# Özet

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 verilerdevasa bir boyuta ulaşmıştır. Önceleri verilerin analizi klasik yöntemler ile yapılıyordu. Baştanaşağı inceleme ile eldeki az verinin incelenmesi sorun değildi. Savaşta girilen parametreler iletop namlusunun açısını hesaplama ile başlayan analiz için karmaşık algoritmalara gerekduyulmaz iken son yıllarda yaşanan veri deposu taşmaları ile bir çok yöntemi de beraberindegetirmektedir. Modern anlamada veri analizi için birçok alanda çalışmalar bulunmaktadır.Makine öğrenmesi ve yapay sinir ağları bu verilerin incelenmesi ve çıkarıma varılması içinçalışmalar yapılan alanlardandır. Ancak son yıllarda artan veri yükü modern anlamdaincelemenin de üstüne çıkmaktadır. Artık makinelerin işlem güçleri eldeki verilerin analiziniyapmakta zorlanmakta geç çözümler üretmektedir. Google, Microsoft ve Imagenet gibi firmalarverilerin analizi için yöntem geliştirerek derin öğrenme metotlarını geliştirmiş bu alanla ilgiliçalışmalar başlatmıştır. Derin öğrenme makine öğrenmesinin bir türü olup, çok katmanlı biryapay sinir ağıdır.[1] Yani makine öğrenmesi ile yapay sinir ağlarının ortak alanı denilebilir.Derin öğrenme ile işlem yapma yetenekleri ve yapay ağ modellerinin gelişmişliği kullanılarakbirçok alanda veri analizi mevcuttur. Yapay sinir ağları da makine öğrenmesi gibi eğitilerekdoğru sonuçlar elde edilebilir. Ancak ev kullanıcıları için böyle bir eğitim işlemci gücü açısındanmümkün olmayacağından eğitilmiş veri setlerini alarak test veri setlerini kullanmaları mümkünolacaktır. Derin öğrenme ile ilgili teknoloji firmalarının belirli kütüphaneleri mevcut olup bukütüphaneler açık kodlu olarak yayınlanmaktadır. Derin kütüphaneleri NVIDIA DIGITS, Theano,Caffe ve Torch’dir. Deeplearning birden fazla alanda kullanılır iken bu dönem ki bitirmetezimizde nesne tespiti konusunda çalışmamızda kullanacağımız bir yöntem olacak.

# Giriş

"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(MultipleInstance 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.

## Derin öğrenme

Derin öğrenme yapay zekâ(AI) alt dallarından makine öğrenmesi ne 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 kullanılsada matris normuna dönüştürebilinenherşeyianaliz 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(hiddenlayer) sahiptir. Fazlaca gizli katmanın oluşu modelin eğitilmesinide 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 belirtiğimiz sayı gittikçe artmaktadır. Bunun sebebi ise ne kadar çok görüntü incelenirse doğruluk oranı o kadar yukarıya çıkacaktır.

### Modelin oluşturulması

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 yada çıktı olarak bağlanması sonucu yapay sinir ağı oluşur.

