse; bugün dünyanın en büyük kütüphanesi Amerika’dadır. Bu kütüphanedeki kaynaklar dijital ortama aktarıldığında 462 Terabyte’lık veri elde edilir. Günümüzde 1 yılda 8 Zettabyte veri üretimi yapılmaktadır [8].

Derin öğrenme algoritmaları, etiketlenmiş bir örüntü üzerinden öğrenme işlemini yaparken örüntü üzerindeki tüm pikselleri giriş verisi olarak almaktadır. Giriş verileri, örüntünün gri ya da renkli olmasına göre değişmektedir. Renkli bir görüntü modele giriş olarak verildiğinde kırmızı, yeşil ve mavi frekansların tamamı kullanılacağından mevcut pikselin 3 katı kadarı giriş verisi olarak kullanılmaktadır. Gri tonlu bir görüntüde ise giriş verisi piksel sayısı kadardır. Birincil olarak konvolüsyon katmanları, ReLu(Rektifiye Doğrusal Birim) katmanları ve havuzlama katmanları uygulanarak görüntü üzerindeki özellik haritası çıkarılmaktadır. Özellik haritası çıkarıldıktan sonra tam bağlantılı katmanlar ve Softmax’ ten meydana gelen özel sınıflandırma katmanları devreye girmektedir. Bu aşamada her pikselin her sınıf için ihtimal değerleri bulunmaktadır. Oluşan bu değerler Softmax tarafından ilgili pikselin hangi sınıfa ait olduğu belirlenmektedir [1].



Şekil 2:Evrişimli Sinir Ağı Katmanları [9]

### **Derin Öğrenme Mimarisi**

Derin öğrenme mimarisinde yer alan 6 mimari yapı tartışılmaktadır. Bu yapılardan ilki ve bu çalışmada kullanılan “Konvolüsyonel Sinir Ağları” mimarisi bilgisayarlı görme görevlerinde etkili ve bu çalışmanın modellenmelerinde kullanılan bir mimaridir. CNN, derin öğrenmede kullanılan temel mimari yapısıdır. İlk CNN çalışması Yann Le Cun tarafından 1988 yılında oluşturulmuştur .Bu mimari yapı 1998 yıllarına kadar geliştirilmiştir. CNN’ler görüntüleri girdi olarak almak için tasarlanmış yapılardır ve bilgisayarlı görmede etkili bir şekilde kullanılmaktadır. CNN, bir veya birden çok konvolüsyonel katman ve standart çok katmanlı bir sinir ağı gibi bir veya daha fazla tamamen bağlı katmanlardan oluşmaktadır. Bu katmanlar; Giriş Katmanı, Konvolüsyon Katmanı, Düzleştirilmiş Doğrusal Birim Katmanı, Havuzlama Katmanı, Tam Bağlantılı Katman, DropOut Katmanı ve Sınıflandırma Katmanıdır. Çalışmanın geliştirilmesinde seçilen mimari yapı da CNN’dir.

Diğer derin öğrenme mimari yapıları hakkında kısa bilgiler verilmiştir. Bu mimari yapılardan biri olan “Tekrarlayan Sinir Ağları”; birimler arası verilerin yönlendirilmiş bir döngü oluşturduğu yapay sinir ağıdır. Tekrarlayan sinir ağları, görüntü tanımlayıcıları kullanmak için diğer mimarilerle birlikte kullanılmaktadır. “Uzun-Kısa Vadeli Hafıza Ağları” mimarisi; değerleri rastgele aralıklarla hatırlayan bir tekrarlayan sinir ağı mimarisidir. Bu mimari yapısı, konuşma/metin işleme çalışmalarında kullanıldığında iyi sonuçlar vermektedir. “Sınırlı Boltzman Makineleri”; girdi seti üzerinde olasılık dağılımlarını öğrenebilen görünür katman ve gizli katman olmak üzere 2 katmandan oluşan bir yapay sinir ağı yapısıdır.

Sınırlı Boltzman Makineleri; sınıflandırma, özellik öğrenimi ve modelleme için uygun bir ağ olarak bilinir. “Derin İnanç Ağları” yapısı; Sınırlı Boltzman Makinelerinin yığını ve oto kodlayıcılarla bileşimi olarak görülür. Derin İnanç Ağları, görüntü tanıma uygulamalarında kullanılmaktadır. “Derin Oto Kodlayıcılar” mimari yapısı ise; denetimsiz öğrenme için ve verinin üretken modellerini öğrenmek için kullanılan, girdi katmanı verilerini, çıktı katmanına kopyalayan bir yapıdır . [10]

### **Derin Öğrenme Metotları**

Derin öğrenmede temel olarak aşağıdaki metotlar uygulanır.

• Sınıflandırma (Classification)

• Kümeleme (Clustering)

• Eğri Uydurma (Regresyon)

• Birliktelik Kural Çıkarımı (Association Rule Mining)

• Transkripsiyon (Transcription)

• Makine Çevirisi (Machine Translation)

• Anormal Durumların Farkındalığı (Anomaly Detection)

• Gürültü Azaltma (Noise Reduction)

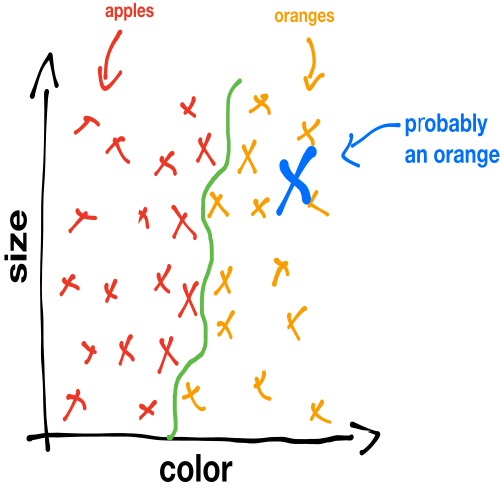
• Metin Özetleme (Text Summarization)

Bu çalışmanın geliştirilmesinde seçilen metot ise Sınıflandırmadır. Sınıflandırma işleminin tanımı aşağıdaki paragrafta verilmiştir.

#### **Sınıflandırma (Classification)**

Sınıflandırma yönteminde giriş ve çıkış parametreleri vardır. Giriş parametrelerine bakarak çıkış parametresi belirlenir. Yani geçmiş bilgilerin hangi sınıftan olduğu biliniyorsa, yeni gelen verinin hangi sınıfa dâhil olacağının bulunmasıdır [11]

Elimizde bulunan meyvelerin renklerine göre sınıflandırılması problemi çözülmek istenebilir. Bu problemi çözümlemek için önce bir model oluşturulur ve verilerle bu model eğitilerek cevabı aranan sınıflandırma işlemleri tahmin edilmeye çalışılır. Bu problemi çözmek için renk ve büyüklük değerlerine bakılarak meyvenin hangi sınıfta yer aldığı belirlenebilir [12].

