**ÖNSÖZ**

Bu çalışma boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocamız Sayın Yrd. Doç. Dr. Tolga ENSARİ’ye en içten dileklerimizle teşekkür ediyoruz.

Ayrıca tüm eğitim hayatımız boyunca bize yol gösteren, destek olan, en önemlisi bize eğitimin ne kadar önemli olduğu bilincini ve çalışma disiplinimi kazandıran , her zaman yanımızda olan ailemize en içten dileklerimizle teşekkürlerimizi sunuyoruz.

[Mehmet ONAR] [1358130107]

[Kübra ATMACA] [1358130101]

İçindekiler

[Şekil Listesi iv](#_Toc501360432)

[TABLO ve grafik LİSTESİ iv](#_Toc501360433)

[Özet v](#_Toc501360434)

[summary 1](#_Toc501360435)

[1. GİRİŞ 2](#_Toc501360436)

[2. GENEL KISIMLAR 2](#_Toc501360437)

[2.1. DERİN ÖĞRENME 2](#_Toc501360438)

[2.2. MODELİN EĞİTİLMESİ 3](#_Toc501360439)

[2.2.1. Dentdrites (Girdilerin Sonlu Kümesi) 4](#_Toc501360440)

[2.2.2. Snaptic Weights (Snaptik Ağırlıklar) 4](#_Toc501360441)

[2.2.3. Axon (Girdilerin Toplamı) 5](#_Toc501360442)

[2.2.3.1 Toplama Fonksiyonları 5](#_Toc501360443)

[2.2.4. Activation Function (Aktivasyon Fonksiyonu) 5](#_Toc501360444)

[2.3. Modelin OLUŞTURULMASI 8](#_Toc501360445)

[2.3.1. Denetlenmiş (Supervised) 8](#_Toc501360446)

[2.3.2. Denetlenmemiş (Unsupervised) 9](#_Toc501360447)

[2.4. Derin öğrenme ile ilgili yapılan çalışmalar 11](#_Toc501360448)

[2.4.1. Yüz Tanıma Sistemi 14](#_Toc501360449)

[2.4.2 Konuşma Tanıma Sistemi 15](#_Toc501360450)

[2.5. SAVUNMA VE GÜVENLİK SEKTÖRÜNDE KULLANIMI 17](#_Toc501360451)

[2.6. MAKİNE ÖĞRENMESİ 18](#_Toc501360452)

[2.6.1 Denetimli Öğrenme 18](#_Toc501360453)

[2.6.2 Denetimsiz Öğrenme 18](#_Toc501360454)

[2.6.3  Yarı Denetimli Öğrenme 19](#_Toc501360455)

[2.6.4  Takviyeli Öğrenme 19](#_Toc501360456)

[2.6.5  Yoğun Öğrenme 19](#_Toc501360457)

[2.7.YAPAY SİNİR AĞLARI 19](#_Toc501360458)

[2.8.DERİN ÖĞRENMEDE KULLANILAN KÜTÜPHANELER 20](#_Toc501360459)

[2.8.1 Torch 20](#_Toc501360460)

[2.8.2 Caffe 21](#_Toc501360461)

[2.8.3 Theano 23](#_Toc501360462)

[2.8.4 Image-Net 24](#_Toc501360463)

[2.8.5 Alexnet (Matlab) 24](#_Toc501360464)

[2.9 YÖNTEM-UYGULAMA 24](#_Toc501360465)

[2.10 MATLAB GELİŞTİRME ORTAMINDA UYGULAMANIN GELİŞTİRİLMESİ 24](#_Toc501360466)

[2.11. ANDROID İLE UYGULAMANIN GELİŞTİRİLMESİ 28](#_Toc501360467)

[2.12.TEST-SONUÇ 30](#_Toc501360468)

[2.12.1 Matlab Geliştirme Ortamında Uygulamanın Test Edilmesi 30](#_Toc501360469)

[2.12.2 Andorid Uygulamasının Test Edilmesi 31](#_Toc501360470)

[kaynaklar 32](#_Toc501360471)

[EKLER 33](#_Toc501360472)

[ÖZGEÇMİŞ 34](#_Toc501360473)

# Şekil Listesi

[Şekil 1 : Yapay Sinir Ağı Modeli 4](#_Toc245783558)

[Şekil 2 : Nöronun Yapısı](#_Toc245783559) 6

[Şekil 3 : Derin Öğrenme Yüz Tespiti Yapısı 8](#_Toc245783560)

[Şekil 4 : Denetlenmiş Öğrenme 9](#_Toc245783560)

[Şekil 5 : Denetlenmemiş Öğrenme 10](#_Toc245783560)

[Şekil 6 : Ağaç Dalına Tünemiş Kuş 11](#_Toc245783560)

[Şekil 7 : Siyah Beyaz Köpek Bar Üzerinden Atlıyor 12](#_Toc245783560)

[Şekil 8 : Pembe Kıyafetli Kız Havada Zıplıyor 12](#_Toc245783560)

[Şekil 9 : Mavi Dalış Kıyafetli Adam Dalga Üstünde Sörf Yapıyor 12](#_Toc245783560)

[Şekil 10 : İki Çocuk Lego Oyuncağı Oynuyor 13](#_Toc245783560)

[Şekil 11 : Google DeepMind Atari Çalışması 13](#_Toc245783560)

[Şekil 12 : Video Akışında Tüm Görüntü Üzerinde Eş Zamanlı Nesne Tespiti Yapılması 14](#_Toc245783560)

[Şekil 13 : Yüz T. Veri Setinde Derin Öğrenme Yöntemiyle Firmaların Yakaladığı Oranlar 15](#_Toc245783560)

[Şekil 14 : Derin Öğrenme İle Konuşma Tanımanın Yapılması 16](#_Toc245783560)

[Şekil 15 : Derin Öğrenme İle Firmaların Konuşma Tanımadaki Hata Oranları 16](#_Toc245783560)

[Şekil 16 : Yeni Nesil Avuç İçine Sığabilen Üç Boyutlu Yüksek Çözünürlüklü Kameralar 17](#_Toc245783560)

[Şekil 17 : Makine Öğrenmesine Ait Başlıca Kavramlar 18](#_Toc245783560)

[Şekil 18 : Neural Network Toolbox 25](#_Toc245783560)

[Şekil 19 : Webcam Suppot Package İnstaller 26](#_Toc245783560)

[Şekil 20 : Kamera Objesi 27](#_Toc245783560)

[Şekil 21 : Sonuç 28](#_Toc245783560)

[Şekil 22 : Sonuç 30](#_Toc245783560)

# TABLO ve grafik LİSTESİ

[Tablo 1 : Bazı Toplama Fonksiyonları 23](#_Toc245783586)

[Tablo 2 : Aktivasyon Fonksiyonları 7](#_Toc245783586)

[Tablo 3 : Uygulamadan Alınan Görüntüler 30](#_Toc245783586)

[Tablo 4 : Uygulamadan Alınan Görüntüler 30](#_Toc245783586)

[Grafik1 : 24](#_Toc245783587)

# Özet

**DERİN ÖĞRENME İLE NESNE TESPİTİ**

İnsanlığın var oluşu ile başlayan bilgi birikimi son yıllarda hızlı bir ivme ile artmaktadır. Son yüzyılda yaşanan teknoloji ve tıp alanında gelişmeler, dünya dışı keşiflerin yapılması, facebook, twitter vb. uygulamalar ile elde edilen veriler, uydu taramaları ve diğer şekilde elde edilen veriler devasal bir boyuta ulaşmıştır. Önceleri verilerin analizi klasik yöntemler ile yapılıyordu. Baştan aşağı inceleme ile eldeki az verinin incelenmesi sorun değildi. Savaşta girilen parametreler ile top namlusunun açısını hesaplama ile başlayan analiz için karmaşık algoritmalara gerek duyulmaz iken son yıllarda yaşanan veri deposu taşmaları ile bir çok yöntemi de beraberinde getirmektedir. Modern anlamada veri analizi için birçok alanda çalışmalar bulunmaktadır. Makine öğrenmesi ve yapay sinir ağları bu verilerin incelenmesi ve çıkarıma varılması için çalışmalar yapılan alanlardandır. Ancak son yıllarda artan veri yükü modern anlamda incelemenin de üstüne çıkmaktadır. Artık makinelerin işlem güçleri eldeki verilerin analizini yapmakta zorlanmakta geç çözümler üretmektedir. Google, Microsoft ve Imagenet gibi firmalar verilerin analizi için yöntem geliştirerek derin öğrenme metotlarını geliştirmiş bu alanla ilgili çalışmalar başlatmıştır.

