

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK NESNE TANIMA

Volkan KAYA

Erzincan Binali Yıldırım University, Faculty of Engineering, Computer Engineering, Erzincan, Turkey
vkaya@erzincan.edu.tr

Servet TUNCER

Fırat University, Technology Faculty, Electrical Electronics Engineering, Elazığ, Turkey
stuncer@firat.edu.tr

Ahmet BARAN

Erzincan Binali Yıldırım University, Faculty of Engineering, Computer Engineering, Erzincan, Turkey
baran@erzincan.edu.tr

Özet

Derin öğrenme, makine öğrenmesinin bir alt çalışma alanı olup son yıllarda oldukça popüler hale gelmiştir. Derin öğrenme gerçek hayatta karşılaşılan problemlerde oldukça geniş bir uygulama alanına sahiptir. Özellikle çözümü zor ve karmaşık olan çok farklı alanlardaki problemlerin çözümünde kullanılmakta ve genellikle çok başarılı sonuçlar elde edilmektedir. Bu şekilde bilgisayarlar için oldukça zor bir problem olan nesne tanıma problemlerinde, derin öğrenme algoritmaları oldukça başarılı sonuçlar üretmektedir. Nesne tanımlama alanında, bir nesnenin eldeki görüntü içerisinde görüntünün neresinde olduğunu tespit ettikten sonra, bu sınırlar içerisindeki nesnenin görsel özelliklerinin kullanılması ile gerçekleştirilen eğitilmiş bir sınıflandırma problemidir. Hâlihazırda nesne tanıma uygulamaları için en çok kullanılan derin öğrenme algoritması Evrişimsel Sinir Ağı (Convolutional Neural Network(CNN)), uzun yıllardır bilinen ve çok fazla uygulama alanına sahip çok katmanlı sinir ağlarının geliştirilmiş halidir. Evrişimsel Sinir Ağı, görüntünün segmentasyonu ve sınıflandırılması gibi klasik görüntü işleme problemlerinde şimdiye kadar çok iyi sonuçlar sergilemiştir. Bu çalışmada, nesne tanıma uygulamasına yönelik literatürdeki derin öğrenme yöntemleri kullanan çalışmalar incelenerek, bu yöntemlerin çeşitli açılardan kıyaslanması ve kullanılmalarına yönelik değerlendirmeler yapılmıştır.

Anahtar Kelimeler: Derin Öğrenme, Nesne Tanıma, Derin Öğrenme Modelleri, Evrişimsel Sinir Ağı

1. Giriş

Derin öğrenme, makine öğrenmesinin bir alt çalışma alanı olup son yıllarda birçok alanda kullanılmaktadır. Özellikle çözümü zor ve karmaşık olan nesne tanıma problemlerinin çözümünde kullanılmakta ve genellikle çok başarılı sonuçlar elde edilmektedir. Derin öğrenme ile nesne tanımlama ilk olarak ILSVRC (ImageNet Large Scale Visual Recognition Competition) yarışmasında büyük etki görmüştür. Nesne tanıma uygulamaları için en çok kullanılan derin öğrenme algoritması, Evrişimsel Sinir Ağı (Convolutional Neural Network(CNN)) kabul edilir (Angelova et al., 2015; Hinton et al., 2012). Evrişimsel Sinir Ağı, görüntü segmentasyonu ve nesne sınıflandırılması gibi klasik görüntü işleme problemlerinde çok iyi sonuçlar sergilemektedir. Bu sinir ağı bir girdi görüntüsünü alıp, nesneleri birbirinden ayırabilen derin öğrenme algoritmasıdır. Birçok nesne tanımda kullanılan evrişimsel sinir ağları, evrişim katmanı, havuzlama katmanı, aktivasyon fonksiyonu, dropout katmanı, tam bağlı katman ve sınıflandırma katmanından oluşmaktadır (Guo et al., 2016). Nesne tanımanın derin öğrenme yöntemlerini kullanarak başarılı olmasının en önemli nedeni öğrenme aşamasında yeteri kadar verinin öğrenilebilmesidir. Eğitim aşamasında büyük miktarda veri kullanılarak evrişimsel sinir ağının eğitilmesi gerekmektedir (Ponti et al., 2017). Veri miktarının artması çalışma süresine etki etmekte ve daha yüksek sistem gereksinimlerine ihtiyaç duymaktadır. Bazı derin öğrenme işlemleri günlerce sürebilmektedir. Bu sebeple evrişimsel sinir ağlarının eğitimi için Merkezi İşlem Birimi (Central Processing Unit(CPU)) yerine Grafik İşlem Birimi (Graphic Processing Unit(GPU)) kullanılmaktadır. GPU'ların geliştirilmesiyle farklı alanlarda derin öğrenme modellerinin kullanımına olanak sağlanarak daha hızlı eğitim ve test işlemi yapılabilmektedir.

Derin öğrenmede nesneyi tanıma, onu bir sınıf içerisinde değerlendirmek anlamına gelmektedir. Nesne tanıma bilgisayarlı görmedeki en zor görevlerden biridir ve her zaman dikkat çekmiştir (Lei Liu et al., 2017; Szegedy et al., 2013). Son yıllarda nesne tanıma için farklı algoritmalar ve yöntemler geliştirilerek, oldukça önemli ilerlemeler kaydedilmektedir (Daş et al., 2019). Ancak nesne tanımda ortam değişimi, arka plan çıkarımı gibi zorluklar hala güçlü bir problem olmaktadır (Li et al., 2018).

Bu çalışmada, derin öğrenme yöntemleri kullanılarak nesne tanıma algoritmalarına yönelik kapsamlı bir araştırma sunulmaktadır.

Çalışmanın 2. Bölümünde, evrişimsel sinir ağları konusunda detaylı bilgiler verilmekte ve özellikle derin öğrenme alanında kullanılan evrişimsel sinir ağları katmanları hakkında güncel bilgiler sunulmaktadır.