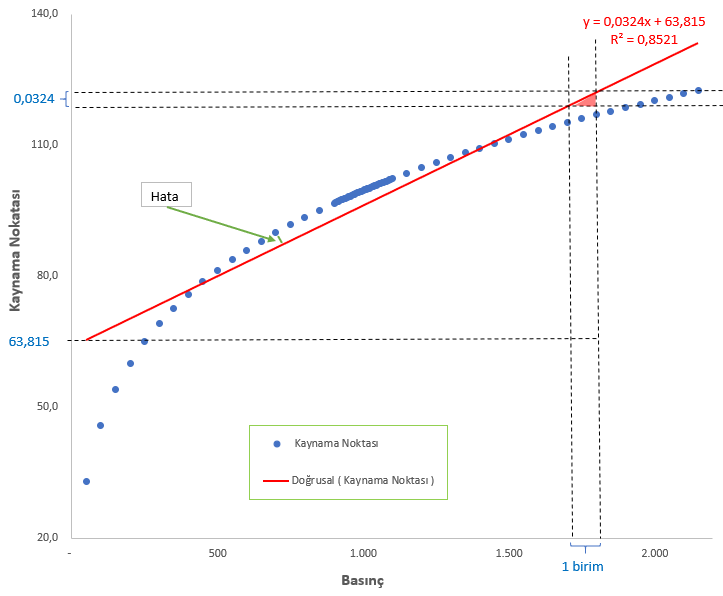


Şekil 3:Sınıflandırma Metodu Gösterimi [13]

#### **Eğri Uydurma (Regresyon)**

Regresyon bir eğri uydurma yöntemidir. Bu eğri uydurma işini, bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi kullanarak oluşturduğumuz Regresyon denklemi ile yaparız.

Örneğin regresyon analizi, aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi belirlemek ve bu ilişkiyi kullanarak o konu ile ilgili tahminler ya da kestirimler yapabilmek amacıyla yapılır [14].

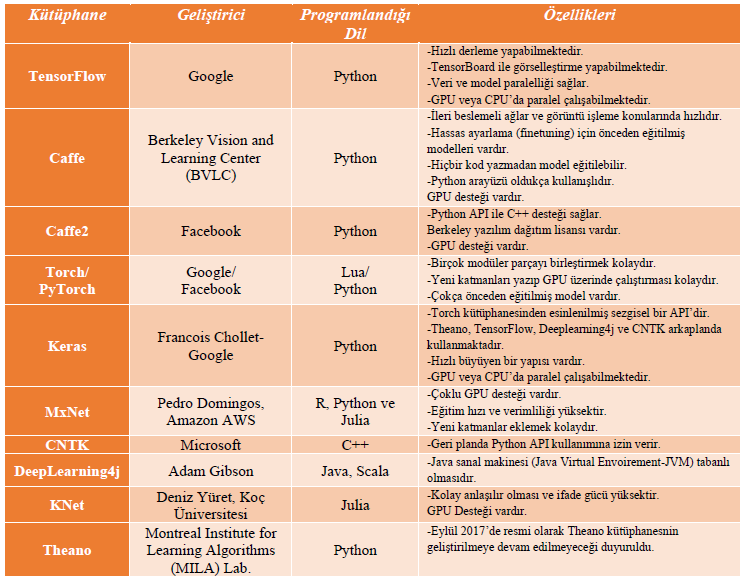


Şekil 4:Basit Regresyon Grafiği [14]

### **Derin Öğrenme Kütüphaneleri**

Makine Öğrenmesi ve Derin Öğrenmede kullanılmak üzere bazı üniversiteler ve şirketler tarafından geliştirilmiş farklı özelliklere sahip ve derin öğrenme çalışmalarını pratikleştirip kolaylaştıran birçok hazır kütüphane ve API’lar (Application Programming Interface-Uygulama Programlama Arayüzü) bulunmaktadır. Bu çalışmada Keras Kütüphanesi kullanılmıştır.

Kullanım yoğunluğuna göre bu kütüphaneler aşağıdaki tablodaki gibidir ;



Şekil 5:Derin Öğrenme Kütüphaneleri [15]

### **Derin Öğrenmede Kullanılan Araçlar ve Diller**

Yapay zekâ, makine öğrenmesi ve derin öğrenme konuları üzerinde kullanılabilecek ücretli, ücretsiz ve açık kaynak kodlu pek çok platform mevcuttur. Yapacağımız çalışmalarda dikkat etmemiz gereken kısımlar; veri seti içerisindeki hangi verilerin kullanılacağı, veriler arasındaki ilişkilerin belirlenmesi ve verilerin hangileri kullanılarak en iyi sonuca varılabileceği gibi konulardır.

Yapay zekâ alanında en çok tercih edilen araçlar ve diller; Matlab, Python, Weka, OpenCV, C++, Java ve C# olarak listelenebilir.

**Matlab**

Matlab, MathWorks şirketi tarafından geliştirilmiştir. Lineer, istatistik, optimizasyon çalışmaları, analiz işlemleri gibi çok karmaşık matematiksel hesaplamaları bile hızlı bir şekilde yapabilme kapasitesine sahiptir. Ayrıca yapay zeka, sinyal işleme, görüntü işleme, sayısal analiz konularında da sıklıkla kullanılan bir dildir.

**Python**

Python, nesne yönelimli, yorumlanabilir, modüler ve etkileşimli üst düzey bir dildir. Programlama dilleri makine mantığı ile insan mantığı arasındaki köprüdür. Dil ve makine mantığının yakınlığı, makinelerde daha hızlı çalışmalarını sağlar. Bununla birlikte, makine mantığına yaklaşmak, insanlar için öğrenmeyi zorlaştırır. Bir programlama dili makine mantığından saparsa ve insan mantığına eğilimliyse, yüksek seviyeli bir programlama dili denir, bu yüzden ki Python yüksek seviyeli bir dildir. Birçok dilden öğrenmek daha kolaydır. Python ile ihtiyaç duyduğunuz pek çok işi, az sayıda kod satırı ile başarabilirsiniz. Masaüstü uygulamaları, web uygulamaları, veri analizi ve görselleştirme uygulamaları gibi birçok programı Python ile kolaylıkla yazabilirsiniz.

Python programlama dili, bilgisayar programcısı Guido van Rossum tarafından geliştirildi. Python dili tanıtıldıktan sonra kısa sürede pek çok araştırmacı tarafından kullanılarak popüler hale geldi. Bu popülerliğin sebebi arasında aşağıdaki başlıklar gösterilebilir.

* Açık kaynak olarak yayınlanması
* Öğrenilmesi kolay
* Tüm platformlarda çalışabilme
* Geniş topluluk desteği
* Modüler sistem

Programlama dilleri yapısal olarak ikiye ayırabiliriz, bunlar derleyici ve yorumlayıcıdır. Derleyici programlama dilleri, yazılan programın bilgisayarın anlayabileceği ve çalıştırabileceği makine koduna çeviren programdır. Buna örnek olarak C# veya Java ile yazılmış kod blokları program üzerinde yer alan derleyici tarafından öncelikle derlenir. Bu derlenme işleminde yazılan tüm kodlar test işleminden geçirilerek kod bloklarında herhangi bir hata olup olmadığı kontrol edilir. Yorumlayıcı programlama dilleri ise derlenme işlemi gerektirmeyen, yüklendiği sistem üzerinde doğrudan çalıştırılan programdır. Bunlara örnek olarak Php ve Python Programlama dillerini verebiliriz. Bu programlar ile yazılan kodlar satır satır çalıştırılır. Çalışması gerekmeyen kod blokları gerek duyulmadıkça çalıştırılmaz.