Derin öğrenme makine öğrenmesinin bir türü olup, çok katmanlı bir yapay sinir ağıdır.Yani makine öğrenmesi ile yapay sinir ağlarının ortak alanı denilebilir. Derin öğrenme ile işlem yapma yetenekleri ve yapay ağ modellerinin gelişmişliği kullanılarak birçok alanda veri analizi mevcuttur. Yapay sinir ağları da makine öğrenmesi gibi eğitilerek doğru sonuçlar elde edilebilir. Ancak ev kullanıcıları için böyle bir eğitim işlemci gücü açısından mümkün olmayacağından eğitilmiş veri setlerini alarak test veri setlerini kullanmaları mümkün olacaktır. Derin öğrenme ile ilgili teknoloji firmalarının belirli kütüphaneleri mevcut olup bu kütüphaneler açık kodlu olarak yayınlanmaktadır. Derin kütüphaneleri NVIDIA DIGITS, Theano, Caffe ve Torch’dir. Deeplearning birden fazla alanda kullanılır iken bu dönem ki bitirme tezimizde nesne tespiti konusunda çalışmamızda kullanacağımız bir yöntem olacak.

# summary

**OBJECT DETERMINATION WITH DEEP LEARNING**

The accumulation of knowledge that began with the presence of humanity has been accelerating in recent years. Developments in technology and medicine in the last century, specialists from extraterrestrial discoveries, facebook, twitter etc. Data obtained by applications, satellite scans and other forms of data obtained. It was done with classical methods. It was not a problem to study the little data at hand from scratch. While there are no complicated algorithms to start calculating with the parameters entered in the war, many methods are brought together with the data storage stones that have been experienced in recent years. In modern sense, there are many field work for the database. Machine learning and construction neural networks are the areas of study and extrapolation of these data. However, in recent years, the incremental data load has also surpassed that of modern research. Now, the machining powers of the machine are difficult to analyze your data and produce late solutions. Google has initiated work on this enhanced field, such as Microsoft and Imagenet.

Deep learning is a solid, multi-layered artificial neural network of machine learning.So machine learning can be called the common area of ​​artificial neural networks. Deep learning and processing abilities and sophistication of artificial networking models. Artificial neural networks. As you train, you can get accurate results. However, if users have some training capability, they can use trained data sets. There are certain libraries of technology firms involved in learning deeply, or these libraries are open-coded. Deep libraries for NVIDIA DIGITS, Theano, Caffe and Torch. When Deeplearning is more than one area, it will be a way to finish this period and use it in our work somewhere.

# 1. GİRİŞ

"Yapay zekâ" kavramının geçmişi modern [bilgisayar](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQmlsZ2lzYXlhcg) bilimi kadar eskidir. Fikir babası, "Makineler düşünebilir mi?" sorunsalını ortaya atarak [makine zekâsını](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTWFraW5lX3playVDMyVBMnMlQzQlQjE) tartışmaya açan [Alan Mathison Turing](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQWxhbl9NYXRoaXNvbl9UdXJpbmc)'dir. [1943](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvMTk0Mw)'te [II. Dünya Savaşı](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvSUkuX0QlQzMlQkNueWFfU2F2YSVDNSU5RiVDNCVCMQ) sırasında [Kripto](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvS3JpcHRvbG9qaQ) analizi gereksinimleri ile üretilen [elektromekanik](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvRWxla3Ryb21la2FuaWs) cihazlar sayesinde [bilgisayar bilimi](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQmlsZ2lzYXlhcl9iaWxpbWk) ve yapay zekâ kavramları doğmuştur.[1] yapay zekâ kendisi ile birlikte birçok alt alana bölünmüştür. Bunlar yapay zekânın alt dallarıdır.

* [Makine Zekâsı](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPU1ha2luZV9aZWslQzMlQTJzJUM0JUIxJmFjdGlvbj1lZGl0JnJlZGxpbms9MQ) (Sembolik Yapay Zekâ)
* [Yapay Sinir Ağları](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvWWFwYXlfU2luaXJfQSVDNCU5RmxhciVDNCVCMQ) ([Sibernetik](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvU2liZXJuZXRpaw) Yapay Zekâ)
* [Doğal Dil işleme](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvRG8lQzQlOUZhbF9EaWxfaSVDNSU5RmxlbWU) (Dil ile düşünme)
* [Konuşma Sentezi](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvS29udSVDNSU5Rm1hX1NlbnRlemk) (Yapay Konuşma)
* [Konuşma Anlama](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUtvbnUlQzUlOUZtYV9BbmxhbWEmYWN0aW9uPWVkaXQmcmVkbGluaz0x) (Konuşma Analizi)
* [Uzman sistemler](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvVXptYW5fc2lzdGVtbGVy)
* [Örüntü Tanıma](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvJUMzJTk2ciVDMyVCQ250JUMzJUJDX1RhbiVDNCVCMW1h)
* [Genetik Algoritmalar](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvR2VuZXRpa19BbGdvcml0bWFsYXI)
* [Genetik Programlama](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvR2VuZXRpa19Qcm9ncmFtbGFtYQ)
* [Bulanık Mantık](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQnVsYW4lQzQlQjFrX01hbnQlQzQlQjFr)
* [Çoklu Örnekle Öğrenme (Multiple Instance Learning)](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPSVDMyU4N29rbHVfJUMzJTk2cm5la2xlXyVDMyU5NiVDNCU5RnJlbm1lKE11bHRpcGxlX0luc3RhbmNlX0xlYXJuaW5nKSZhY3Rpb249ZWRpdCZyZWRsaW5rPTE)

Bitirme tezimizin konusu olması sebebi ile yukarıda ki alt alanlardan Yapay Sinir Ağları ve makine öğrenmesi alt alanları ile ilgileneceğiz. Çünkü derin öğrenme konusu literatürde doğrudan yapay zekânın alt alanlarında gösterilmemektedir.

# 2. GENEL KISIMLAR

## 2.1. DERİN ÖĞRENME

Derin öğrenme yapay zekâ (AI) alt dallarından makine öğrenmesine yapay sinir ağlarının ilgilendiği bir alandır. Son yıllarda kullanım alanlarına bağlı olarak popülerliği gittikçe artmaktadır. Derin öğrenmenin tanımı “Derin öğrenme makine öğrenmesinin bir türü olup, çok katmanlı bir yapay sinir ağı.” Şeklinde yapılabilir. Tanımın tam anlaşılabilmesi için makine öğrenmesi ve yapay sinir ağının tanımlarını bilmek gerekir. Metnin başında dediğimiz gibi derin öğrenme konusunu bu kadar meşhur yapan en önemli konu birçok alanda kullanılması ve parametrelere bağlı üretilen sonucun klasik sınıflandırma algoritmaları ve klasik analiz yöntemlerine oranla daha yüksek olmasıdır. Derin öğrenme başta savunma sanayi olmak üzere endüstriyel ve diğer tüm alanlarda kullanılması mümkündür. Kumaş üreten bir fabrikada hatalı kumaşların tespitinden, bölgesel görüntü alan bir insansız hava aracının elde ettiği görüntülerde olağan dışılığın tespitine kadar kullanım yelpazesi geniştir. Derin öğrenme alanı günümüzde görüntüler üzerinden analiz yapılmasında fazlalıkla kullanılsa da matris normuna dönüştürebilinen her şeyi analiz edebilir, sonuçlar üretebilir. Bunlar; ses, görüntü, borsa verileri vb. şeklinde artırılabilir.