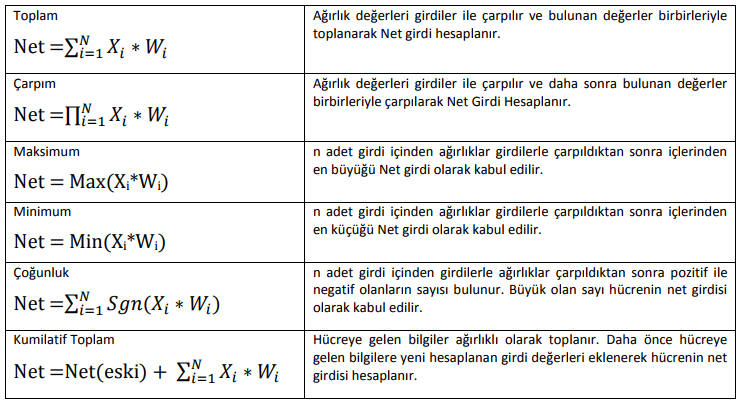


Şekil 1 yapay sinir ağı modeli

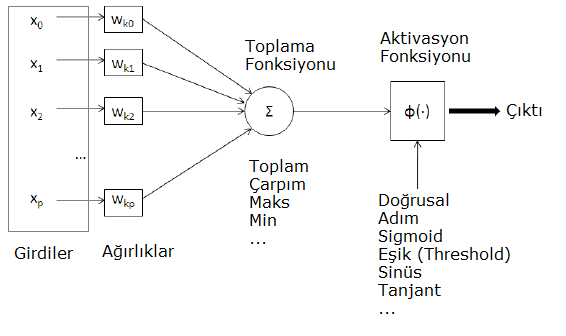
Şekil 1 ile gösterilmiş her bir daire bir nöronu temsil etmektedir. Giriş seviyesindeki nöronlar verileri alır snaptic ağırlıkları hesaplandıktan sonra çıktıları gönderilir. Giriş seviyesindeki nöronların çıktıları gizli katmandaki nöronların girişleri olarak belirlenmiştir. Yine burada hesaplanan değerler 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.

* **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
* **SnapticWeights(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.
* **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.

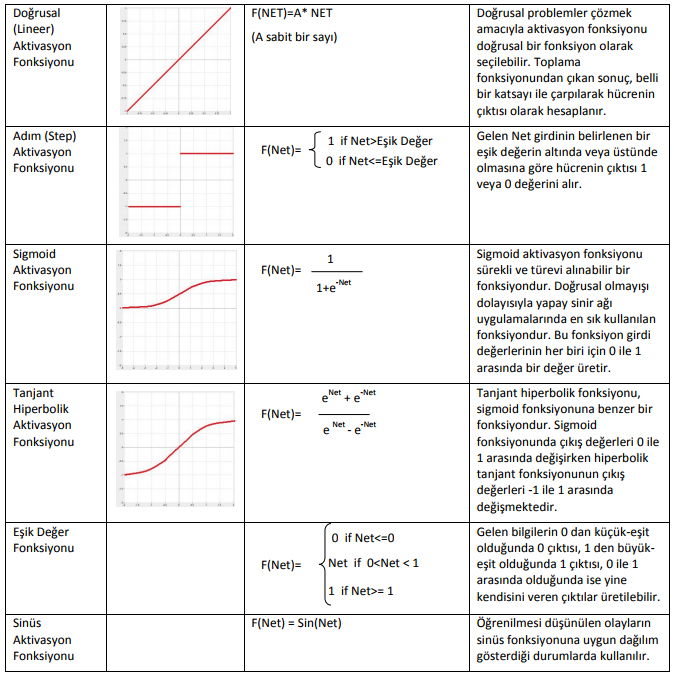
Tablo 1. Bazı Toplama Fonksiyonları



* **Activationfucntion(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 2 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.



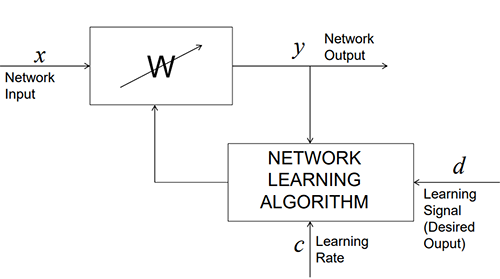
Şekil 3Derin öğrenme yüz tespiti yapısı[2]

### Modelin eğitilmesi

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. Bu algoritmaların isinleri aşağıdaki gibidir.

* 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.



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

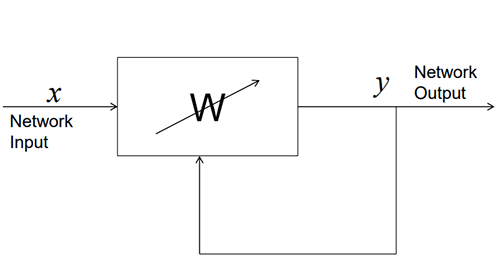
Adapte edilmiş ağırlık vektörü

(t+1)=(t)+ Δ(t)

Genel Discrete-Time Form

* 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.



Öğrenme sinyali mevcut değil.

**I’NİNCİ NÖRON İÇİN 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

### Derin öğrenme ile ilgili yapılan çalışmalar

Bir ağaç dalına tünemiş kuş (Derin öğrenme kullanılarak üretilmiştir.)

[Stanford Üniversitesi’nden öncü araştırmacı AndrejKarpathy](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.”“pembe kıyafetli kız havada zıplıyor.”“mavi dalış kıyafetli adam dalga üstünde sörf yapıyor.”iki genç kız lego oyuncağı ile 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

Google son dönemde bünyesine kattığı, [DeepMind](http://deepmind.com/" \t "_blank) firması ile yürttüğü 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" \t "_blank)) ([Google neden 400 Milyon Dolara DeepMindfimasını satın aldı?](http://www.technologyreview.com/news/524026/is-google-cornering-the-market-on-deep-learning/" \t "_blank))

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).