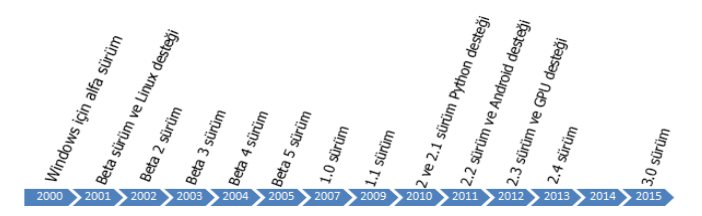
Son yıllarda yapay zekâ alanındaki gelişmelerin artması sonucu Python üzerinde de bu konu ile alakalı kütüphaneler oluşturulmuştur. Bunlardan bazıları “Tensorflow”, “Keras” ve “Scikit Learn” kütüphaneleridir [8].

**WEKA**

WEKA, makine öğrenmesi alanında sıklıkla kullanılan programlardan biridir. Veri düzenleme, sınıflandırma, regresyon ve kümeleme gibi birçok alanda kullanılan araçları içermektedir. WEKA, GNU (Genel Kamu Lisansı) ve açık kaynak olarak servis edilmektedir. Waikato üniversitesi içerisinde geliştirilen program “Waikato Environment for Knowledge Analysis “(Bilgi Analizi için Waikato Ortamı) kelimelerinin ilk harfleri yazılarak WEKA olarak adlandırılmıştır.

Program içerisinde istatistik ve makine öğrenmesi alanında kullanılan kütüphaneler mevcuttur. Veri işleme, regresyon gibi işlemler dışında veri görselleştirme, özellik çıkarımı gibi işlemler de yapılmaktadır.

**OpenCV**

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında İntel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. OpenCV kütüphanesi içerisinde görüntü işlemeye ve makine öğrenmesine yönelik iki bin beş yüzden fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR gibi işlemler rahatlıkla yapılabilmektedir.

Şekil 6:Yıllara Göre OpenCV Sürümleri [16]

**C++**

C++ 1979 yılında Bjarne Stroustrup tarafından C dili kullanılarak geliştirilmeye başlanan bir programlama dilidir. C++ günümüzde pek çok platformda çalışabilecek sistem, uygulama ve sürücü yazılımları üretmek için kullanılmaktadır. C#, Python, Php ve Java gibi programlama dillerinin oluşturulmasında etkili olan C++ programlama dili günümüzde en çok kullanılanlar arasında yerini korumaktadır. Yapay zekâ çalışmalarında da kullanılan C++ birçok araştırmacı tarafından çalışmalarında kullanılmaktadır [8].

**Java**

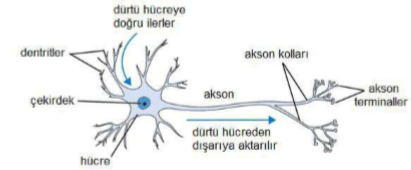
Java, Sun Microsystems firması tarafından 1995 yılında çıkarılan bir programlama dilidir. Java dili web içerikleri, mobil programlar, masaüstü programları, kurumsal yazılımlar ve yapay zekâ gibi alanlarda kullanılır. Java ile kodlanan programların çalışması için özel bir platform olmasına gerek yoktur. Windows, MacOs, Linux, Unix gibi işletim sistemlerinin tamamında Java rahatlıkla kullanılabilir. Bu durum Java’nın tercih edilmesini arttıran bir etkendir [8].

**C#**

C# Microsoft firması tarafından geliştirilmiş, öğrenim aşamasında ücretsiz olarak sunulan, fakat ticari olarak ücretli sunulan bir programlama dilidir. C ve C++ dillerinin etkileşiminde türetilen nesne yönelimli bir dildir. Masaüstü, sunucu, gömülü sistemler, sunucu işlemler, web programcılığı gibi yazılımlar geliştirilmektedir. Son zamanlarda gelişen teknoloji ile görüntü işleme, yapay zekâ gibi alanlarda da kullanılmaktadır [8].

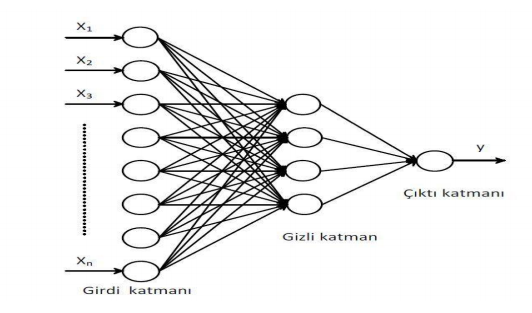
## Yapay Sinir Ağları

Yapay zekâ biliminin alt dallarından biri olan yapay sinir ağları (YSA), birbiriyle bağlanan ağırlıklı bağlantıların oluşturduğu, insan beyninin sinir ağlarına benzeyen ve kendi içinde belli bir belleğe sahip olan paralel bilgi işleme sistemleridir. Diğer bir tanımıyla, YSA, sinir sistemindeki biyolojik ağ yapısını taklit edinmek üzere hazırlanmış bilgisayar programlarıdır.



Şekil 7:İnsan Nöron(Sinir Hücresi) Yapısı [1]

Biyolojik sinir ağlarında olduğu gibi YSA’da da temel bilginin işlendiği en küçük temel unsur sinir hücresidir. Yapay sinir hücreleri bir ya da birden fazla girdi alarak çıkış üretirler ve bu çıkışlar başka nöronların girişi olabilir.



Şekil 8: Bir Yapay Sinir Ağ Yapısı Örneği [17]

Şekil 8 ile bir YSA yapısı ifade edilmektedir. Bu katmanlar;

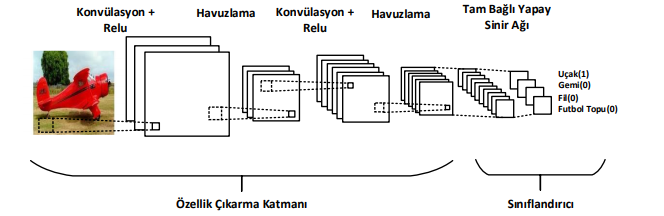
Girdi Katmanı: Dışarıdan gelen bilgileri ara (gizli) katmana iletmekle görevli hücrelerden oluşurlar.

Ara (gizli) Katman: Girdi katmanından gelen bilgileri işleyerek çıktı katmanına iletir. Bu katmanda doğrusal olmayan davranışları sağlayan işlemler mevcuttur ve bilgiler bu katmandaki hücrelerde işlenir. Bir YSA’da birden fazla ara katman olabilir.

Çıktı Katmanı: Ara katmandan gelen bilgileri işleyerek girdi setine karşılık gelen çıktı setini üretirler [17].