Derin öğrenme verilerin analizini kolaylaştırsa da modelinin oluşturulması bir hayli zordur. Derin öğrenmede analizi yapılacak ses ya da görüntü verilerinin analiz edilmesi için yapay sinir ağları ve makine öğrenmesi alanlarının çok iyi tanınması gerekmektedir. Derin öğrenme modeli yapay sinir ağı modeli ile benzerlik göstermektedir. Ancak yapay sinir ağı katmanına göre fazlaca gizli katmana (hidden layer) sahiptir. Fazlaca gizli katmanın oluşu modelin eğitilmesini de zorlaştırmaktadır. Bu zorluğu anlamak için şöyle bir örnek vermek gerekebilir. Makalenin ilerleyen bölümlerinde anlatacağımız derin öğrenme kütüphanelerinden olan Image Net oluşturduğu modelini14,197,122 adet görüntü ile eğitmiştir. Bu sayıyı bize anlamlı gelmesi için şöyle bir ifadeye yer vermemiz gerekiyor. İmage-net bir görüntüde bir papatya olup olmadığının tespiti için 14,197,122 adet papatya fotoğrafının özniteliğini çıkartıyor. Yukarıda belirttiğimiz sayı gittikçe artmaktadır. Bunun sebebi ise ne kadar çok görüntü incelenirse doğruluk oranı o kadar yukarıya çıkacaktır.

### 

## 2.2. MODELİN EĞİTİLMESİ

Derin öğrenme yapay sinir ağı modelinin zorluğundan daha önce bahsetmiştik ancak bitirme tezimizde olması açısından modelin nasıl oluşturulacağından kısaca bahsetmemiz gerekecek. Bunun için ilk önce yapay sinir ağı modelindeki elemanları tanımlamamız gerekiyor. Bir yapar sinir ağı elamanları aşağıdaki yapay nöronlardan oluşur. Bu nöronların birbirine girdi ya da çıktı olarak bağlanması sonucu yapay sinir ağı oluşur.



Şekil Yapay Sinir Ağı Modeli

Şekil 1 ile gösterilmiş her bir daire bir nöronu temsil etmektedir. Giriş seviyesindeki nöronlar verileri alır snaptic ağırlıkları hesaplandıktan sonra çıktıları gönderilir. Giriş seviyesindeki nöronların çıktıları gizli katmandaki nöronların girişleri olarak belirlenmiştir. Yine burada hesaplanan değerler çıktı seviyesindeki nöronların giriş parametre değerleri olması için diğer nöronlara gönderilir. Yapının temeli bu şekilde oluşur. Burada sıklıkla değindiğimiz nöronlar ise aşağıda belirttiğimiz 4 temel bileşenden oluşur.

### 2.2.1. Dentdrites (Girdilerin Sonlu Kümesi)

Girdiler nöronlara gelen verilerdir. Girdiler yapay sinir hücresine bir diğer hücreden gelebileceği gibi direk olarak dış dünyadan da gelebilir. Bu girdilerden gelen veriler biyolojik sinir hücrelerinde olduğu gibi toplanmak üzere nöron çekirdeğine gönderilir.

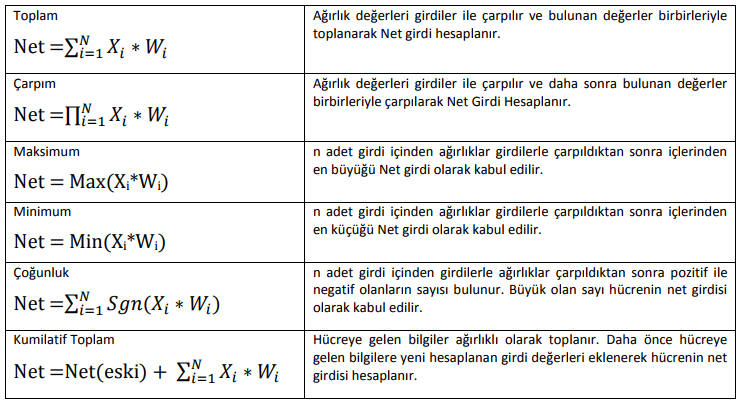
### 2.2.2. Snaptic Weights (Snaptik Ağırlıklar)

Yapay sinir hücresine gelen bilgiler girdiler üzerinden çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilinmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir. Ağırlığı sıfır olan girdilerin çıktı üzerinde herhangi bir etkisi olmamaktadır.

### 2.2.3. Axon (Girdilerin Toplamı)

Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur. Bazı durumlarda gelen girdilerin değeri dikkate alınırken bazı durumlarda ise gelen girdilerin sayısı önemli olabilmektedir. Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Genellikle deneme yanılma yoluyla toplama fonksiyonu belirlenmektedir. Bazen her hücrenin toplama fonksiyonunun aynı olması gerekmez. Bu konulara karar vermek tasarımcıya aittir.

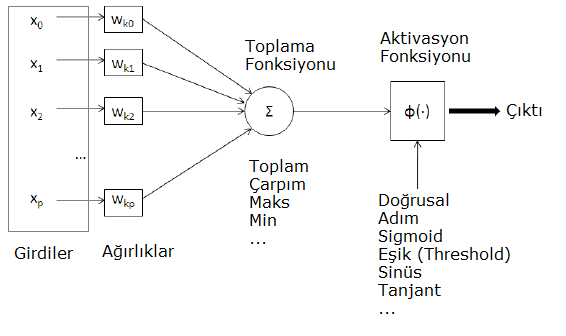
### 2.2.3.1 Toplama Fonksiyonları



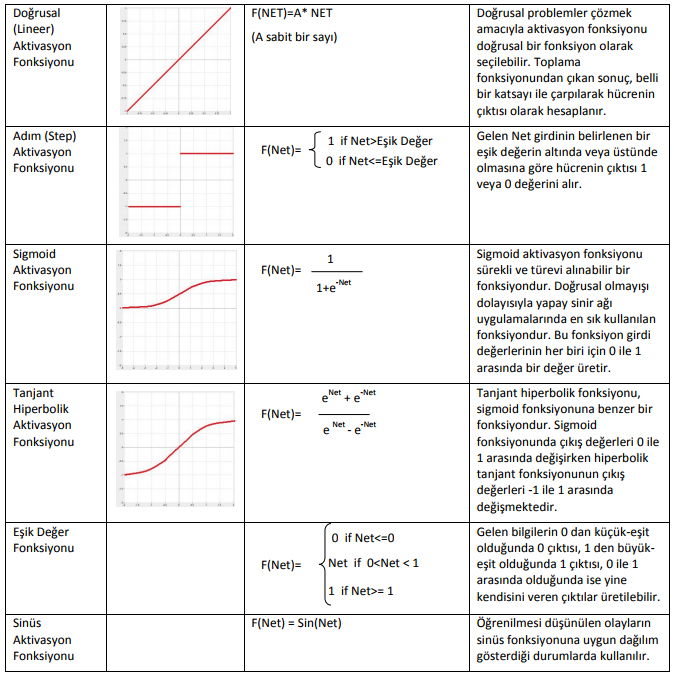
Tablo 1. Bazı Toplama Fonksiyonları

### 2.2.4. Activation Function (Aktivasyon Fonksiyonu)

Bu fonksiyon hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. Yapay sinir ağlarının bir özelliği olan “doğrusal olmama” aktivasyon fonksiyonlarının doğrusal olmama özelliğinden gelmektedir. Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamanın yavaşlamaması için türevi kolay hesaplanır bir fonksiyon seçilir. Günümüzde en yaygın olarak kullanılan “Çok katmanlı algılayıcı” modelinde genel olarak aktivasyon fonksiyonu olarak “Sigmoid fonksiyonu” kullanılır.



Şekil Nöronun Yapısı



Tablo 2. Aktivasyon Fonksiyonları

Yukarıda Şekil 2 de nöronun yapısı gösterilmiştir. Oluşturulacak olan model hangi amaca hizmet edecek ise yapay sinir ağımızı modelimizi ona göre geliştirmemiz gerekmektedir. Ağı oluşturmadan önce girişe etki edecek tüm giriş değerlerini işlemlere dâhil etmemiş gerekmektedir. Sistemin eğitilmesi içinde ağırlıkların belirlenmesi ile oluşacaktır.