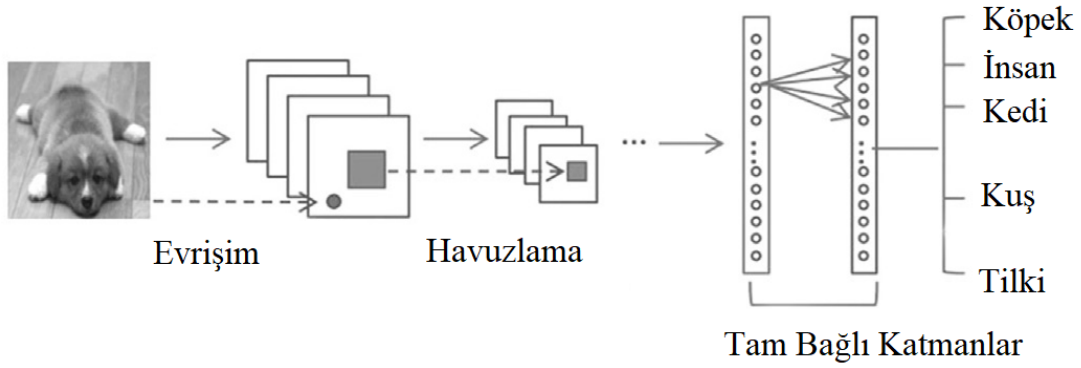
3. bölümde ise son yıllarda nesne tanıma için derin öğrenme modellerinden, LeNet, AlexNet, ZfNet, VggNet, GoogleNet, ResNet kapsamlı bir şekilde anlatılmaktadır. Çalışma değerlendirmenin yapıldığı sonuç bölümü ile sonlanmaktadır.

2. Materyal ve Yöntem

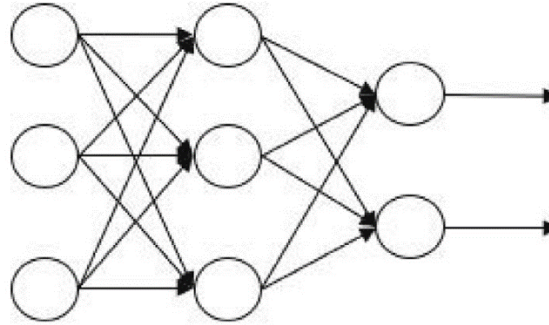
Derin öğrenme, makine öğrenmesinin bir alt çalışma alanı olup gelişmiş yapay sinir ağlarından oluşmaktadır. Derin öğrenme ilk olarak Hinton'un yapmış olduğu bir çalışmada yer almıştır. Bu çalışma derin evrişimsel sinir ağı olarak literatürde yer almaktadır (Hinton & Salakhutdinov, 2006). Son yıllarda derin öğrenmenin kullanıldığı çok sayıda çalışmalar yapılmıştır. Nesne tanıma, görüntü ve video işleme, doğal dil işleme, robotik, hastalık teşhisleri, biyomedikal, otonom araç sistemleri gibi alanlarda kullanılmaktadır (Ciaburro & Venkateswaran, 2017). Derin öğrenmenin farklı alanlarda kullanılmasının en önemli sebebi elde ettiği yüksek doğruluktur (İnik & Ülker, 2017). Derin öğrenme gizli katmana sahip çok katmanlı yapay sinir ağından oluşmaktadır. Bu katmanlar içerisinde görüntüye ait özellik çıkarımları gerçekleştirilmektedir. Derin öğrenme algoritmalarının başında evrişimsel sinir ağı gelmektedir. 2012 yılında yapılan ILSVRC yarışmasında evrişimsel sinir ağı üstün başarı göstererek birçok alanda kullanılmaya başlamıştır (Krizhevsky et al., 2012; Salur et al., 2019). Derin öğrenme, ileri düzeyli evrişimsel sinir ağı yapısını temsil etmektedir. Evrişimsel sinir ağı, giriş katmanı, evrişim katmanı, havuzlama katmanı, aktivasyon fonksiyonu, dropout katmanı, tam bağlantılı katman ve sınıflandırma katmanlarından oluşmaktadır (Lecun et al., 1998).

2.1. Evrişimsel Sinir Ağları

Evrişimsel sinir ağı nesne tanıma gibi uygulamalarda en çok kullanılan derin öğrenme algoritmalarından birisidir ve son yıllarda nesne tanıma gibi alanlarda yoğun bir şekilde kullanılmaktadır. Şekil 1'de gösterildiği gibi temel bir evrişimsel sinir ağı, evrişimli katman, havuzlama katmanı ve tam bağlı katmanlardan oluşmaktadır. Şekil 2'de ise evrişimli sinir ağı yapısı görülmektedir.



Şekil 1: Evrişimli sinir ağı yapısı (Zhang et al., 2019)



Şekil 2: Sinir ağı yapısı (Zhang et al., 2020)

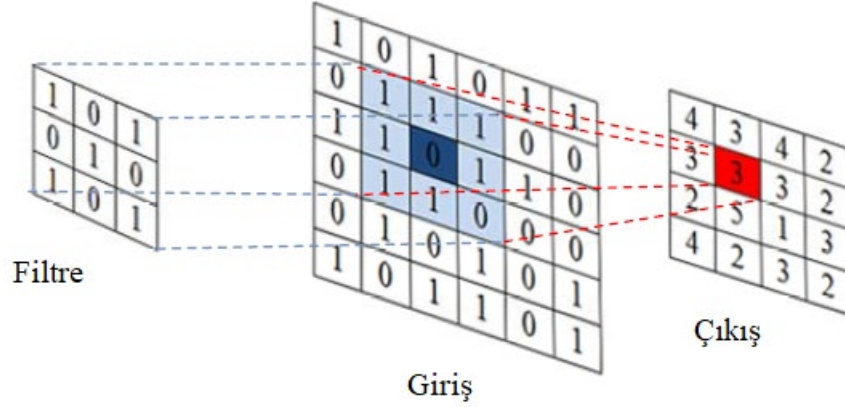
2.1.1. Giriş Katmanı

Evrişimli sinir ağının ilk katmanı olup veri giriş katmanı olarak da adlandırılmaktadır. Giriş olarak verilen verinin çözünürlüğü ve boyutu, tasarlanacak modelin mimarisine göre belirlenmesi gerekmektedir. Giriş verisinin uygunluğu modelin başarımını, eğitim süresini, test süresini ve bellek ihtiyacını etkileyeceğinden iyi seçilmesi gerekmektedir. Giriş veri sayısının yüksek seçilmesi başarı oranını artırabilmektedir. Bunun yanın da eğitim ve test süresinin de uzamasına neden olmaktadır. Bunun için daha yüksek işlemci ve bellek kapasitesine ihtiyaç duyulmaktadır. Giriş veri boyutunun küçük seçilmesi eğitim ve test süresinin azalmasına neden olmaktadır. (İnik & Ülker, 2017). Buna bağlı olarak model performansının düşmesine ve başarı oranının azalmasını neden olmaktadır. Bunu için giriş görüntüsünün boyutu seçilirken modelin ihtiyacına göre ideal bir sayı seçilmesi gerekmektedir (Doğan & Türkoğlu, 2019; Tamura & Tateishi, 1997).