# **EVRİŞİMSEL SİNİR AĞLARI**

Evrişim işleminin temel hedefi giriş olarak verilen görüntüden, sesten ya da metinden bilgilerin veya özelliklerin çıkarılmasıdır. Sınıflandırma başta olmak üzere birçok bilgisayarlı görü alanında kullanılan evrişimsel sinir ağları klasik sinir ağlarından farklı olarak, evrişim özellik çıkarma ve sınıflandırma katmanlarını içermektedir. Evrişim katmanı konvolüsyon (convolution), aktivasyon fonksiyonu (relu) ve havuzlama (pooling) aşamalarını içermekte ve yüksek boyutlu veriden düşük boyutlu özniteliklerin çıkarılmasını sağlamaktadır. Sınıflandırma katmanı ise düşük boyutlu öznitelikler ile kategorilerin eşleşmesini sağlar.



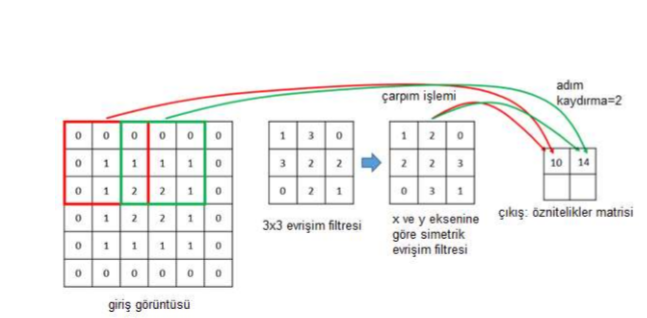
Şekil 9:Klasik bir Evrişimsel Sinir Ağı mimarisi [18]

## Evrişim İşlemi Ve Evrişimsel Sinir Ağını Oluşturan Katmanlar

### **Giriş Katmanı**

Bu katman ESA’nın ilk katmanını oluşturmaktadır. Giriş verimiz işlenmeden bu aşamaya girer. Giriş verisinin çözünürlüğü ve boyutu modelin başarımı için önemlidir. Eğer görüntü boyutu yüksek seçilirse eğitim süresini ve test süresini uzatabilir ve bellek ihtiyacımız artabilir. Ama sonuç olarak modelin başarısı da bu şekilde artar. Giriş verisinin boyutu küçük seçildiğinde bellek ihtiyacı azalır, eğitim süresi düşebilir ancak modelin performansı düşebilir. İdeal giriş verisini bulmak hem performans açısından hem de eğitim süresi açısında önemlidir [1].

### **Konvolüsyon Katmanı**

Bu katman CNN’nin ana yapı taşıdır. Resmin özelliklerini algılamaktan sorumludur. Bu katman, görüntüdeki düşük ve yüksek seviyeli özellikleri çıkarmak için resme bazı fitreler uygular. Örneğin, bu filtre kenarları algılayacak bir filtre olabilir. Bu filtreler genellikle çok boyutludur ve piksel değerleri içerirler.(5x5x3) 5 matrisin yükseklik ve genişliğini, 3 matrisin derinliğini temsil eder [19]. 

Şekil 10: Örnek bir Evrişim İşlemi [1]

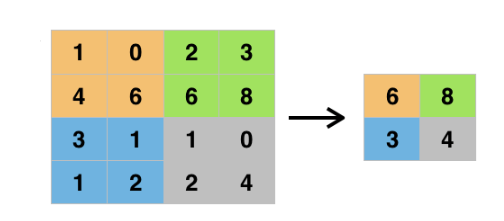
Öncelikle, filtre görüntünün sol üst köşesine konumlandırılır. Burada, iki matris arasındaki indisler birbiri ile çarpılır ve tüm sonuçlar toplanır, daha sonra sonucu çıktı matrisine depolanır. Ardından, bu filtreyi sağa bir basamak kadar hareket ettirip işlemi tekrarlanır. Bu işlem tüm satırlar için uygulanır. Tüm işlemler bittikten sonra çıktı matrisi oluşturulur. Burada çıktı matrisinin 3×3 olmasının nedeni ise 5×5 matrisinde, 3×3 filtresi yatayda ve dikeyde üç kez hareket etmesinden kaynaklanır.

 Bu matrise Feature Map denir. Filtre tarafından temsil edilen özellikte görüntünün bulunduğu yeri gösterir. Yani filtreyi görüntü üzerinden hareket ettirerek ve basit matris çarpımını kullanarak, özelliklerimizi tespit ediyoruz [19].

### **Düzleştirilmiş Doğrusal Birim Katmanı(ReLu)**

Evrişimsel Sinir Ağları nöronlarının çıktıları için en yaygın şekilde kullanılan birimdir. ReLu, konvolüsyonel katmanlardan sonra gelir. Bu katmana aktifleştirme katmanı da diyebiliriz. ReLu deki amaç katmana yaptığı etki negatif değerleri sıfıra çekmesidir. ReLu, kendinden önceki katmanda doğrusal olan yapıyı doğrusal olmayan yapıya dönüştürmek için kullanılır. Dolayısıyla ağın öğrenimi daha hızlanmış olur.

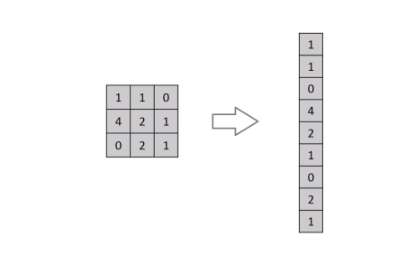
### **Havuzlama Katmanı(Pooling Layer)**

Havuzlama katmanı uygulamasında temel amaç, kendinden sonra gelen Konvolüsyon katmanı için giriş boyutunu azaltmaktır. Genellikle ReLu katmanından sonra gelir. Bu katman sonucu boyuttaki azalma bilgi kaybına sebep olur. Bilgi kaybından dolayı bir sonraki katmanlarda daha az hesap yükü oluşturur ve sistemin ezberlemesini önler.

Şekil 11:Max Pooling İşlemi [19]

### **Düzleştirme Katmanı(Flattening Layer)**

Bu katmanın görevi basitçe, son ve en önemli katman olan Fully Connected Layer’ın girişindeki verileri hazırlamaktır. Genel olarak, sinir ağları, giriş verilerini tek boyutlu bir diziden alır. Bu sinir ağındaki veriler ise Convolutional ve Pooling katmanından gelen matrislerin tek boyutlu diziye çevrilmiş halidir.



Şekil 12:Flatting İşlemi [19]

### **Tam Bağlantılı Katman(Full-Connection Layer)**

Evrişimli Sinir Ağları mimarilerinde birbirini takip eden Konvolüsyonel, ReLu ve havuzlama katmanlarında sonra tam bağlantılı katman gelir. Verileri Flattening işleminden alır ve Sinir ağı yoluyla öğrenme işlemini geçekleştirir.

### **Dropout Katmanı**

ESA ‘da eğitim aşamasında fazla eğitimden dolayı bazen ağ ezberleme yapar. Ağın ezber yapmasının önüne geçmesi için bu katman kullanılır. Bu katmanın asıl amacı ağdaki bazı düğümlerin kaldırılmasıdır [1].