Derin öğrenmede bir görüntüdeki yüzleri tespit etmek istiyorsak modelimizi ona göre geliştirmemiz gerekmektedir. Şekil 3 ile belirtilmiş bir derin öğrenme yapısında modelin oluşumunu şekillendirmek yapıyı özetlemek için oluşturulmuştur.



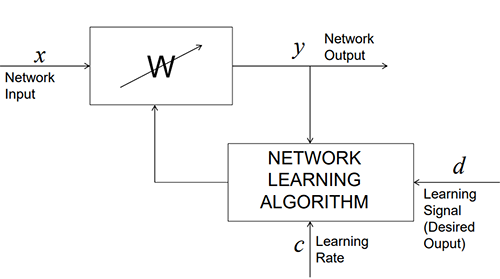
Şekil Derin Öğrenme Yüz Tespiti Yapısı[2]

## 2.3. Modelin OLUŞTURULMASI

Bir yapay sinir ağının eğitilmesinin ne kadar zor olduğuna değinmiştik. Bir görüntüdeki objenin tespit edilmesi için ne kadar görüntü tarandığını da yukarıda değindik. Yapay sinir ağı modelinin eğitilmesi demek; oluşturduğumuz ağdaki snaptik ağırlıkların ayarlanması anlamına gelir. Ancak bu ağırlıkların ayarlanması ise gelişi güzellikten ziyade daha önceden eğitilmiş veri kümelerinin kullanılması ile ayarlanabilir. Eğitilmiş veri kümesinde girdiye karşılık alınacak değerler mevcuttur örneğin bir görüntünün matris formundaki değerleri ağımızın girdisi olsun bunun karşılığında görüntüde yüz tespinin yapıldığı durumda çıktı kümesi eğitilmiş veri setinin içerisindedir. Eğer biz içerisinde yüz olan bir resim gönderdiğimiz zaman eğitilmiş veri seti ile aynı değerleri alamadıysak ağımız henüz eğitilmemiş demektir. Binlerce görüntü ile bu modeli test edip snaptik ağırlıkları ayarlamamız gerekmektedir. Ancak bu şekilde ağımıza öğrendi diyebiliriz. Yapay sinir ağının eğitilmesi için birden fazla algoritma mevcuttur bu algoritmalar temel olarak denetlenen(supervised) ve denetlenmeyen (unsupervised) algoritmalar olarak ayrılmıştır.

### 2.3.1. Denetlenmiş (Supervised)

Danışmanlı öğrenme sırasında ağa verilen giriş değerleri için çıktı değerleri de verilir. Ağ verilen girdiler için istenen çıkışları oluşturabilmek için kendi ağırlıklarını günceller. Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplanarak ağın yeni ağırlıkları bu hata payına göre düzenlenir. Hata payı hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır ve bu farka göre her hücreye düşen hata payı bulunur. Daha sonra her hücrenin kendine gelen ağırlıkları günceller.



Şekil 4: Denetlenmiş Öğrenme

Öğrenme kuralı, uygun ağ davranışı setinin eğitim seti ile sağlanır.

,,..,, input to the network

corresponding correct desired output

Girişler ağa uygulandığında, ağ çıktıları istenen çıktılarla karşılaştırılır.

i’ninci nöron için genel kurallar

Ağırlık vektörü;

Öğrenme sinyali; r = f()

Genel kurala göre t zamanındaki öğrenme aşamasında üretilen ağırlık vektörünün , (Δ) artışı Δ(t)=c.r[(t),x(t),(t)].x(t)

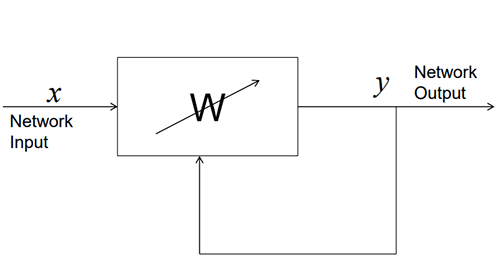
Öğrenme hızı c>0

Adapte edilmiş ağırlık vektörü (t+1)=(t)+ Δ(t)

Genel Discrete-Time Form

### 2.3.2. Denetlenmemiş (Unsupervised)

Danışmasız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar.



Şekil 5: Denetlenmemiş Öğrenme

Öğrenme sinyali mevcut değil.

i’ninci nöron için genel kurallar

Ağırlık Vektörü; Öğrenme Sinyali; r=f()

Genel kurala göre t zamanındaki öğrenme aşamasında üretilen ağırlık vektörünün , (Δ) artışı Δ(t)=c.r[(t),x(t)].x(t)

Öğrenme hızı c>0

Adapte edilmiş ağırlık vektörü (t+1)=(t)+ Δ(t)

Genel Discrete-Time Form

## 2.4. Derin öğrenme ile ilgili yapılan çalışmalar



Şekil 6: Ağaç Dalına Tünemiş Kuş

Bir ağaç dalına tünemiş kuş, derin öğrenme kullanılarak üretilmiştir.

[Stanford Üniversitesi’nden öncü araştırmacı Andrej Karpathy](http://cs.stanford.edu/people/karpathy/deepimagesent/); çalışmalarında birisi resim tanıma diğeri doğal dil işleme olmak üzere iki sinir ağını birleştirmiştir. Bu sayde tıpkı LEGO’ların birleştirildiği gibi sinir ağları sadece örnek resimdeki objeyi kuş veya ağaç olarak sınıflandırmakla kalmayıp ayrıca resim içerisindeki tüm nesnelerin birbiriyle olan ilişkisini ortaya koyabilmiştir.

Günümüzde artan kamera sayısı dikkate alındığında, görüntü içindeki nesnelerin birbiriyle olan ilişkisinin bir insan gibi makineler tarafından anlamlı bir şekilde ortaya konması görüntüleri yorumlama konusunda kullanıcılara inanılmaz bir farkındalık katmıştır. Bu sayede yüzlerce görüntü akışı (video) makineler tarafından insan nesne tanıma seviyesinin üzerinde bir başarıyla değerlendirilmektedir.

Aşağıda resimlerdeki açıklamalar derin öğrenme ile elde edilmiştir.



Siyah beyaz köpek bar üzerinden atlıyor

Şekil 7: Siyah Beyaz Köpek Bar Üzerinden Atlıyor



Pembe kıyafetli kız havada zıplıyor.

Şekil 8: Pembe Kıyafetli Kız Havada Zıplıyor



Mavi dalış kıyafetli adam dalga üstünde sörf yapıyor.

Şekil 9: Mavi Dalış Kıyafetli Adam Dalga Üstünde Sörf Yapıyor



İki çocuk lego oyuncağı oynuyor.

Şekil 10: İki Çocuk Lego Oyuncağı Oynuyor.

[DARPA](http://www.homelandsecuritynewswire.com/darpa-seeks-deep-learning-ai-cope-flood-information), insansız hava araçlarının düşman toprakları üzerinde elde ettiği görüntü ve videoların karargâha aktarımıyla oluşturulan büyük veri (BigData) yığınıyla baş edebilmek maksadıyla daha iyi bir istihbarat katmanı geliştirilmesi kapsamında 2009 yılında derin öğrenme çalışmalarına destek vermeye başlamıştır.