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

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.

Yüz tanıma veri setinde derin öğrenme yöntemiyle firmaların yakaladığı oranlar

Derin Öğrenmeyle Konuşma Tanıma

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.

Derin öğrenme ile konuşma tanımanın yapılmasıDerin öğrenme ile 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.

Derin Öğrenmenin Savunma ve Güvenlik Sektöründe Kullanımı

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.

Yeni nesil avuç içine 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 kullanımının adaha 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.

## Makine öğrenmesi

[Makine](http://www.endustri40.com/karanlik-fabrikalar-ile-insansiz-uretim/) öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir.

Makine öğreniminin başlıca uygulamaları [makine algılaması](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPU1ha2luZV9hbGclQzQlQjFsYW1hcyVDNCVCMSZhY3Rpb249ZWRpdCZyZWRsaW5rPTE), [bilgisayarlı görme](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUJpbGdpc2F5YXJsJUM0JUIxX2clQzMlQjZybWUmYWN0aW9uPWVkaXQmcmVkbGluaz0x), [doğal dil işleme](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvRG8lQzQlOUZhbF9kaWxfaSVDNSU5RmxlbWU), [sözdizimsel örüntü tanıma](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPVMlQzMlQjZ6ZGl6aW1zZWxfJUMzJUI2ciVDMyVCQ250JUMzJUJDX3RhbiVDNCVCMW1hJmFjdGlvbj1lZGl0JnJlZGxpbms9MQ), [arama motorları](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQXJhbWFfbW90b3JsYXIlQzQlQjE), [tıbbi tanı](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPVRhbiVDNCVCMV8oeWFwYXlfemVrYSkmYWN0aW9uPWVkaXQmcmVkbGluaz0x), [biyoinformatik](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQml5b2luZm9ybWF0aWs" \o "Biyoinformatik), [beyin-makine arayüzleri](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUJleWluLW1ha2luZV9hcmF5JUMzJUJDemxlcmkmYWN0aW9uPWVkaXQmcmVkbGluaz0x) ve [kiminformatik](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUtpbWluZm9ybWF0aWsmYWN0aW9uPWVkaXQmcmVkbGluaz0x" \o "Kiminformatik (sayfa mevcut değil)), [kredi kartı dolandırıcılığı](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUtyZWRpX2thcnQlQzQlQjFfZG9sYW5kJUM0JUIxciVDNCVCMWMlQzQlQjFsJUM0JUIxJUM0JTlGJUM0JUIxJmFjdGlvbj1lZGl0JnJlZGxpbms9MQ) denetimi, [borsa](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvQm9yc2E) çözümlemesi, [DNA dizilerinin](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvRE5BX2RpemlzaQ) sınıflandırılması, [konuşma](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUtvbnUlQzUlOUZtYV90YW4lQzQlQjFtYSZhY3Rpb249ZWRpdCZyZWRsaW5rPTE) ve [elyazısıtanıma](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPUVseWF6JUM0JUIxcyVDNCVCMV90YW4lQzQlQjFtYSZhY3Rpb249ZWRpdCZyZWRsaW5rPTE" \o "Elyazısı tanıma (sayfa mevcut değil)), bilgisayarlı görmede [nesne tanıma](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPU5lc25lX3RhbiVDNCVCMW1hJmFjdGlvbj1lZGl0JnJlZGxpbms9MQ), [oyun oynama](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvU3RyYXRlamlfb3l1bnU), [yazılım mühendisliği](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvWWF6JUM0JUIxbCVDNCVCMW1fbSVDMyVCQ2hlbmRpc2xpJUM0JTlGaQ), [uyarlamalı web siteleri](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPVV5YXJsYW1hbCVDNCVCMV93ZWJfc2l0ZXNpJmFjdGlvbj1lZGl0JnJlZGxpbms9MQ) ve [robot gezisidir](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3cvaW5kZXgucGhwP3RpdGxlPVJvYm90X2dlemlzaSZhY3Rpb249ZWRpdCZyZWRsaW5rPTE).