### **Sınıflandırma Katmanı**

Tam bağlantılı katmandan sonra gelen katman sınıflandırma katmanıdır. Sınıflandırma işlemi derin öğrenme algoritmalarında bu katmanda yapılmaktadır. Eğitim esnasında kaç sınıfımız varsa tahmin edilecek ya da sınıflandırma yapılacak, aynı sayıda bu katmanın çıkış değeridir. Sınıflandırıcı olarak genellikle softmax kullanılır [1].

# **MATERYAL ve YÖNTEM**

## Çalışma Veri Seti

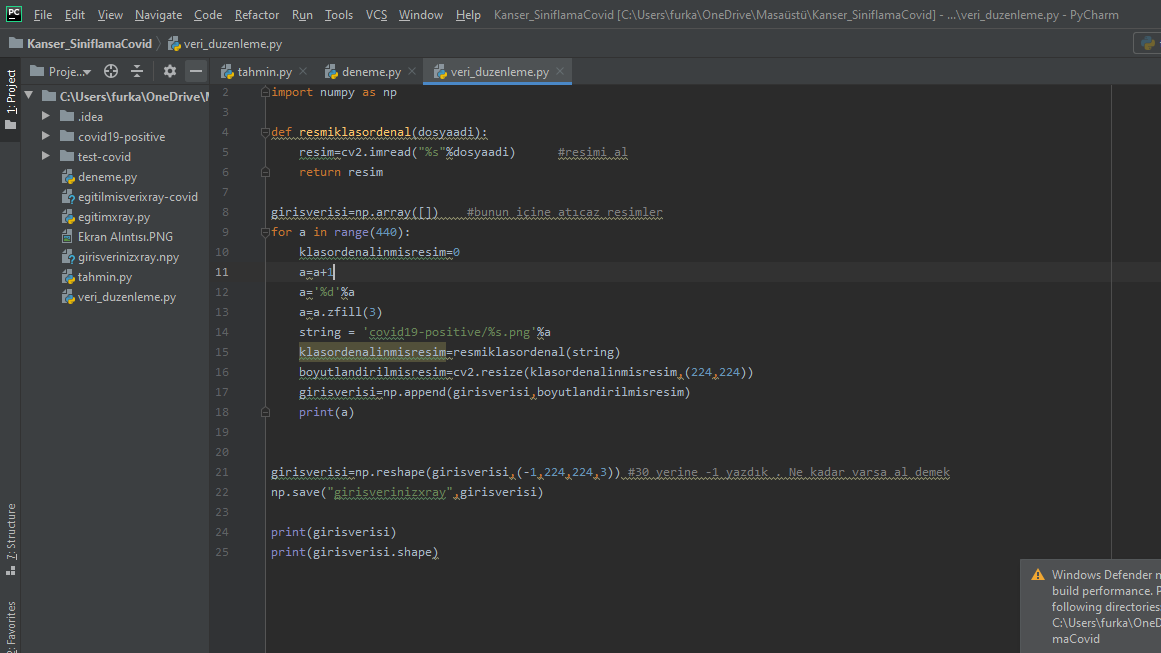
Araştırmada kullanılan Xray görüntü verileri; 200 sağlıklı birey, 200 Covid-19 hastası birey grubundan oluşmaktadır. Ayrıca eğitim aşamasında kullanılmamış 63 adet test verisi vardır. Bu çalışmada Kaggle sitesinde araştırmacıların, kullanıcılar için sunduğu veri setleri kullanılmıştır.

Kullanılan verileri sunan araştırmacılar;

* [Wei Hao Khoong](https://www.kaggle.com/khoongweihao) [20]
* [Paul Mooney](https://www.kaggle.com/paultimothymooney) [21]
* Devakumar kp [22]
* [bachir](https://www.kaggle.com/bachrr) [23]

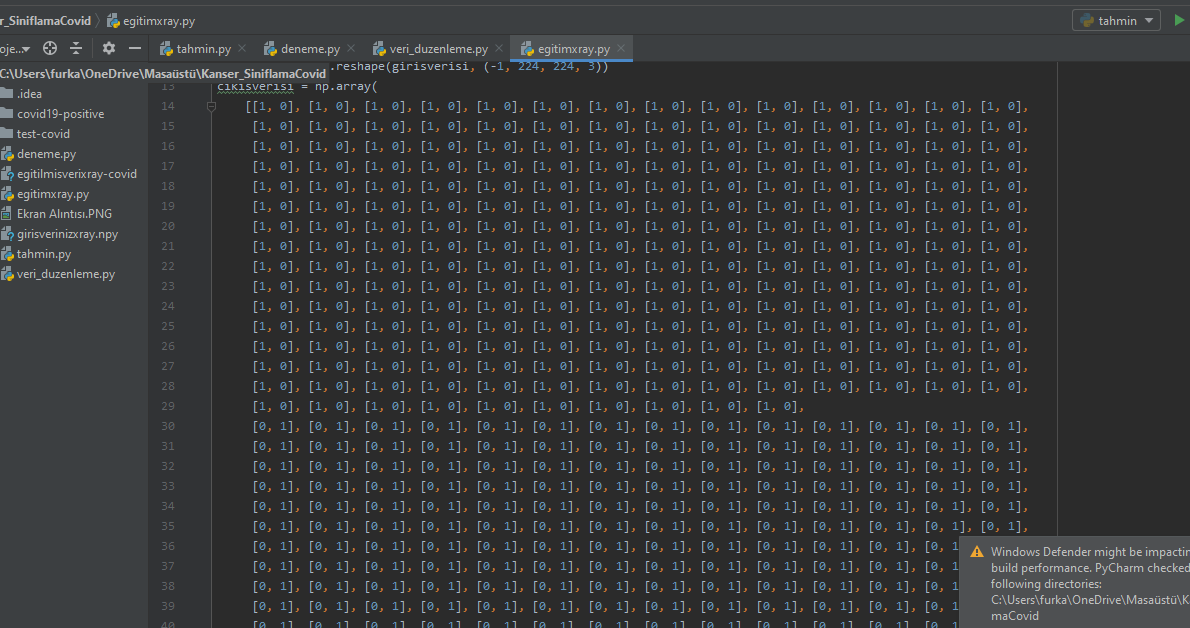
## Eğitim ve Test Seti Oluşturma

Sınıflandırma işlemi için kullanılacak verileri “.npy” uzantılı bir dosya haline getirmek için aşağıdaki işlemler uygulandı. Döngü yardımıyla verilerimizi teker teker alındı. Veriler sınıflandırma için en optimum büyüklük olan 224,224 piksel haline dönüştürüldü.