[DARPA doğal dil işleme için kendi derin öğrenme projesi üzerinde çalışıyor.](https://gigaom.com/2014/05/02/darpa-is-working-on-its-own-deep-learning-project-for-natural-language-processing/)

[](http://www.wired.co.uk/news/archive/2015-02/25/google-deepmind-atari)

[Google DeepMind Atari Çalışmas](http://www.wired.co.uk/news/archive/2015-02/25/google-deepmind-atari)ı

Şekil 11: Google DeepMind Atari Çalışması

Google son dönemde bünyesine kattığı, [DeepMind](http://deepmind.com/) firması ile yürüttüğü derin öğrenme çalışmaları kapsamında, Atari video oyunlarını kullanılarak makineler için sadece ağı eğitmekle kalmayıp, ayrıca ortam içerisinde nasıl hareket edileceğini de öğretmişlerdir. Bu sayede eğitilen ağ oyun serisini başarıyla tamamlamıştır. Bu çalışma ile Google sahip olduğu ve sürekli artan veri havuzunu zamanı geldiğinde geliştirdiği algoritmalar ile kullanarak akıllı sistemler ortaya çıkarabileceğini göstermiştir. ([DeepMind Yayınları](http://deepmind.com/publications.html)) ([Google neden 400 Milyon Dolara DeepMindfimasını satın aldı?](http://www.technologyreview.com/news/524026/is-google-cornering-the-market-on-deep-learning/))

Resim veya vide akışındaki her kare için çeşitli filtrelemeler ve alt bölümlere ayırma gibi işlemler sonucunda esim karesi üzerinde tespit edilen nesneler önceden eğitilmiş ağa sokularak sınıflandırılmaktadır.

Araştırmacılar akıllı telefonların ve diğer mobil cihazların kamera görüş hattında bulunan nesneleri hemen tanıyıp, nesneleri tanımlayan metinleri nesnelerin üzerinde bir çevre katman olarak gösterecek şekilde çalışma yapmaktadır (PurdueUniversityimage/e-Lab).



Şekil 12: Video Akışında Tüm Görüntü Üzerinde Eş Zamanlı Nesne Tespiti Yapılması

Video akışında tüm görüntü üzerinde eş zamanlı nesne tespiti yapılması ; sağda orijinal görüntü, solda ise eş zamanlı sınıflandırılmış ve katmanlı olarak etiketlenmiş görüntü.

### 2.4.1. Yüz Tanıma Sistemi

Derin öğrenme yüz tanıma yarışması kapsamında 6.000 çift yüz resmi üzerinde tanıma işlemi en düşük hata seviyesini yakalamaya yönelik çeşitli firmaların yürütmüş olduğu çalışmalar neticesinde makinelerin yüz tanıma hata eşiği insan hata eşiğinin altına inmiştir.



Şekil 13: Yüz Tanıma Veri Setinde Derin Öğrenme Yöntemiyle Firmaların Yakaladığı Oranlar

### 2.4.2 Konuşma Tanıma Sistemi

Derin öğrenme konusunda öncü düşünür olarak ün yapan ve Çin’in en büyük arama motorunun baş uzmanı [Andrew Ng](http://cs.stanford.edu/people/ang/) ([g+](https://plus.google.com/113710395888978478005/posts)), son çalışmasında Baidu Derin Konuşma motorunun gürültülü ortamlarda bile derin öğrenme kullanarak sesli komutları anlayıp işlediğine vurgu yapmıştır. Bu çalışmada GPU işlemcileri kullanılarak 100.000 saatten daha fazla konuşma örnekleri sinir ağları ile eğitilerek bu alanda en düşük hata oranına ulaşılmıştır.



Şekil 14: Derin Öğrenme İle Konuşma Tanımanın Yapılması



Şekil 15: Derin Öğrenme İle Firmaların Konuşma Tanımadaki Hata Oranları

Yukarıdaki çalışmada GPU işlemcileri kullanılarak 100.000 saatten daha fazla konuşma örnekleri sinir ağları ile eğitilerek bu alanda en düşük hata oranına ulaşılmıştır.

Çoğu kişi %95 doğruluk ile %99 doğruluk arasındaki farkı anlamamaktadır. %99 doğruluk oranı oyun değiştiren bir orandır. Bu doğruluk oranına ulaşıldığında akıllı cihazlar tamamen sesle kullanılabilecek hale gelecektir.

Konuşma tanımanın gelişmesi nesnelerin internetinin (internet of things) yaygınlaştırmasını destekleyecektir. Bu sayede günlük yaşamda kullanılan tüm cihazlar ve araçlar insan ergonomisine uygun yapıda çalışarak yaşamı kolaylaştıracak şekilde birbirleriyle sürekli etkileşim halinde bulunacaktır.

## 2.5. SAVUNMA VE GÜVENLİK SEKTÖRÜNDE KULLANIMI

Teknolojinin etkinliğinin artarken boyutsal olarak küçülmesi ve enerji ihtiyacının buna bağlı olarak azalmasıyla kameralar günlük hayata üssel oranda katkı sağlamaktadır. Dünya genelinde üretilen mobil cihaz sayısı yılda 2 milyar adetken kamera sayısı bu sayının çok daha üzerindedir.



Şekil 16: Yeni Nesil Avuç İçine Sığabilen Üç Boyutlu Yüksek Çözünürlüklü Kameralar

Savunma alanında gerek silah üstü optiklere yönelik geliştirilen sistemlere işlemci desteği sağlanması gerekse tüm hareket eden platformlara konulan kameralar anlık incelenmesi gereken verinin miktarını büyük oranda artırmıştır. Kameraların savunma ve güvenlik alanlarında kullanımını daha da artacağı değerlendirildiğinde, sadece resim veya video akışındaki nesnelerin ne olduğu değil ayrıca nesnelerin birbirleriyle olan ilişkisini metne döken sistemlerin büyük bir insan kaynağı tasarrufu sağlayarak, her bir kamera sistemin anlık akıllı değerlendirme yapısına kavuşmasının yolunu açmaktadır.

Sonuç olarak; 2007 yılında başlayan mobil devrimin sonucu olarak son iki yılda büyük çıkış yapan derin öğrenme, nesnelerin interneti alanındaki gelişmeye paralel olarak yarı ve tam otonom sistemler ile robotların günlük yaşama katkısı giderek artacaktır. Gelişen teknoloji ile belirli bir uzmanlık alanında tecrübe artırılmış gerçeklik uygulamaları ile zahmetsiz bir şekilde sistemlere transfer edilebilecektir.

## 2.6. MAKİNE ÖĞRENMESİ

Makine öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir.

Makine öğreniminin başlıca uygulamaları makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, biyoinformatik, beyin-makine ara yüzleri ve kiminformatik, kredi kartı dolandırıcılığı denetimi, borsa çözümlemesi, DNA dizilerinin sınıflandırılması, konuşma ve el yazısı tanıma, bilgisayarlı görmede nesne tanıma, oyun oynama, yazılım mühendisliği, uyarlamalı web siteleri ve robot gezisidir.



Şekil 17: Makine Öğrenmesine Ait Başlıca Kavramlar

### 2.6.1 Denetimli Öğrenme

Veriler etkileşimli sistemlerden alınarak belirli bir düzende organize edilmesidir.

### 2.6.2 Denetimsiz Öğrenme

 Sınıf bilgisi barındırmayan verilerin içerisindeki gruplar irdelenmesidir.

### 2.6.3  Yarı Denetimli Öğrenme

Bu kavram tam olarak yukarıdaki iki kavramın arasında yer alır ve etiketlenmemiş büyük miktarda bir veri ile etiketlenmiş küçük miktarda bir verinin beraber kullanılmasıdır.

### 2.6.4  Takviyeli Öğrenme

Öğreticinin, sistemin ürettiği sonuç için doğru ya da yanlış olarak bir değerlendirmesidir.

### 2.6.5  Yoğun Öğrenme

Hiyerarşik öğrenme olarak da bilinir. Bu öğrenme yöntemi derin grafiklerde birçok doğrusal ve doğrusal olmayan dönüşümlerden ve çoklu işlem katmanlarından oluşturulmuş verilerde, üst düzey soyutlamalar kullanılarak elde edilen model girişimlerine dayalı bir dizi [algoritmalarla](http://www.endustri40.com/programlanabilir-otomasyon-kontrol-cihazipac/) geliştirilmiş makine öğrenmesidir.