2.1.2. Evriřim Katmanı

Evriřimsel sinir aęlarının temelini oluřturan bu katman dnüşüm katmanı olarak da adlandırılmaktadır (Doęan & Trkoęlu, 2019). Bu katmanda belirli bir filtre girdi verisi zerinde dolařtırılarak, belirgin zelliklere sahip yeni bir girdi oluřturulması gerekleřtirilmektedir (Pang et al., 2018). Bu ařamada nemli rol oynayan filtreler, girdi verisinden kk yeni bir matris elde edilmesini saęlamaktadır (Salur et al., 2019). Evriřim iřlemi sinir aęlarının nemli bir parasını oluřturmaktadır. Tercih edilecek filtre, aęın bařarı oranını ve eęitim srecini etkileyeceęinden ideal bir filtre seilmelidir (Cireřan et al., 2011; İnik & lker, 2017).

Filtreler 1x1, 2x2, 3x3, 5x5, 7x7 gibi boyutlarda kullanılarak evriřim iřlemi gerekleřtirilmekte (Lingqiao Liu et al., 2015) ve bu ařamada zellik haritası oluřturulmaktadır. Filtrelemenin eęitim ařamasının her adımında uygulanması ve yinelenmesi nemli blgelerin tespiti iin gereklidir. řekil 3’de evriřim iřlemi grlmektedir.



řekil 3: Evriřim iřlemi (Liu hui-bin et al., 2016)

Evriřimli sinir aęlarında yapılan hesaplamalar sonucunda, girdi ve ıkıř boyutu arasında farklılıklar meydana gelmektedir. Evriřim iřleminden sonra oluřacak yeni grntnn(matris) boyutu ařaęıdaki denklem ile hesaplanmaktadır (Altan, 2019).

$$n_{out} = \left(\frac{n_{in} + 2p - f}{s} \right) + 1 \quad (1)$$

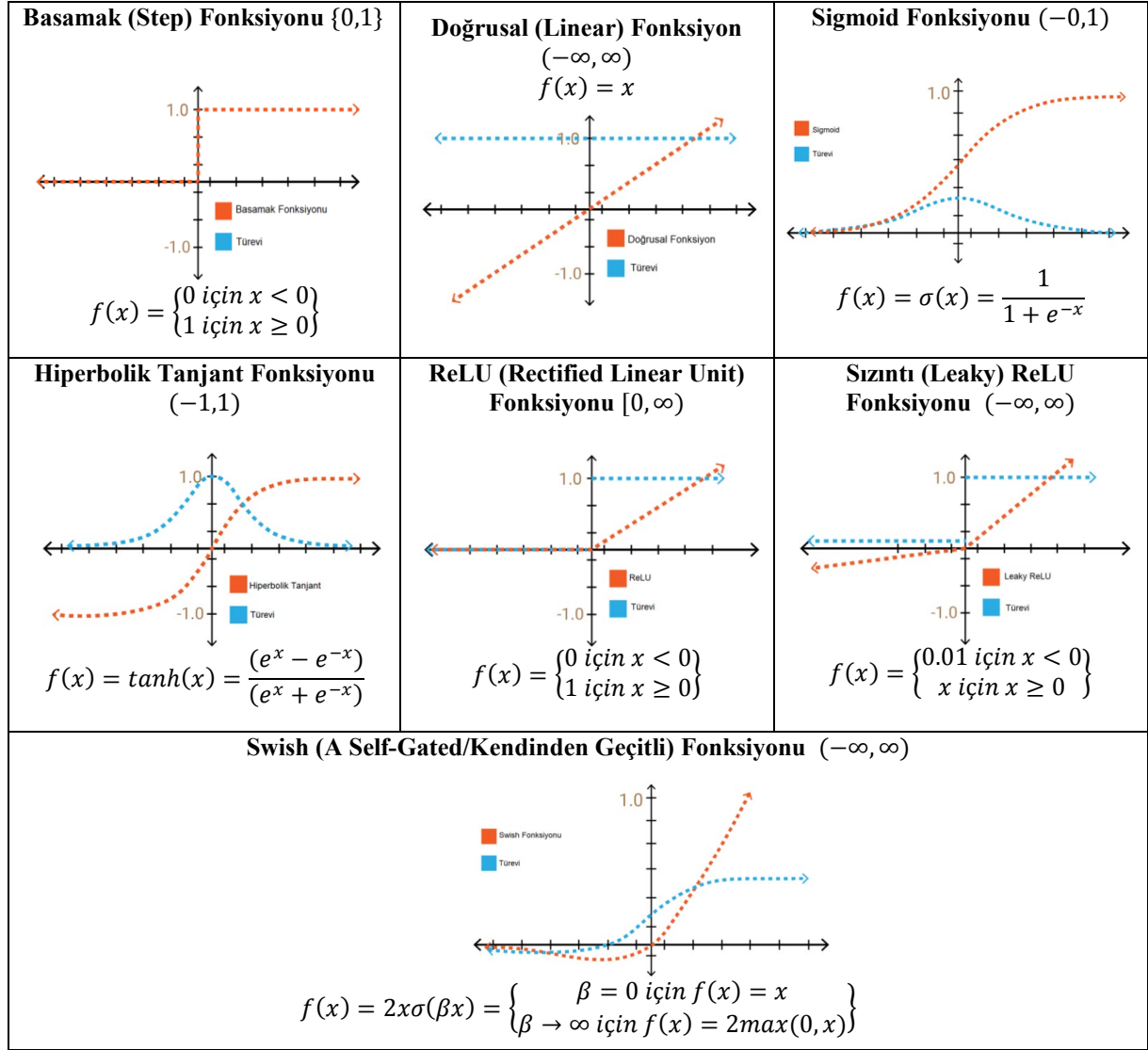
Bu denkleme gre ıkıř matrisinin boyutu n_{out} , girdi resminin boyutu n_{in} , piksel ekleme padding(p), filtre boyutu f, adım kaydırma stride(s) baęlı olarak deęiřmektedir.

2.1.3. Aktivasyon Fonksiyonu

Bu katman evriřim katmanından sonra gelmekte ve evriřimsel sinir aęı iin en yaygın kullanılan katmandır (İnik & lker, 2017). Aktivasyon fonksiyonu, sinir aęına girdi olarak gelen deęere karřılık yeni bir ıktı deęeri reten fonksiyondur. Her evriřim sinir aęı aktivasyon fonksiyonuna tabi tutularak yeni bir matris deęeri retmekte ve kendinden sonra gelen katmanın girdi deęerini oluřturmaktadır. Bu nedenle aktivasyon fonksiyonları sinir aęlarında nemli rol oynamaktadır. Evriřimsel sinir aęında aktivasyon fonksiyonu kullanılmadıęında ıktı deęeri basit bir doęrusal fonksiyon gibi davranmaktadır (Salur et al., 2019).