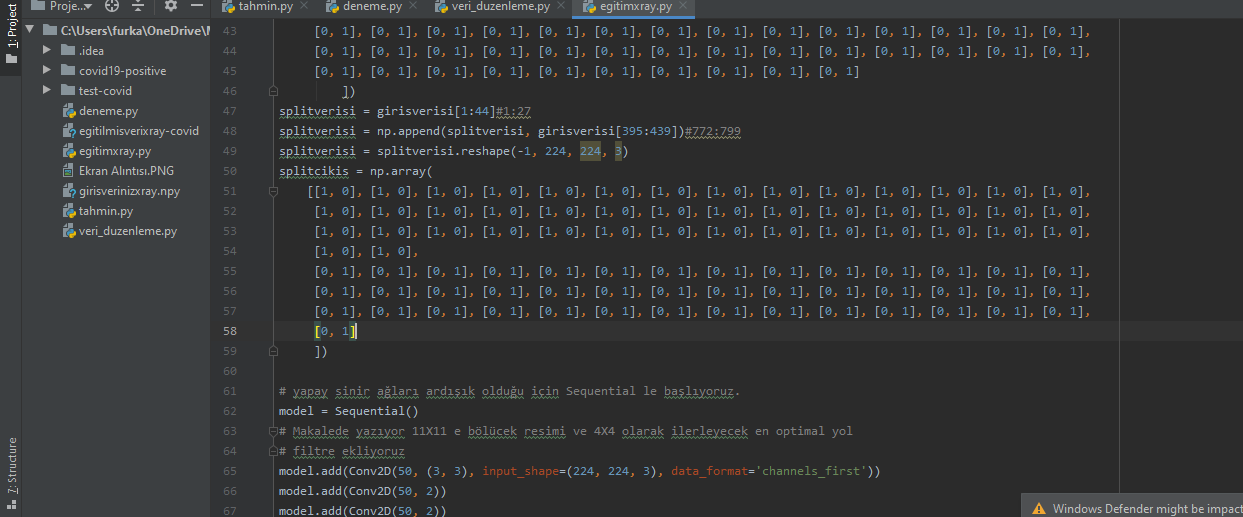
Şekil 13: Eğitim Verisi Oluşturma İşlemi

Eğitim aşamasında ilk iki yüz veriyi(Covid-19 Positive Birey), [1,0] olarak etiketlendi. Son iki yüz veri ise(Sağlıklı Birey), [0,1] olarak etiketlendi.



Şekil 14:Verilerin Etiketlenme İşlemi

Verilerin doğrulanması için gerekli olan validation değerleri aşağıdaki gibi eklenmiştir.



Şekil 15: Validation Değerleri

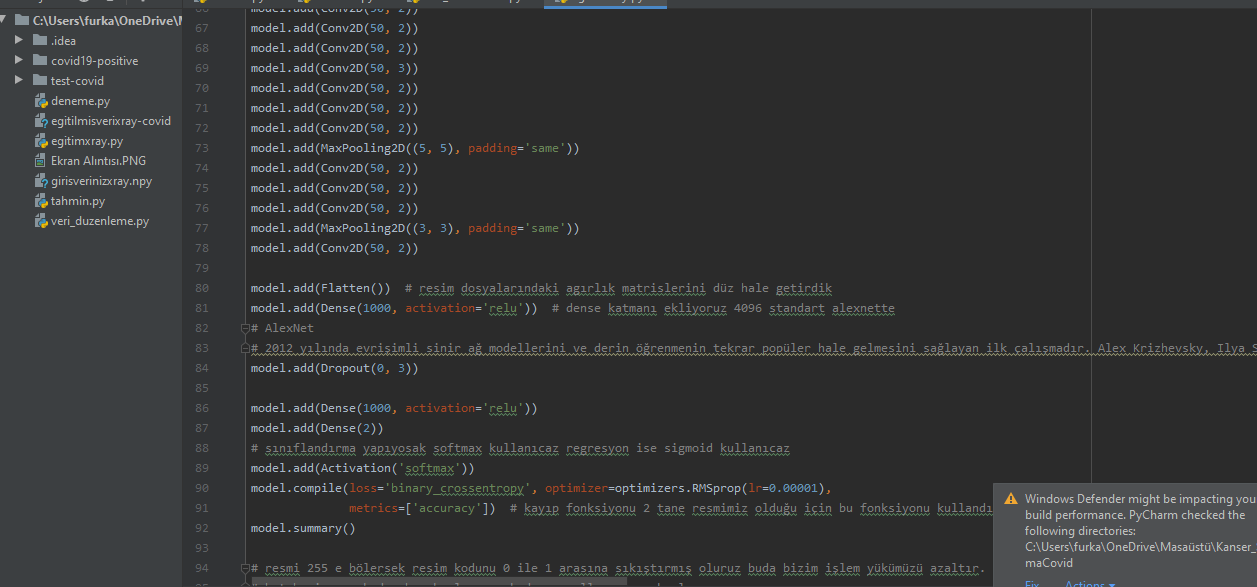
Buna göre verilerin %25 ini “splitverisi” adıyla ayrıldı. 1-44 arasındaki veriler “Covid-19 Positive Bireyler”. 396-440 arasındaki veriler ise “Sağlıklı Bireyler”.

## Model Katmanları Oluşturma

Model katmanları oluşturulurken katman sayısının çokluğu verilerin işlenmesi ve gerekli bilgi çıkarımı için önemli bir husustur. Çalışma görüntü işleme üzerine olduğu için katmanlarda Conv2D Evrişimsel sinir ağı yapısı tercih edilmiştir.

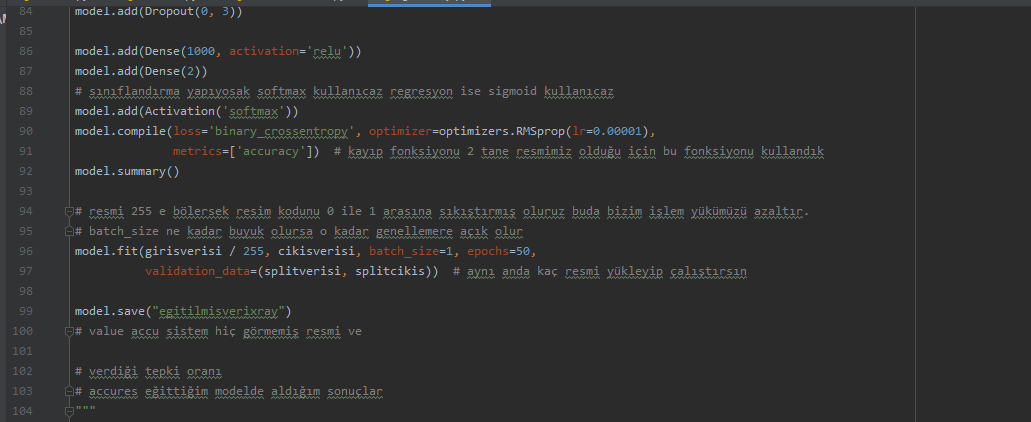
Aktivasyon fonksiyonu olarak son katmanda “sigmoid” ve “softmax”, model içerisinde ise “relu” aktivasyon fonksiyonları kullanılmıştır. Eğitim esnasında hız önemli bir değer olduğunda Relu tercih edilen fonksiyonlar arasında birinci sırayı almaktadır.