## 2.7.YAPAY SİNİR AĞLARI

Yapay sinir ağları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir. Yapay sinir ağları; insan beyninden esinlenerek, öğrenme sürecinin matematiksel olarak modellenmesi uğraşı sonucu ortaya çıkmıştır. Bu nedenledir ki, bu konu üzerindeki çalışmalar ilk olarak beyni oluşturan biyolojik üniteler olan nöronların modellenmesi ve bilgisayar sistemlerinde uygulanması ile başlamış, daha sonraları bilgisayar sistemlerinin gelişimine de paralel olarak bir çok alanda kullanılır hale gelmiştir.

İnsan beyninin çalışma prensibini taklit ederek çalışan bu sistemler, her ne kadar bilgisayar teknolojisi hızlı bir gelişim göstermiş, işlem hızları nano saniyeler mertebesine inmiş olsa da, bırakalım insan beynini, ilkel bir canlı beyninin fonksiyonları dahi baz alındığında, böyle bir organizmanın yanında çok ilkel kalmaktadır. Nano saniyeler bazındaki işlem hızları ile YSA'lar, mili saniyeler mertebesindeki işlem hızları ile işlem yapan insan beyninin işlevselliğinin henüz çok uzağındadır. Burada kısa bir hatırlatma yapmak gerekirse; insan beyninde yaklaşık 10¹¹ sinir hücresinin varlığından bahsedilmekle birlikte, bu sayının bilgisayar ortamında modellenmesi şu an için mümkün görünmemektedir. Fakat karar hızı açısından insan beyni ile henüz yarışamasalar bile, YSA'lar yapısallıkları ve hassas eşleştirmelerin başarı ile gerçekleştirebilmeleri ile gün geçtikçe daha fazla uygulama alanı bulmaktadır.

## 2.8.DERİN ÖĞRENMEDE KULLANILAN KÜTÜPHANELER

Yapay sinir ağı modelinin kurulmasının ve eğitilmesinin zor olduğundan daha önce bahsetmiştik. Modeli kurmuş olsak bile doğruluk oranlarının artırılması için ağın eğitilmesi sürecinde milyonlarca görüntü kullanmak gereklidir. Eğer biz bir yüz tanıma sistemi yapmak istiyorsak ağ modelini oluşturup, eğitmek zorunda mıyız? Bu bizim için verimsiz bir yöntem olacaktır. Ayrıca bir ağın eğitilmesi işlemi aylarca süreceği için işleri kontrol altında tutmak zorluğu iyice artıracaktır. Bu durumlarda ne yapacağız? Eğitilmiş veri kümelerini ve bu sistemlerini dışarıda kullanılması için birçok firma hizmet vermektedir. Yayınladıkları bu kütüphaneler ile zahmetsizce yapacağımız işlemleri sürdürebiliyoruz. Bu kütüphanelerden en meşhur olanları aşağıdaki gibi verilmiştir. Bu kütüphaneler ve özellikleri aşağıdaki gibidir.

### 2.8.1 Torch

Torch makine öğrenme algoritmaları için kapsamlı destek sunan bilimsel hesaplama yapısıdır. Kolay ve hızlı kodlama dili LuaJIT ve C/CUDA uygulaması temelini kullanan kolay ve verimli bir yapıya sahiptir.

Çekirdek özelliklerin bir özeti:

* Güçlü bir n-boyutlu dizi (n-dimensional array)
* İndeksleme, kesme, yer değiştirme vb. bir çok rutin
* Luajıt ile hayranlık oluşturan c arayüzü
* Lineer cebir rutinleri
* Sinir ağı ve enerji tabanlı modeller
* Sayısal en iyileme rutinleri
* Hızlı ve etkili gpu desteği
* IOS, ANDROİD ve FPGA arka ucuna port ile gömülebilme

**Neden Torch ?**

Torch işlemleri oldukça basitleştirirken bilimsel algoritmalarınızı hazırlamayı azami esneklik ve hızda yapmanızı amaçlamaktadır. Torch Lua topluluğunun hazırladıklarının yanısıra makine öğrenmesi, bilgisayarlı görü, sinyal işleme, paralel işlem, resim, video, ses ve ağ iletişimi gibi [geniş ekosistem içindeki topluluk tarafından hazırlanmış paketler](https://github.com/torch/torch7/wiki/Cheatsheet) ile gelmektedir.

Torch’un kalbinde kompleks sinir ağı topolojisini uygulamada had safhada esnek, kullanımı kolay olan popüler sinir ağı ve en iyileme kütüphaneleri vardır. Sinir ağının isteğe bağlı grafiğini oluşturabilir ve CPU’lar ve GPU’lar arasında etkili bir şekilde paralel işlem yapabilirsiniz.

**Torch Kullanımı**

Torch’u kendi başınıza indirmek ve denemek için [Torch Başlangıç Rehberine](http://www.derinogrenme.com/torch/torch-baslangic-rehberi/) bakabilirsiniz. Torch açık kaynak kodludur, dolayısıyla [GitHub repo](https://github.com/torch/torch7) üzerinde sunulan kod ile de başlayabilirsiniz.

Torch sürekli geliştirilmektedir: hali hazırda Facebook, Google, Twitter, NYU, IDIAP, Purdue ile çeşitli firma ve araştırma laboratuarları tarafından kullanılmaktadır.

### 2.8.2 Caffe

Caffe derin öğrenme yapısı hızlı ve modüler olacak şekilde tasarlanmıştır. Berkeley Vision and Learning Center – [BVLC](http://bvlc.eecs.berkeley.edu/) (Berkeley Görüntü ve Öğrenme Merkezi) ve kullanıcı topluluğu tarafından geliştirilmiştir. [Yangqing Jia](http://daggerfs.com/) tarafından UC Berkeley’de doktora döneminde hazırlanmıştır. Caffe [BSD 2-Clause license](https://github.com/BVLC/caffe/blob/master/LICENSE) altında kullanıma sunulmuştur.

**Neden Caffe?**

İfade Yapısı yeni ve farklı fikirlerin yada uygulamaların ortaya atılmasını cesaretlendirmektedir. Model ve optimizasyonlar kodlama yapılmaksızın ayar dosyası üzerinden yapılabilmektedir. GPU makine üzerinde eğitim işlemini yapmak için CPU ve GPU değişimi bir etiket ayarı ile gerçekleştirilebilmekte böylece küme bilgisayarlara veya mobil cihazlara yayılım sağlanabilmektedir.

Genişletilebilir kod yapısı aktif geliştirmeyi desteklemektedir. Caffe’nin ilk yılında, 1.000 geliştirici tarafından birçok önemli değişiklik katkısı sağlanmıştır. Hem kod hem de model olarak modern bir yapının tesis edilmesine katkı sağlayanlara teşekkür ederiz.

Hız Caffe’nin araştırma deneyimleri ve endüstri uygulamaları için mükemmel hale getirmiştir. Caffe bir tek NVIDIA K40 GPU\* işlemciyle 60 Milyonun üzerinde resmi bir günde işleyebilir.  Bu da anlam çıkarma için 1 ms/resim, öğrenme için 4 ms/resim demektir. Şuna inanıyoruz ki Caffe erişilebilir en hızlı convnet uygulamasıdır.

Topluluk: Caffe hali hazırda akademik araştırma projelerine, girişim prototiplerine ve hatta geniş ölçekli görüntü alanındaki endüstriyel uygulamalar ile konuşma ve multimedyaya güç katmaktadır.

**Dokümantasyon**

* [Caffe ile Görüntü için Deep Learning Kullanımı (DIY Deep Learning for Vision with Caffe)](https://docs.google.com/presentation/d/1UeKXVgRvvxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/edit#slide=id.p) Öğretici sunum.
* [Öğretici Dokümantasyon](http://caffe.berkeleyvision.org/tutorial) Pratik rehber ve yapı referansı.
* [arXiv / ACM MM ‘14 raporu](http://arxiv.org/abs/1408.5093) ACM Multimedya Açık Kaynak yarışması için 4 sayfalık rapor (arXiv:1408.5093v1).
* [Kurulum Kılavuzu](http://caffe.berkeleyvision.org/installation.html) Ubuntu, Red Hat, OS X’de test edildi.
* [Model Zoo](http://caffe.berkeleyvision.org/model_zoo.html) BVLC suggests a standard distribution format for Caffe models, and provides trained models.
* [Developing & Contributing](http://caffe.berkeleyvision.org/development.html) Caffe’yi geliştirmek ve katkı sağlamak içn rehber.
* [API Dokümantasyonu](http://caffe.berkeleyvision.org/doxygen/annotated.html) Kod yorumlarından otomatik oluşturulmuş geliştirici dokümanı.