Şekil 4: Aktivasyon fonksiyonları ve matematiksel ifadeleri

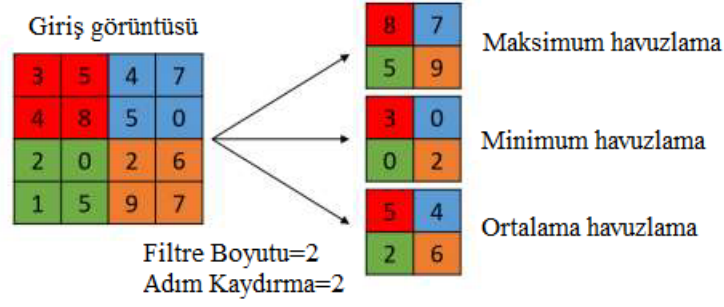
Aktivasyon fonksiyonu genellikle Basamak(Step), Doğrusal(Linear), Sigmoid, Softmax, Hiperbolik Tanjant, Doğrultulmuş Doğrusal Birim (Rectified Linear Unit(ReLU)), Sızıntı(Leaky) ReLU ve Kendinden Geçitli (A Self-Gated(Swish)) fonksiyonları kullanılmaktadır. Sıklıkla kullanılan aktivasyon fonksiyonlarının grafikleri ve



matematiksel denklemleri Şekil 4'de görülmektedir.

2.1.4. Havuzlama Katmanı

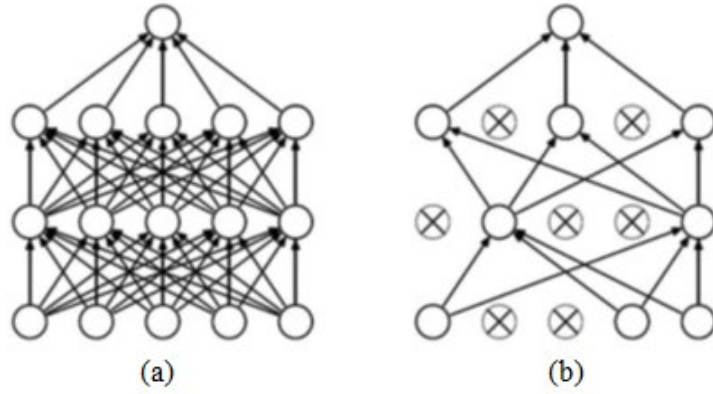
Evrişim işleminden sonra (Castelluccio et al., 2015) ağdaki parametre ve veri boyutunu azaltmak için kullanılan katmandır. Havuzlama işlemi sonucunda veri boyutunun azaltılması sinir ağının daha hızlı olmasını sağlayarak, sinir ağının hesaplama miktarını ve kullanılan bellek miktarını azaltmaktadır (Hinton, 2012). Veri boyutunun azaltılması belli bir filtreleme işlemi ile gerçekleştirilmektedir. Bu filtreleme, en büyük değerlerini (maximum pooling), en küçük değerlerini (minimum pooling) veya değerlerin ortalamasını (average pooling) alarak havuzlama işlemi yapmaktadır. Bu katmanda genellikle en büyük değerleri havuzlama yöntemi daha başarılı sonuçlar vermektedir. Şekil 5'de örnek bir havuzlama işlemi görülmektedir.



Şekil 5: Havuzlama işlemi (Türkoğlu et al., 2020)

2.1.5. Dropout Katmanı

Evrişimsel sinir ağı katmanları arasındaki aşırı öğrenmeyi ve ezberlemeyi ortadan kaldıran katmandır. Bu katmanda sinir ağında ezber yapan rasgele düğümlerin ortadan kaldırılması ile ağı ezber yapmasını engellemektedir (Srivastava et al., 2014). Bu nedenle bu katmanın kullanılması ağı performansını artırmaktadır (Xiao et al., 2016). Şekil 6 (a)'da evrişimsel sinir ağı yapısı, (b)'de ise dropout katmanının uygulanmış sinir ağı yapısı görülmektedir.

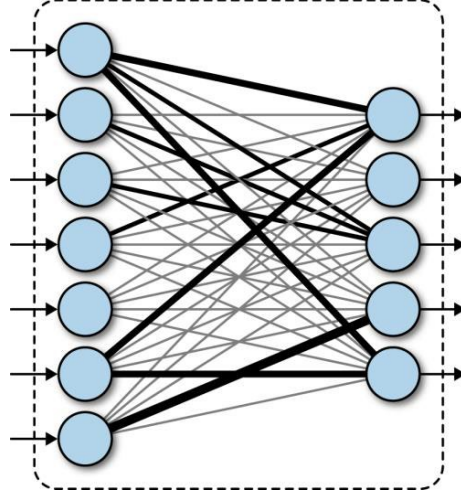


Şekil 6: (a) Evrişimsel sinir ağı yapısı

(b) Dropout uygulanmış sinir ağı yapısı

2.1.6. Tam Bağlı Katman

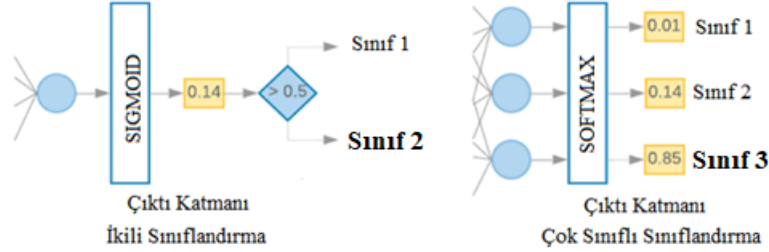
Yapılan uygulamalarda kullanılan veri setleri çok boyutlu yapılardan oluşabilmektedir. Bu katman kendinden önce gelen katmandaki nöronların tamamını tek satırlı bir dizi şekline getirmektedir (Adler et al., 2016; Doğan & Türkoğlu, 2018). Örneğin en son katmanda elde edilen veri boyutu $3 \times 5 \times 64$ değerlerinde olduğu düşünülürse, bu katman sayesinde verilerin yeni boyutu $3 \times 5 \times 64 = 960$ olacaktır. Şekil 7'de tam bağlı katman yapısı görülmektedir.



Şekil 7: Tam bağlı katman yapısı (Tammina, 2019)

2.1.7. Sınıflandırma Katmanı

Sınıflandırma katmanının görevi, tam bağlı katmandan elde edilen veriyi belirlenen sınıf sayısı kadar sınıflandırma yaparak çıkış değeri üretmektedir. Sınıflandırma katmanında genellikle sigmoid ve softmax aktivasyon fonksiyonları kullanılmaktadır (Cireşan et al., 2011). Örneğin 10 farklı nesnenin sınıflandırılması yapılacaksa, sınıflandırma katmanı değeri 10 olmalıdır. Tam bağlı katman sonucunda çıkış değeri 960 seçilirse, sınıflandırma katmanı için 960x10 ağırlık matrisi elde edilmektedir. Sınıflandırma sonucunda 10 farklı nesne 0-1 aralığında bir değer alarak, değeri en yüksek olan nesne sınıflandırma tahmin değerini oluşturmaktadır.