Model üzerine belli bir aşamadan sonra MaxPooling2D katmanı eklenmiştir. Bu katman model yapıları arasında daha doğru sonuca ulaşmak için verileri belli oranlarda azaltmak için kullanılmaktadır. Aynı şekilde model içerisinde belli noktalara dropout katmanı eklenerek aşırı uydurma (ezberleme) olayı ortadan kaldırılmıştır. Modelin son katmanları içerisine ise flatten (düzleştirme) katmanı eklenerek tahmin edilecek değer için veriler tek vektör yapıya dönüştürülmüştür. Tahmin edilecek 2 sınıf olduğu için en son katmanda bulunan Dense ile katman yapıları sonlandırılmıştır.



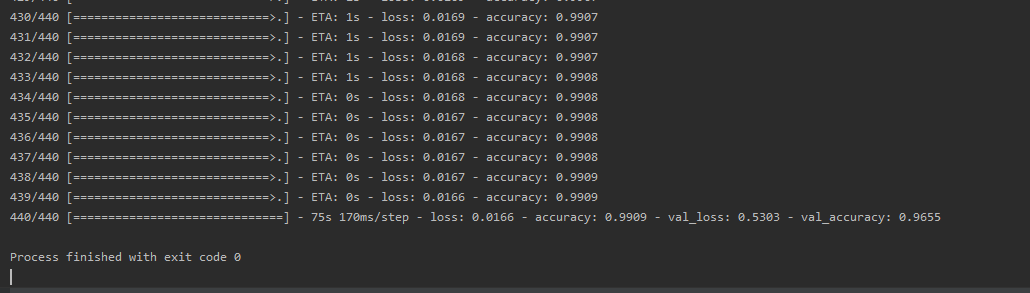
Şekil 16:Modelde Kullanılan Katmanlar

Model derleme kriteri olarak 3 argüman bulunmaktadır. Bunlardan ilki tahmin edilen değer ile doğru değer arasındaki hata oranını gösteren “loss” değeridir. İkincisi eğitimin prosedürünün ayarlandığı “optimizer” değeri, üçüncüsü ise eğitim işleminin başarısını hangi bilgiye göre gösterileceğini belirleyen metric ayarıdır. Çalışmada metric ayarı olarak “accuracy” (doğruluk), sensitivity” (duyarlılık), “specificity” (özgüllük) değerleri belirlenmiştir. Yapılan tahminlerin doğru olma durumu eğitimin başarısını göstermektedir. Çalışmada sınıflandırma etiketleri az sayıda olduğundan “loss” değeri olarak “binary\_crossentropy” tercih edilmiştir.



Şekil 17:Model Derleme Kriterleri

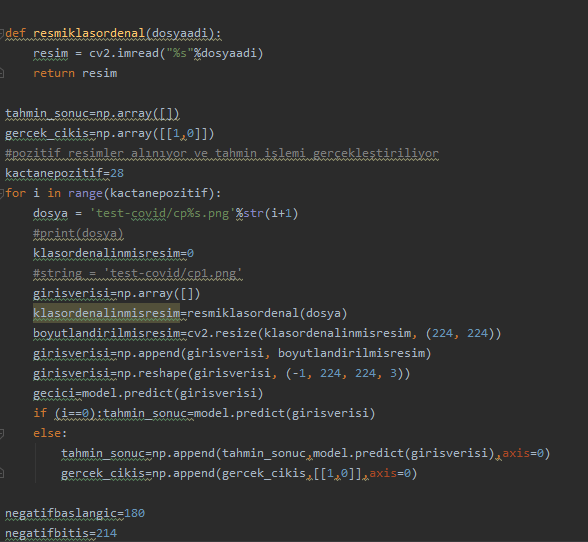
50 tekrar sonrası modelden alınan değerler bunlardır;



Şekil 18:Eğitim Aşamasında Alınan Değerler

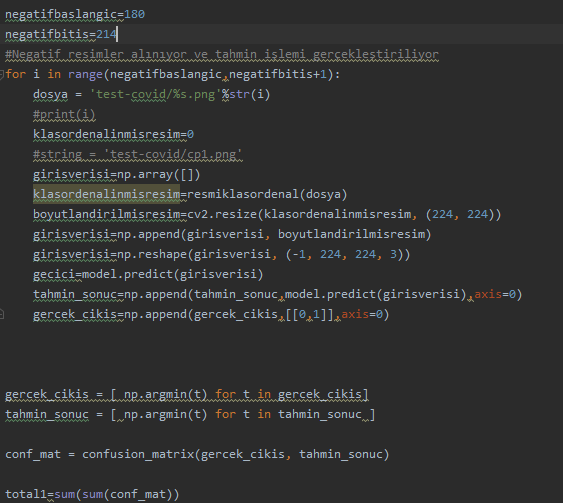
## Eğitilmiş Modeli Test Etme

“test-covid” dosyasına test için ayrılan kümeden rastgele bir değer aktarılarak bu resim için tahmin işlemi gerçekleştirilir. “reshape” komutu ile veri dosyası (1,224,224,3) şekline dönüştürülür.



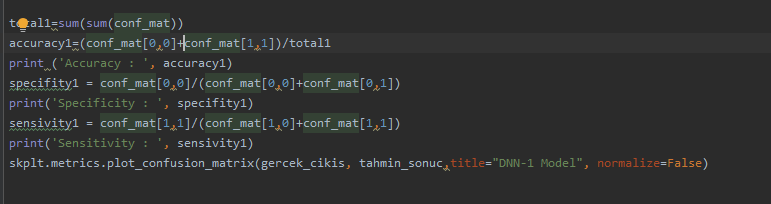
Şekil 19:Test Edilecek Covid-19 Pozitif Görsellerin Seçilmesi

Bu aşamada ise eğitim verisine katılmamış “Sağlıklı Bireylerin” görsellerini seçiyoruz.



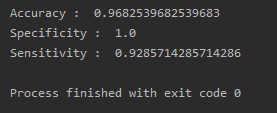
Şekil 20:Test Edilecek Hasta Olmayan Görsellerin Seçilmesi

Bu aşamada ise eğitilen modelin başarımını test etmek için gerekli kodlar yazıldı.



Şekil 21:Eğitilmiş modeli test etmek için gerekli kodlar

Modelin “Accuracy”, “Specificity”, “Sensitivity” değerleri aşağıdaki gibidir.



Şekil 22:Modelin “Accuracy”, “Specificity”, “Sensitivity” Değerleri

# **SONUÇ ve ÖNERİLER**

## Sonuç

Bu tez çalışmasında Kaggle sitesinden araştırmacıların, kullanıcılar için sunduğu veri setleri sınıflandırılmak üzerine kullanılmıştır. Derin öğrenme mimarileri araştırılmış, analiz edilmiş ve kendi veri setimiz kullanarak eğitimleri gerçekleştirilmiştir. Evrişimli Sinir Ağı mimarisine ait parametrelere bağlı olarak eğitimler gerçekleştirilmiş ve sonuçlar kaydedilmiştir. Bu uygulamalara bağlı olarak çeşitli sonuçlar elde edilmiştir. Tez çalışmasına katkıları ve sonuçları aşağıda maddeler halinde ifade edilmektedir.