**Örnekler**

* [ImageNet dersi](http://caffe.berkeleyvision.org/gathered/examples/imagenet.html) “CaffeNet”i ImageNet verisi üzerinde eğitme ve test etme.
* [LeNet MNIST dersi](http://caffe.berkeleyvision.org/gathered/examples/mnist.html) MNIST el yazısı dijital verisi üzerinde “LeNet”i eğitme ve test etme.
* [CIFAR-10 dersi](http://caffe.berkeleyvision.org/gathered/examples/cifar10.html) CIFAR-10 verisi üzerinde Caffe’yi eğitme ve test etme.
* [Sitil tanıma için ince ayar yapma](http://caffe.berkeleyvision.org/gathered/examples/finetune_flickr_style.html) “Flickr Sitili” veri seti üzerinde ImageNet-eğitimli CaffeNet’e ince ayar yapma .
* [Feature extraction with Caffe C++ code.](http://caffe.berkeleyvision.org/gathered/examples/feature_extraction.html) Caffe olanaklarını kullanarak CaffeNet / AlexNet özelliklerini ayıklama.
* [Web demo](http://caffe.berkeleyvision.org/gathered/examples/web_demo.html) Flask web sunucusu üzerinde çalışan resim sınıflandırma demosu.
* [Siamese Network Dersi](http://caffe.berkeleyvision.org/gathered/examples/siamese.html) MNIST verisi üzerinde bir siamese networkü eğitme ve test etme

**Notebook Örnekleri**

* [ImageNet sınıflandırma](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/classification.ipynb) Önceden eğitilmiş ImageNet modelini kullanarak Python arayüzü ile resim sınıflandırma.
* [Filtre görselleştirme](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/filter_visualization.ipynb) Örnek bir resimle özellik çıkarma ve eğitilmiş filtreleri görselleştirme, katman katman gözden geçirilmiş.
* [R-CNN tespit etme (detection)](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/detection.ipynb) Python’da detektör olarak önceden eğitilmiş bir model çalıştırma.
* [Sınıflandırma için hazır SGD](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/hdf5_classification.ipynb) Resimsiz HDF5 verisi üzerinde lojistik regresyon eğitimi yapmak için  genel SGM en iyileyici olarak Caffe kullanma.
* [Model parametrelerini düzenleme](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/net_surgery.ipynb) How to do net surgery and manually change model parameters, making a fully-convolutional classifier for dense feature extraction.
* [Gömülü Siamese ağ](http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/siamese/mnist_siamese.ipynb) Özellik çıkarma ve gömülü Siamese ağ çizme.

### 2.8.3 Theano

Theano çok boyutlu diziler dahil matematik ifadelerini etkili bir şekilde tanımlamayı, en iyilemeyi ve değerlendirmeyi sağlayan bir Python kütüphanesidir. Theano özellikleri:

* **NumPy ile sıkı entegrasyon** – Theano-derlenmiş fonksiyonları içinde numpy.ndarray  kullanma.
* **GPU kullanımında açıklık** – Veri-yoğun hesaplamanın CPU’ya göre 140x kat daha hızlı yürütülmesi (sadece float32).
* **Etkili sembolik türev alma** – Theano bir veya daha fazla girdili fonksiyon için türev işlemi yapar.
* **Hız ve kararlılık optimizasyonları** –x değerinin çok küçük olduğu durumlarda bile log(1+x) için doğru cevabı alma.
* **Dinamik C kodu üretimi** – İfadeleri daha hızlı değerlendirme.
* **Kapsamlı birim-testi ve kendini-doğrulama** – Birçok hata türünü ortaya çıkarma ve tanımlama.

Theano 2007’den bu yana hesaplama içeren yoğun bilimsel araştırmalara güç katmaktadır. Ayrıca ders ortamında kullanım için yeterli anlaşılırlığa sahiptir (Montreal Üniversitesi, IFT6266 dersi).

### 2.8.4 Image-Net

### 2.8.5 Alexnet (Matlab)

## 2.9 YÖNTEM-UYGULAMA

Bitirme projesi boyunca yapay sinir ağ modellerinin zorluğundan bahsettik. Zor olması sebebi ile modeli ve modelin eğitimini kendimizin yapmayacağını belirttik bunun yerine büyük kuruluşların yüksek kapasitede ki bilgisayarları ile eğittiği modeli kullanacağız. Projemiz kapsamında iki adet proje geliştireceğiz bunlardan ilki matlab geliştirme ortamında bilgisayar ile ve kamerası kullanılarak yapılacak. Diğeri ise android telefon üzerinden telefonun kamerası kullanılarak yapılacak. Matlab geliştirme ortamında ise alexnet derin öğrenme kütüphanesini kullanacağız. Bu kütüphane ile ilgili detayları yukarıda vermiştik. Şimdi ise uygulama kısmını yapacağız.

## 2.10 MATLAB GELİŞTİRME ORTAMINDA UYGULAMANIN GELİŞTİRİLMESİ

Uygulama kısmında ilk olarak kuracağımız ortam Matlab geliştirme ortamı olacak. Geliştirme ortamı kurulduktan sonra matlab.inc tarafından geliştirilen ve makine öğrenmesi uygulamalarının kolay ve pratik geliştirilmesi için hazırlanmış Neural Network Toolbox’ı kuracağız bu toolbax ile ilgili açıklamalar <https://www.mathworks.com/products/neural-network.html> adresinde mevcuttur. Bu toolbox sadece hazır oluşumların yanında modellerimizi oluşturup geliştireceğimiz araçları da bize sağlıyor. Verilen adreste de belirttiği üzere derin öğrenme ağınızı oluşturun eğitin ve simüle edin diyor



Şekil 18: Neural Network Toolbox

Matlab geliştirme ortamında Neural Network Toolbox kurduktan sonra bilgisayarın webcam görüntülerinden çerçevede hangi objelerin olduğunun tespiti yapılacağından webcam Select supportpackagetoinstall menüsünden kamera ile ilgili kütüphaneleri indireceğiz.





Şekil 19: *Webcam Suppot Package İnstaller*

Matlab geliştirme ortamı, neuralnetworkstoolbox ve webcamsupportlibraryi de kurduktan sonra geliştirmeye başlayabiliriz. Matlab derin öğrenme ile ile ilgili uygulamamızı matlab>src klasörü altında geliştireceğiz burada object\_detection.m dosyasını oluşturacağız. Dosyamızın içesindeki kodlar ise şu şekilde olacak.

**clear**

**camera = webcam; % Connect tothecamera**

**nnet = alexnet; % Loadtheneural net**

**whiletrue**

**picture = camera.snapshot; % Take a picture**

**picture = imresize(picture,[227,227]); % Resizethepicture**

**label = classify(nnet, picture); % Classifythepicture**

**image(picture); % Show thepicture**

**title(char(label)); % Show thelabel**

**drawnow;**

**end**

**camera = webcam**

komutu burda sistemde aktif olan kamera bilgisini alacak camera objesine aktaracak kamera objesi içerisinde aşağıdaki gibi bir içerik barındırıyor.



Şekil 20: *Kamera Objesi*

**nnet = alexnet; % Loadtheneural net**

komutunda Alexnet kütüphanesi nnet objesine aktarılıyor burada eğitilmiş ağ yapısı yükleniyor.

**picture = camera.snapshot; % Take a picture**

**picture = imresize(picture,[227,227]); % Resizethepicture**

komutu sistemde yüklü kamera ile camera.snapshot komutu ile görüntü alınıyor ve 227x227 boyutunda yeniden yapılandırılıyor.