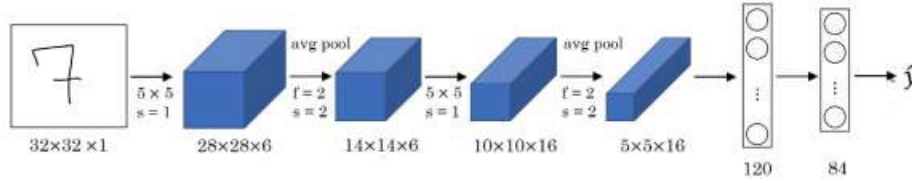
Şekil 8: Sınıflandırma işlemi

(<https://Developers.Google.Com/Machine-Learning/Guides/Text-Classification/Step-4>)

3. Derin Öğrenme Modelleri

3.1. LeNet

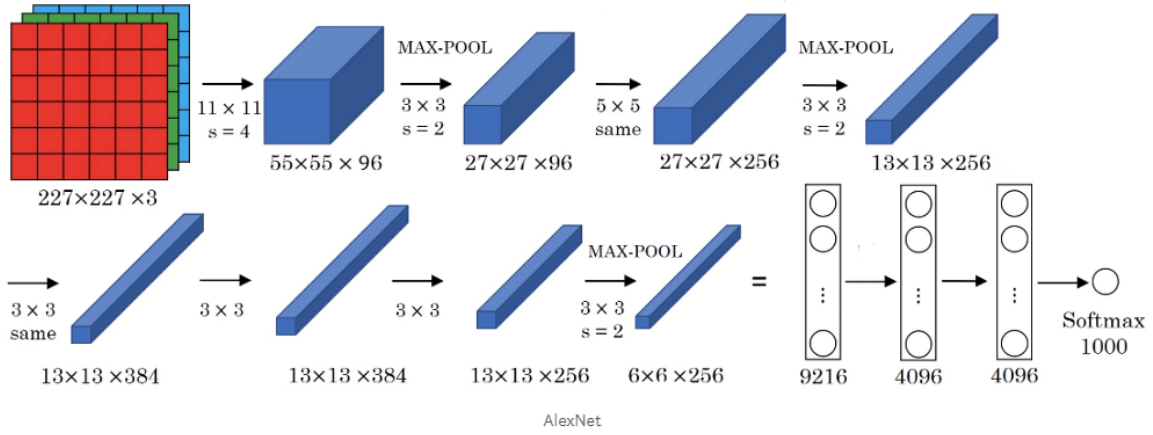
1998 yılında LeCun ve Arkadaşları tarafından yayınlanan ve bu alanda ilk başarılı sonucu veren evrimsel sinir ağı modelidir. Bu model 5 katmanlı evrimsel sinir ağı kullanılarak adres numaraları ve banka çekleri üzerindeki el yazısı rakamlarını sınıflandırmak için geliştirilmiştir. LeNet modeli MNIST (Lecun et al) veri setini kullanarak eğitilmiştir. Bu modelin en önemli özelliği, boyut azaltmada maksimum havuzlama yerine ortalama havuzlama yönetimi kullanmasıdır. LeNet modelinde 10 sınıfa ait 60 bin parametre hesaplanmaktadır. Aktivasyon fonksiyonu olarak sigmoid ve hiperbolik tanjant fonksiyonları kullanılmaktadır. (Lecun et al., 1998). Şekil 9'da bu modelin yapısı görülmektedir.



Şekil 9: LeNet model yapısı (Le Cun et al., 1989)

3.2. AlexNet

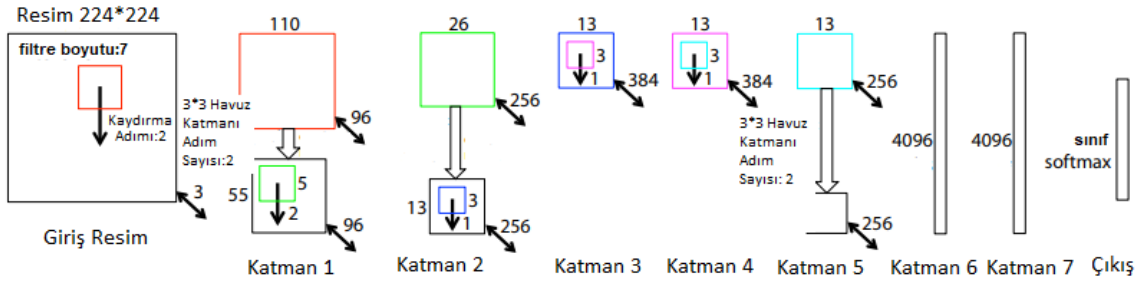
Krizhevsky ve arkadaşları tarafından geliştirilmiş bir modeldir. Bu model 2012 yılında düzenlenen ILSVRC yarışmasında yaklaşık bir milyon veriseti kullanarak, 1000 farklı nesneyi sınıflandırmaktadır. Ayrıca derin öğrenmenin ve evrimsel sinir ağının tekrardan gündeme gelmesini sağlayan modeldir. Evrimsel ve havuzlama katmanlarının bulunmasından dolayı LeNet modeline benzerlik göstermektedir. Bu model toplam 25 katmandan oluşmaktadır. Aktivasyon fonksiyonu olarak ReLU, havuzlama katmanında maksimum havuzlama ve sınıflandırma katmanında softmax aktivasyon fonksiyonu kullanılmaktadır. Bu modelde paralel iki grafik işlem birimi (GPU) kullanılarak nesne tanımda %83,6 oranında başarı sağlamıştır (Krizhevsky et al., 2012) Şekil 10'da AlexNet'e ait katmanlar ve modeli görülmektedir.