Derin öğrenme sistemlerinde sonuçların başarılı olmasında veri setinin boyutu ve çeşitliliği önemlidir. Aynı zamanda eğitim tur sayısı, öğrenme oranı gibi parametre ayarları da başarının yükselmesinde rol oynamaktadır.

Modelimizde kullanılan 400 eğitim verisi ve 63 test verisiyle çıkan sonuçlara göre %96,82 oranla sınıflandırma sonuçları vermiştir.

Bu modellere ait eğitim sonucunda oluşan loss fonksiyonu sonucu incelendiğinde tamamında da kayıp değeri 0’a olabildiğince yaklaşmış ve eğitimin başarısı test sonuçlarına da yansımıştır. Modelin loss değeri 0.02 değerinin altına kadar inerek eğitimin başarısı gözlemlenmiştir.

Veri setinde kullanılan görseller, farklı kaynaklardan alındığı için ve veri sayısının düşüklüğünden dolayı modelin başarısını düşürmesinde etkisi olmuştur.

## Öneriler

Yapay zekâ ve onun alt alanı olan derin öğrenme teknolojileri büyük bir hızla gelişmeye devam etmektedir. Özellikle derin öğrenme konusunda son yıllarda pek çok başarılı çalışma gerçekleştirilmiştir. Bu tez çalışmasında kullanılan Keras kütüphanesi gibi öğrenme yapılarının başarım oranlarının daha yukarılara çıkacağı ön görülmektedir. Bu teknoloji benzer semptomlar ve belirsizliklerin hâkim olduğu zor hastalıkların teşhisinde başarılı bir şekilde kullanılabilir. Bu nedenle hastalıklarla ilgili veri setleri üzerinde yeni derin öğrenme kütüphaneleri ve metotları kullanılarak daha yüksek doğruluk oranları ile bu hastalıklar erken teşhis edilebilir. Bu tez çalışmasında elde edilen en başarılı model yapısının katmanları ilerde geliştirilerek Keras kütüphanesi üzerinden doğruluk katsayıları daha da yüksek seviyelere çıkarılabilir. Veri sayısını arttırma adına hastanelerde bulunan ilgili merkezlerden alından farklı veriler veri setine entegre edilerek daha başarılı sınıflandırma oranlarına ulaşılabilir.

# EKLER

**EK**-1 **Lisans Bitirme Çalışması için dış kapak ve iç kapak sayfası örneği**

T.C.

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ BÖLÜMÜ LİSANS BİTİRME ÇALIŞMASI

**Derin Öğrenme ve Görüntü İşleme Tekniklerini Kullanarak Covid-19 Röntgen Verilerinin Sınıflandırılması**

Hazırlayan  
Muhammed Furkan GÖKBEL

Danışman

Osman ÖZKARACA

25 Mayıs 2020

MUĞLA

# Kaynakça

|  |  |
| --- | --- |
| [1] | Z. Tan, *DERİN ÖĞRENME YARDIMIYLA ARAÇ SINIFLANDIRMA,* 2019. |
| [2] | «Covid19Bilgi,» [Çevrimiçi]. Available: https://covid19bilgi.saglik.gov.tr/tr/covid-19-yeni-koronavirus-hastaligi-nedir. [Erişildi: 17 Mayıs 2020]. |
| [3] | B. UZUN. [Çevrimiçi]. Available: https://bertanuzun.com/yapay-zeka/). [Erişildi: 18 Mayıs 2020]. |
| [4] | E. ALADAĞ. [Çevrimiçi]. Available: https://www.emrealadag.com/makine-ogrenmesi-nedir.html. [Erişildi: 12 Mayıs 2020]. |
| [5] | «Yapay Zeka Portalı,» [Çevrimiçi]. Available: yapayzeka.ai. [Erişildi: 16 Mayıs 2020]. |
| [6] | «Endüstri 4.0 Platformu,» [Çevrimiçi]. Available: https://www.endustri40.com/makine-ogrenimi-nedir/. [Erişildi: 16 Mayıs 2020]. |
| [7] | «Medium,» [Çevrimiçi]. Available: https://medium.com/t%C3%BCrkiye/makine-%C3%B6%C4%9Frenmesi-nedir-20dee450b56e. [Erişildi: 16 Mayıs 2020]. |
| [8] | O. Alkan, *PARKİNSON HASTALIĞININ TEŞHİSİNDE DERİN ÖĞRENME YÖNTEMİ İLE SPECT GÖRÜNTÜ ANALİZİ,* 2019. |
| [9] | «MC.AI,» [Çevrimiçi]. Available: https://mc.ai/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir/. [Erişildi: 15 Mayıs 2020]. |
| [10] | G. GÜNDÜZ ve İ. H. CEDİMOĞLU, *Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini,* 2019. |
| [11] | M. Bilgin, *Makine Öğrenmesi,* İstanbul: Papatya Yayıncılık, 2018. |
| [12] | N. Gürsakal, *Makine Öğrenmeesi ve Derin Öğrenme,* Bursa: Dora Yayıncılık, 2017. |
| [13] | M. KOPTUR. [Çevrimiçi]. Available: https://makineogrenimi.wordpress.com/2017/05/26/siniflandirma-classification/. [Erişildi: 18 Mayıs 2020]. |
| [14] | «ANALYTİCHOUSE VERİ ANALİZ VE DANIŞMANLIK,» [Çevrimiçi]. Available: https://www.analytichouse.com/regresyon/. [Erişildi: 18 Mayıs 2020]. |
| [15] | M. A. KIZRAK ve B. BOLAT, *Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma,* İstanbul, 2018. |
| [16] | M. PİŞKİN, 14 Mayıs 2020. [Çevrimiçi]. Available: http://mesutpiskin.com/blog/wp-content/uploads/2017/01/OpenCV%20Kitap.pdf. [Erişildi: 19 Mayıs 2020]. |
| [17] | İ. AYDOĞAN ve G. ZIRHLIOĞLU, *Öğrenci Başarılarının Yapay Sinir Ağları ile Kestirilmesi,* 2018. |
| [18] | K. FIRILDAK ve M. F. TALU, *Evrişimsel Sinir Ağlarında Kullanılan Transfer Öğrenme,* 2019. |
| [19] | T. ERGİN, «Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?,» [Çevrimiçi]. Available: https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad. [Erişildi: 20 Mayıs 2020]. |
| [20] | W. H. Khoong, «Kaggle,» [Çevrimiçi]. Available: https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets. [Erişildi: 8 Mayıs 2020]. |
| [21] | P. Mooney, «Kaggle,» [Çevrimiçi]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. [Erişildi: 9 Mayıs 2020]. |
| [22] | D. kp, «Kaggle,» [Çevrimiçi]. Available: https://www.kaggle.com/imdevskp/corona-virus-report. [Erişildi: 5 Mayıs 2020]. |
| [23] | bachir, «Kaggle,» [Çevrimiçi]. Available: https://www.kaggle.com/bachrr/covid-chest-xray. [Erişildi: 10 Mayıs 2020]. |
| [24] | [Çevrimiçi]. Available: yapayzeka.ai. [Erişildi: 15 Mayıs 2020]. |