**label = classify(nnet, picture);**

classify fonksiyonu içerisine aldığı modeli ve resimi analiz ederek hangi sınıfa ait olduğunu string olarak döndürüyor label stringi bu veriyi tutuyor. Resim tekrar gösterilerek sınıfı ile birlikte ekranda yansıtılıyor.

Sonuç olarak aşağıdaki gibi görüntü ve üzerinde görüntü içeresindeki objenin ne olduğunu gösteriyor



Şekil 21: *Sonuç*

## 2.11. ANDROID İLE UYGULAMANIN GELİŞTİRİLMESİ

Matlab geliştirme ortamında yaptığımız uygumanın benzerini clarifai apisini kullanarak geliştireceğiz android uygulamasının bir çok sınıfı clarifai neural network yapısı ile sağlandığından uygulama tarafında sadece geliştirilmiş fonksiyon ve sınıfları kullanmak işlemimizi gerçekleştirmek için yeterli olacaktır. Geliştirdiğimiz uygulamada ise kameradan görüntü alarak clarifai apisine sorgu atacağız ve karşılığında objelerin etiketlerini alacız.

new AsyncTask<Void, Void, ClarifaiResponse<List<ClarifaiOutput<Concept>>>>() {  
@Override protected ClarifaiResponse<List<ClarifaiOutput<Concept>>>doInBackground(Void... params) {  
// The default Clarifai model that identifies concepts in images  
final ConceptModel generalModel = App.*get*().clarifaiClient().getDefaultModels().generalModel();  
  
// Use this model to predict, with the image that the user just selected as the input  
return generalModel.predict()  
 .withInputs(ClarifaiInput.*forImage*(ClarifaiImage.*of*(imageBytes)))  
 .executeSync();  
}  
  
@Override protected void onPostExecute(ClarifaiResponse<List<ClarifaiOutput<Concept>>> response) {  
 setBusy(false);  
 if (!response.isSuccessful()) {  
 showErrorSnackbar(R.string.*error\_while\_contacting\_api*);  
 return;  
}  
final List<ClarifaiOutput<Concept>> predictions = response.get();  
 if (predictions.isEmpty()) {  
 showErrorSnackbar(R.string.*no\_results\_from\_api*);  
 return;  
}  
adapter.setData(predictions.get(0).data());  
imageView.setVisibility(*VISIBLE*);  
preview.setVisibility(*GONE*);  
imageView.setImageBitmap(BitmapFactory.*decodeByteArray*(imageBytes, 0, imageBytes.length));  
}  
  
private void showErrorSnackbar(@StringRes int errorString) {  
 Snackbar.*make*(  
root,  
errorString,  
Snackbar.*LENGTH\_INDEFINITE*).show();  
}  
 }.execute();  
}

yukarıda main threaden ayrı yapılan sorgulama işleminde işlemlerin özeti yer almaktadır.

final ConceptModel generalModel = App.*get*().clarifaiClient().getDefaultModels().generalModel();

komutu ile yapının bir örneğini oluşturuyoruz.

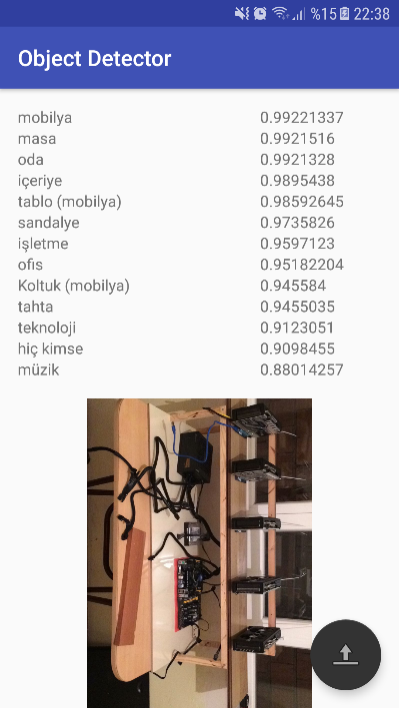
return generalModel.predict()  
 .withInputs(ClarifaiInput.*forImage*(ClarifaiImage.*of*(imageBytes)))  
 .executeSync();

komutu ile ise elimizde örneği imageBytes olan resim örneğinin modelin tahmin etme fonksiyonuna göndererek elimize bir

ClarifaiResponse<List<ClarifaiOutput<Concept>>>>

Objesi oluşturuyoruz bura bizim görüntüler üzerindeki etiketlerimiz yer alıyor.

Listelenen veriler ise aşağıdaki gibidir.



Şekil 22: Sonuç

## 2.12.TEST-SONUÇ

### 2.12.1 Matlab Geliştirme Ortamında Uygulamanın Test Edilmesi

Uygulamamız alexnet derin öğrenme modeli ile geliştirildi. Alexnet modelinin objelerin tespiti konusunda doğruluğunun test edilmesi işlemi de seçilen objelerin çeşitliliği ve ortamın ışığına dayandırılarak yapılacaktır. Bunun için aydınlık karanlık ve ortalama bir ışığa sahip ortamda görüntüler taratılmış olup, elde edilen veriler aşağıdaki tabloya aktarılmıştır.

Uygulamadan alınan görüntüler ise aşağıdaki gibidir.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Mause | Klayve | Bardak | Kalem | Gazete | Kitap |
| Karanlık |  |  |  |  |  |  |
| Ort. Işık |  |  |  |  |  |  |
| Aydınlık |  |  |  |  |  |  |

Tablo 2: Uygulamadan Alınan Görüntüler

### 2.12.2 Andorid Uygulamasının Test Edilmesi

Uygulamamız ClarifAi modeli ile geliştirildi. Modelin objelerin tespiti konusunda doğruluğunun test edilmesi işlemi de seçilen objelerin çeşitliliği ve ortamın ışığına dayandırılarak yapılacaktır. Bunun için aydınlık karanlık ve ortalama bir ışığa sahip ortamda görüntüler taratılmış olup, elde edilen veriler aşağıdaki tabloya aktarılmıştır.

Uygulamadan alınan görüntüler ise aşağıdaki gibidir.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Mause | Klayve | Bardak | Kalem | Gazete | Kitap |
| Karanlık |  |  |  |  |  |  |
| Ort. Işık |  |  |  |  |  |  |
| Aydınlık |  |  |  |  |  |  |

Tablo 3: Uygulamadan Alınan Görüntüler

# kaynaklar

* [1]wikipedia<https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvWWFwYXlfemVrw6IjY2l0ZV9ub3RlLTE> erişim tarihi (12/11/2017)
* [2] Ozan TAŞOVA, YAPAY SİNİR AĞLARI İLE YÜZ TANIMA, Haziran, 2011
* [3] <http://www.endustri40.com/makine-ogrenimi-nedir/> erişim tarihi (13/11/2017)
* [4]<http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-5.Hafta-YapaySinirAglari.pdf> erişim tarihi (13/11/2017)
* [5] <https://www.mathworks.com/products/neural-network.html> Matlab, erişim tarihi

(14/11/2017)

* [6]<http://ce.istanbul.edu.tr/Dosyalar/Dersler/2017/1/CSBM4068/Ders_CSBM4068_NN_Ch4_20171024_711.pdf> erişim tarihi (15/11/2017)
* [7]http://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/

# 

# EKLER

* Proje kod dosyası( sayfa)
* Proje paketi(1 cd)

# ÖZGEÇMİŞ

**Mehmet ONAR**

1993 Konya doğumlu. Kayaağzı İlköğretim Okulunda ilk ve ortaokulunu, liseyideMeram Anadolu Ticaret Lisesi’nde tamamladı. 2011 yılında Konya Selcuk Üniversitesi Teknik Bilimler Meslek Yüksek Okulunda başladığı üniversite akademik hayatını2013 yılından beri İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne devam ettirmektedir.

**Kübra ATMACA**

1985 İstanbul doğumlu. İlkokulu Mecidiyeköy İlköğretim İlkokulu’nda tamamladı. Liseyi 2001-2005 yılları arasında İstanbul Şişli Yunus Emre Süper Lisesi’nde tamamladı. 2013 yılından beri İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne devam etmektedir.