Şekil 10: AlexNet model yapısı (Gündüz & Cedimoğlu, 2019)

3.3. ZFNet

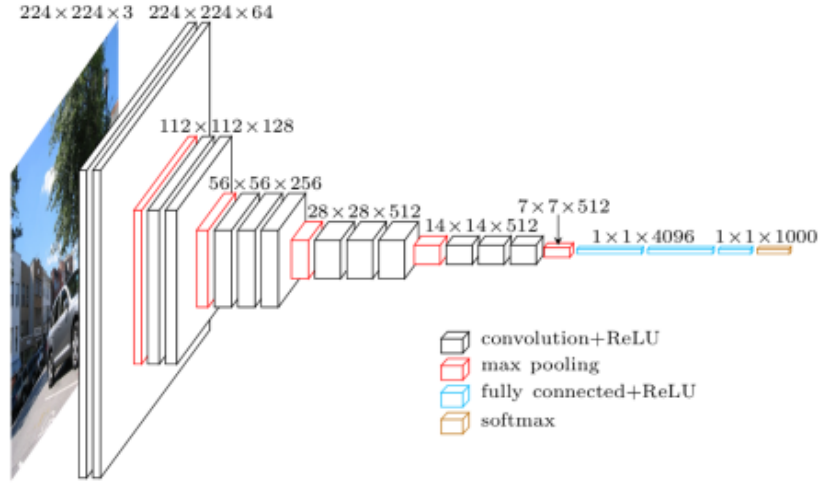
Bu model AlexNet mimarisinden esinlenerek tasarlanmış ve 2013 yılında ILSVRC yarışmasında üstün başarı sağlamıştır. AlexNet'e göre karşılaştırıldığında, nesne tanıma hata oranında önemli ölçüde iyileşme sağlamaktadır. Şekil 11'de ZFNet modeli verilmektedir. Bu modelin AlexNet'e göre farkı, ilk katmandaki filtre boyutu ve adım sayısı düşürülerek, orta katmandaki filtre boyutlarının artırılmasıdır. Bu modelde büyük filtre boyutları ve adım sayıları girdi boyutlarını küçültmekten hesaplamaya yükünü azaltsa da, küçük filtre boyutları ve adım sayıları giriş katmanından daha fazla bilginin korunmasına katkı sağlamaktadır. ZFNet modeli ReLU aktivasyon fonksiyonu, kayıp fonksiyonu ve toplu stokastik gradyan kullanarak eğitilmektedir. Bu model önceki yöntemlerden farklı olarak "Deconvolution Network" olarak adlandırılan bir teknik geliştirmiş ve derin öğrenme modelini daha başarılı bir noktaya taşımıştır. Modelin eğitimi için GTX 580 ekran kartı kullanılmış ve eğitim yaklaşık olarak 12 gün sürmüştür (Zeiler & Fergus, 2014).



Şekil 11: ZFNet model yapısı (Zeiler & Fergus, 2014)

3.4. VGGNet

2014 yılında ILSVRC yarışmasında görsel geometri grubu tarafından geliştirilmiş, grafik işlem birimi (GPU) destekli basit bir evrimsel sinir ağı modelidir (Simonyan & Zisserman, 2015). ILSVRC yarışmasında %89 başarı oranı yakalamış bir derin öğrenme algoritmasıdır (Chen et al., 2018). VGGNet'i diğer modellerden ayıran en önemli farkı, ikili veya üçlü evrişim katmanlarını havuzlama katmanlarının izlemesidir. Şekil 12'de VGGNet modeli görülmektedir. Bu modelde yaklaşık 138 milyon parametrenin hesabı yapılarak, (Zeiler & Fergus, 2014) 1000 sınıflı nesne tanımlanmaktadır. VGGNet, VGG16 ve VGG19 adlı iki farklı model yapısı kullanılmaktadır. VGG16 modelinde 13 evrişim katmanı, 3 tam bağlı katman, havuzlama, aktivasyon fonksiyonu, dropout ve sınıflandırma katmanlarıyla birlikte toplam 41 katman bulunmaktadır. VGG19 modelinde ise VGG16 modeline ek olarak 3 evrişim katmanın eklenmesiyle oluşmaktadır. Bu modelde de diğer modellerde olduğu gibi girişten çıkışa doğru veri boyutu azalırken kanal sayısı artmaktadır (Simonyan & Zisserman, 2015). Ayrıca her evrişim işleminde giriş ve çıkış boyutları aynı olup 3x3 filtre boyutu, havuzlama katmanlarında 2 adım sayısı ve 2x2 filtre boyutu kullanılmaktadır (Krizhevsky et al., 2012).

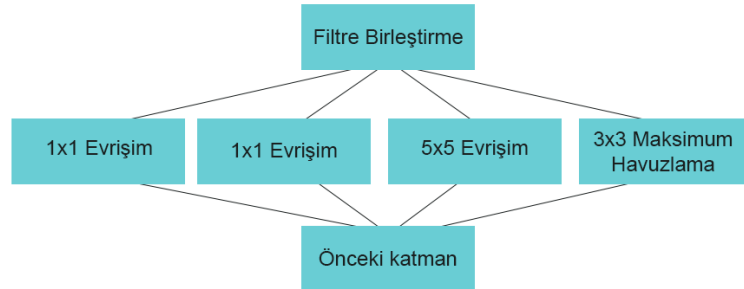


Şekil 12: VGGNet model yapısı

3.5. GoogLeNet

Christian Szegedy ve arkadaşları tarafından 2014 yılında ILSVRC yarışmasını %93,33 başarı oranı ile kazanan evrişimsel sinir ağı modelidir. Inception modülü adı verilen evrişimsel sinir ağı modeli 144 katmandan oluşmaktadır. GoogLeNet, (Szegedy et al., 2015) AlexNet'e göre 12 kat daha az işlem gücüne ihtiyaç duyar (Szegedy et al., 2017). Bu modelde paralel ağ parçaları kullanılarak daha önceki modellerden farklı bir sıralı yaklaşım kullanılmaktadır. Inception modülü evrişim katmanlarında 1x1, 3x3, 5x5 filtre boyutu ve 3x3 boyutunda maksimum havuzlama işlemlerinin paralel gerçekleştirilmesiyle oluşmaktadır (Kızrak & Bolat, 2018). Şekil 13'de temel inception modülü görülmektedir.