Makine öğrenmesine ait başlıca kavramlar;

**Denetimli Öğrenme:**Veriler etkileşimli sistemlerden alınarak belirli bir düzende organize edilmesidir

**Denetimsiz Öğrenme:** Sınıf bilgisi barındırmayan verilerin içerisindeki gruplar irdelenmesidir

**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.

**Takviyeli Öğrenme**: Öğreticinin, sistemin ürettiği sonuç için doğru ya da yanlış olarak bir değerlendirmesidir.

**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.

## Yapay sinir ağları

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 işlen 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.

# Derin Öğrenmede Kullanılan 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.

## 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)
* indeksleme, kesme, yer değiştirme vb. bir çok rutin
* LuaJIT ile hayranlık oluşturan C arayüzü
* lineer cebir rutinleri
* sinir ağı ve enerji tabanlı modeller
* sayısal en iyileme rutinleri
* Hızlı ve etkili GPU desteği
* iOS, Android 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" \t "_blank) 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.

## 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. Topluluğumuza katılmak için [caffe-kullanıcı grubu](https://groups.google.com/forum/" \l "!forum/caffe-users" \t "_blank) ve  [Github](https://github.com/BVLC/caffe/" \t "_blank) sayfalarını ziyaret edebilirsiniz.

\* ILSVRC2012-kazananı [SuperVision](http://www.image-net.org/challenges/LSVRC/2012/supervision.pdf" \t "_blank) modeli ve IO önbellekleme. [Performans detayları](http://caffe.berkeleyvision.org/performance_hardware.html) için tıklayınız.

## 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.

## 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).

## Image-Net//buralara kütüphaneleri açıklayacağız

## Alexnet(Matlab)//buralara kütüphaneleri açıklayacağız

# 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 isealexnet derin öğrenme kütüphanesini kullanacağız. Bu kütüphane ile ilgili detayları yukarıda vermiştik. Şimdi ise uygulama kısmını yapacağız.

## Matlab Geliştirme ortamında uygulamanın geliştirilmesi

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 araçlarıda bize sağlıyor. Verilen adrestede belirttiği üzere derin öğrenme ağınızı oluşturun eğitin ve simüle edin diyor



Şekil 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 webcamsuppotpackageinstaller

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 olaccak.

**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**

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



Şekil 6 kamera objesi

**nnet = alexnet; % Loadtheneural net**

komutundaalexnet 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);**

classfy fonksiyonu içerisine aldığı modeli ve resimi analiz ederek hangi sınıfa ait olduğunu string olarak döndürüyor labelstringi 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 7 sonuç

## Andorid ile uygulamanın geliştirilmesi

Matlab geliştirme ortamında yaptığımız uygumanın benzerini clarifai apisini kullanarak geliştireceğiz android uygulamasının bir çok sınıfı clarifai neural network yapısı ile sağlandığından uygulama tarafaında 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.

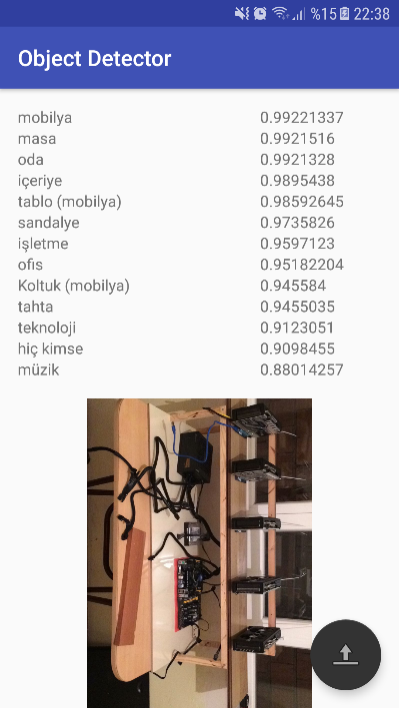
return generalModel.predict()  
 .withInputs(ClarifaiInput.*forImage*(ClarifaiImage.*of*(imageBytes)))  
 .executeSync();

komutu ile ise elimizde örneği imageBytes olan resim örneğinin modelin tahmin etme fonkiyanuna gönderek elimize bir

ClarifaiResponse<List<ClarifaiOutput<Concept>>>>

Objesi oluşturuyoruz bura bizim görüntüler üzerindeki etketlerimiz yer alıyor.

Listelenen veriler ise aşağıdaki gibidir.



Şekil sonuç

# Test – Sonuç

## 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.

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

Uygulamadan alınan görüntüler ise aşağıdaki gibidir.

## 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.

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

# Kaynakça

[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/