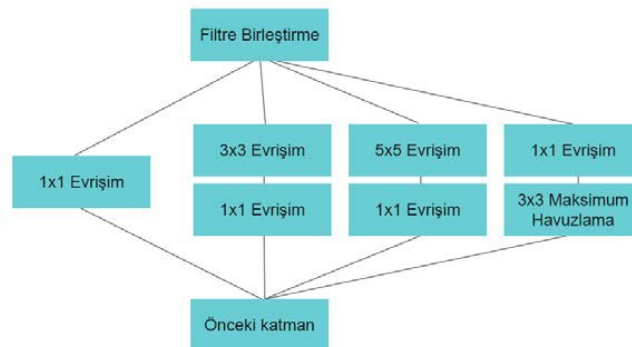
Şekil 13: yapısı



Inception modül

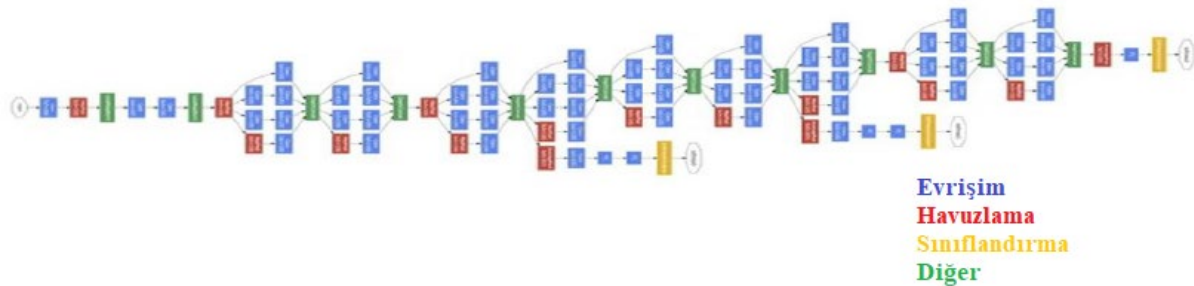
Inception farklı filtre işlemleri ve maksimum havuzlama işlemi uygulanmaktadır. Klasik evrişim ağının her katmanında evrişim işlemi, filtre boyutu ya da havuzlama işlemine gerek duyulurken, inception modülünde tüm bu işlemler paralel olarak gerçekleştirilmektedir. Bu modülde filtre boyutunun artması parametre sayısını ve hesaplama maliyetini artırmaktadır (Szegedy et al., 2017).

modülünde 3 boyutlu evrişim



Şekil 14: Boyut düşürme işleminden sonra Inception yapısı

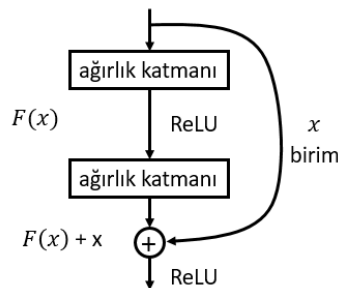
Hesaplama maliyetini düşürmek için, Şekil 14’de gösterildiği gibi Inception modülü içerisine 1x1 boyutunda evrişim katmanı eklenerek, evrişim işlemi gerçekleştirilmektedir. Yapılan bu işlem darboğaz olarak adlandırılmakta ve parametre sayısını 10 kat azaltmaktadır. Bununla birlikte hesap karmaşıklığını azaltarak modelin performansını artırmaktadır. Şekil 15’de GoogLeNet model yapısı görülmektedir.



Şekil 15: GoogLeNet Ağ model yapısı (Szegedy et al., 2015)

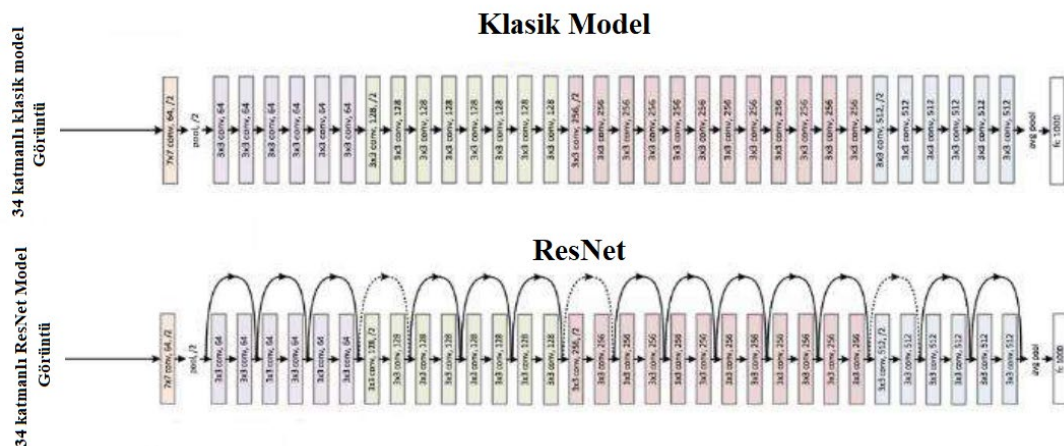
3.6. ResNet

Microsoft Asya araştırma ekibi tarafından 2015 Yılında ILSVRC yarışmasını %95,51 başarı oranı ile kazanan evrimsel sinir ağı modelidir. ResNet daha önceki mimarilere göre daha derin olarak tasarlanmıştır ve toplam 152 katmandan meydana gelmektedir. Bu mimarinin en önemli özelliği kalıntı bloklarından oluşmasıdır. Kalıntı blokta girdi olarak alınan değer sırasıyla evrim-aktivasyon-evrim işlemlerinden geçirilerek, $F(x)$ fonksiyonu adında bir çıktı elde edilmektedir. Daha sonra bu fonksiyona orijinal girdi değeri eklenerek yeni çıktı değeri elde edilmektedir. Bu çıktı $H(x)$ fonksiyonu olarak adlandırılmakta ve $H(x) = F(x) + x$ olarak ifade edilmektedir (He et al., 2016). Ağı oluşturan bu yapı Şekil 16’da verilmektedir.



Şekil 16: Kalıntı öğrenme yapısı

Klasik evrişim işleminde $H(x), F(x)$ fonksiyonuna eşit iken bu yapıda evrişim işleminden sonra girdiye orijinal veri de eklenmektedir. Şekil 17’de klasik model ve ResNet model yapısı görülmektedir.



Şekil 17: Klasik model ve ResNet model yapısı (He et al., 2016)

4. Sonuçlar

Derin öğrenme yöntemlerini kullanarak nesne tanıma, son yıllarda önemli bir araştırma konusu olmuştur. Derin öğrenme yaklaşımının ortaya çıkmasıyla evrimsel sinir ağlarının gizli katmanlı yapıları uygulanmaya başlanmış ve nesne tanıma gibi birçok problemlerin çözümünde derin öğrenme algoritmaları başarılı bir şekilde kullanılmıştır.

Bu çalışmada öncelikle derin öğrenme çalışmalarına destek sağlamakta olan evrimsel sinir ağlarının yapısı ve katmanları hakkında detaylı bilgiler verilmiştir. Derin öğrenme yöntemlerini kullanarak nesne tanıma işlemlerinde başarımının artırılması üzerine yaklaşımlar öne sürülmüştür. Ayrıca derin öğrenme algoritmaları ile ilgili son zamanlarda yapılan çalışmalar hakkında detaylı bilgiler sunulmuş ve başarımlar bakımından karşılaştırılmıştır.

Kaynaklar

- Adler, A., Elad, M., & Zibulevsky, M. (2016). *Compressed Learning: A Deep Neural Network Approach. 1*, 0–3. <http://arxiv.org/abs/1610.09615>
- Altan, G. (2019). DeepGraphNet: Grafiklerin Sınıflandırılmasında Derin Öğrenme Modelleri. *European Journal of Science and Technology, October*, 319–327. <https://doi.org/10.31590/ejosat.638256>
- Angelova, A., Krizhevsky, A., & Vanhoucke, V. (2015). Pedestrian detection with a Large-Field-Of-View deep network. *Proceedings - IEEE International Conference on Robotics and Automation, 2015-June(June)*, 704–711. <https://doi.org/10.1109/ICRA.2015.7139256>
- Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). *Land Use Classification in Remote Sensing Images by Convolutional Neural Networks*. 1–11. <http://arxiv.org/abs/1508.00092>
- Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>
- Ciaburro, G., & Venkateswaran, B. (2017). Neural network with R. In *Packt* (Vol. 91). Packt Publishing.
- Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 22, 1237–1242. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>
- Daş, R., Polat, B., & Tuna, G. (2019). Derin Öğrenme ile Resim ve Videolarda Nesnelerin Tanınması ve Takibi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31(2), 571–581. <https://doi.org/10.35234/fumbd.608778>
- Doğan, F., & Türkoğlu, İ. (2018). Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması The Comparison Of Leaf Classification Performance Of Deep Learning Algorithms. *Sakarya University Journal of Computer and Information Sciences*, 1(April).
- Doğan, F., & Türkoğlu, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*, 10(2), 409–445. <https://doi.org/10.24012/dumf.411130>
- Gündüz, G., & Cedimoğlu, İ. H. (2019). Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini. *Sakarya University Journal of Computer and Information Sciences*, 2(1), 9–17. <https://doi.org/10.35377/saucis.02.01.517930>
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27–48. <https://doi.org/10.1016/j.neucom.2015.09.116>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Hinton, G. E. (2012). A practical guide to training restricted boltzmann machines. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTU, 599–619. <https://doi.org/10.1007/978-3-642-35289-8-32>
- Hinton, G. E., & Salakhutdinov, R. R. (2006). *Reducing the Dimensionality of*. 313(July), 504–507.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. 1–18. <http://arxiv.org/abs/1207.0580>
- <https://developers.google.com/machine-learning/guides/text-classification/step-4>. (Erişim Tarihi:20.08.2020).
- <https://developers.google.com/machine-learning/guides/text-classification/step-4>
- İnik, Ö., & Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri deep learning and deep learning models used in image analysis. *Gaziosmanpaşa Journal of Scientific Research*, ISSN, 85–104.
- Kızrak, M. A., & Bolat, B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma. *Bilişim Teknolojileri Dergisi*, 263–286. <https://doi.org/10.17671/gazibtd.419205>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Handbook of Approximation Algorithms and Metaheuristics*, 1097–1105. <https://doi.org/10.1201/9781420010749>
- Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., & Hubbard, W. (1989). Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11), 41–46. <https://doi.org/10.1109/35.41400>
- Lecun, Yann, Cortes Corinna, Burges, C. J. C. *THE MNIST DATABASE of handwritten digits*. <http://yann.lecun.com/exdb/mnist/>, (Erişim Tarihi:20.08.2020).

- Lecun, Y., Bottou, L., Bengio, Y., & Ha, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, November, 1–46. <https://doi.org/10.1109/5.726791>
- Li, P., Wang, D., Wang, L., & Lu, H. (2018). Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76, 323–338. <https://doi.org/10.1016/j.patcog.2017.11.007>
- Liu hui-bin, Fei, W., Chen Qiang, & Pan Yong. (2016). *RECOGNITION OF INDIVIDUAL OBJECT IN FOCUS PEOPLE GROUP BASED ON DEEP LEARNING*.
- Liu, Lei, Pan, Z., & Lei, B. (2017). *Learning a Rotation Invariant Detector with Rotatable Bounding Box*. <http://arxiv.org/abs/1711.09405>
- Liu, Lingqiao, Shen, C., & Van Den Hengel, A. (2015). The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June, 4749–4757. <https://doi.org/10.1109/CVPR.2015.7299107>
- Pang, Y., Sun, M., Jiang, X., & Li, X. (2018). Convolution in convolution for network in network. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5), 1587–1597. <https://doi.org/10.1109/TNNLS.2017.2676130>
- Ponti, M. A., Ribeiro, L. S. F., Nazare, T. S., Bui, T., & Collomosse, J. (2017). Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask. *Proceedings - 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials SIBGRAPI-T 2017*, 2018-Janua, 17–41. <https://doi.org/10.1109/SIBGRAPI-T.2017.12>
- Salur, M. U., Aydın, İ., & Karaköse, M. (2019). Gömülü Derin Öğrenme ile Tehdit İçeren Nesnelerin Gerçek Zamanda Tespiti. *DÜMF Mühendislik Dergisi*, 10(2), 497–509. <https://doi.org/10.24012/dumf.492433>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Srivastava, N., Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, & Ruslan Salakhutdinov. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Physics Letters B*, 15, 1929–1958. [https://doi.org/10.1016/0370-2693\(93\)90272-J](https://doi.org/10.1016/0370-2693(93)90272-J)
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June, 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep Neural Networks for object detection. *Advances in Neural Information Processing Systems*, February.
- Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10), p9420. <https://doi.org/10.29322/ijssrp.9.10.2019.p9420>
- Tamura, S., & Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network: Four layers versus three. *IEEE Transactions on Neural Networks*, 8(2), 251–255. <https://doi.org/10.1109/72.557662>
- Türkoğlu, M., Hanbay, K., Sıvrıkaya, I. S., & Hanbay, D. (2020). *Derin Evrimsel Sinir Ağı Kullanılarak Kayısı Hastalıklarının Sınıflandırılması*. 9(1), 334–345.
- Xiao, T., Li, H., Ouyang, W., & Wang, X. (2016). Learning deep feature representations with domain guided dropout for person re-identification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, 1249–1258. <https://doi.org/10.1109/CVPR.2016.140>
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1), 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
- Zhang, Q., Bai, C., Chen, Z., Li, P., Yu, H., Wang, S., & Gao, H. (2019). Deep learning models for diagnosing spleen and stomach diseases in smart Chinese medicine with cloud computing. *Concurrency Computation*, March, 1–10. <https://doi.org/10.1002/cpe.5252>
- Zhang, Q., Bai, C., Liu, Z., Yang, L. T., Yu, H., Zhao, J., & Yuan, H. (2020). A GPU-based residual network for medical image classification in smart medicine. *Information Sciences*, 536, 91–100. <https://doi.org/10.1016/j.ins.2020.05.013>