

Derin Öğrenmeye Giriş

Computer Vision

Kursa Başlamadan Önce

- Bu kurs tamamlandığı takdirde giriş düzeyi yapay zeka algoritmaları ve veri analizine temel oluşturabilecek genel bilgileri edinmiş olacaksınız
- Spesifik konular anlaşılması zor ve kişide ders esnasında mantığın oturması kolay olmayacağından bol bol bireysel pratik gerekmektedir
- Slaytlar yazılara boğulmadan görsellerle anlatılacaktır. Bu yüzden ders esnasında not tutulması **son derece** önemlidir
- Konu başlıkları temel düzey algoritmalar için yeterli olduğundan başlıklar araştırılmalı, bol bol uygulama ve teorik bilgiler içeren sitelerde araştırma yapılmalıdır

Bu Eğitimde Neler Öğreneceksiniz

1. Bilgisayarlı Görü alanına derinlemesine giriş yapacak ve görüntü işleme, sayısal analiz, piksel kavramlarını yakından tanıyacağız.
2. Görüntü üzerinde yapılan Data Augmentation, özellik çıkarımı gibi işlemleri göreceğiz, Transfer Learning ile önceden eğitilmiş model kavramını öğreneceğiz.
3. Bilgisayarlı Görü alanında kullanılan Algoritmalar ile tanışacak ve bu algoritmalarından biri olan Evrişimli Sinir Ağı mimarisini tüm detayları ile işleyerek sıkça kullanılan modelleri, aktivasyon fonksiyonlarını tanıyacağız.
4. Popüler Computer Vision alanlarından olan Object Detection ve Image Classification alanlarına hızlı bir giriş yapacak ve bu alanlarda yapılan uygulamaları, kullanılan modelleri, hazır veri setlerini inceleyeceğiz.

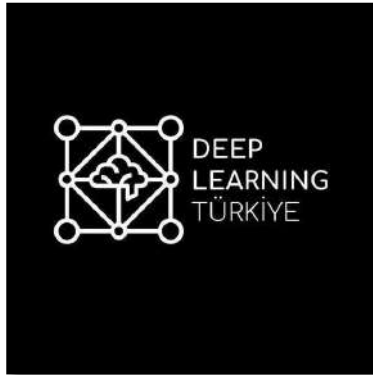
İrem Kömürcü

Github: irem-komurcu

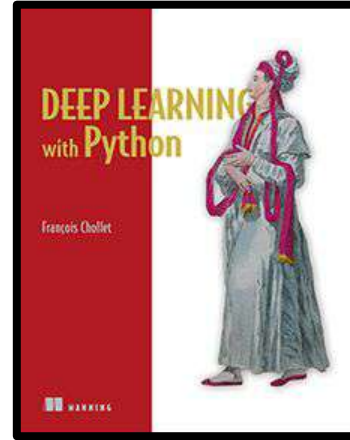
Linkedin: iremkomurcu

Twitter: iremkomurcu

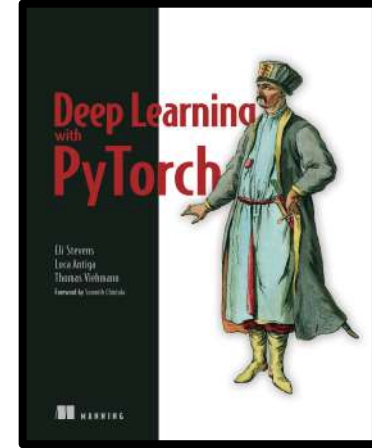
Kaynaklar



Deep Learning Türkiye



Deep Learning with Python



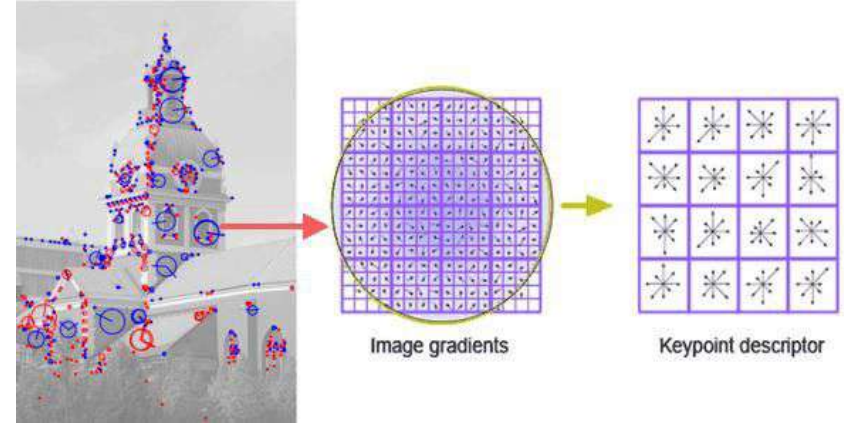
Deep Learning with PyTorch

Görüntü İşleme ve Bilgisayarlı Görü

Bilgisayarlı Görü ve Sayısal Görüntü İşleme

Sayısal görüntü işleme; bilgisayar algoritmaları kullanarak resimler üzerinde görüntü işlemenin gerçekleştirilmesidir.

1. Sisteme bir dizi görüntü girer.
2. Bu görüntülerden özellikleri önceden işlemek ve çıkarmak için bir **Özellik Çıkarıcı** (Feature Extraction) kullanılır.
3. Bir Makine Öğrenimi sistemi, bir modeli eğitmek ve tahminlerde bulunmak için çıkarılan özelliği kullanır.



Görüntü İşleme

Bir Computer Vision(CV) sistemini uygulamaya çalışırken, iki ana bileşeni dikkate almamız gerekir;

- Görüntü elde etme donanımı
- Görüntü işleme yazılımı

Aslında, sistemimiz çevresel değişikliklere karşı değişmez olmalı ve tasarlanan görevini tekrar tekrar yapabilmelidir. Bu gereksinimleri karşılamak için, sistemimizin donanımına veya yazılımına bazı kısıtlamalar uygulamak gerekebilir.



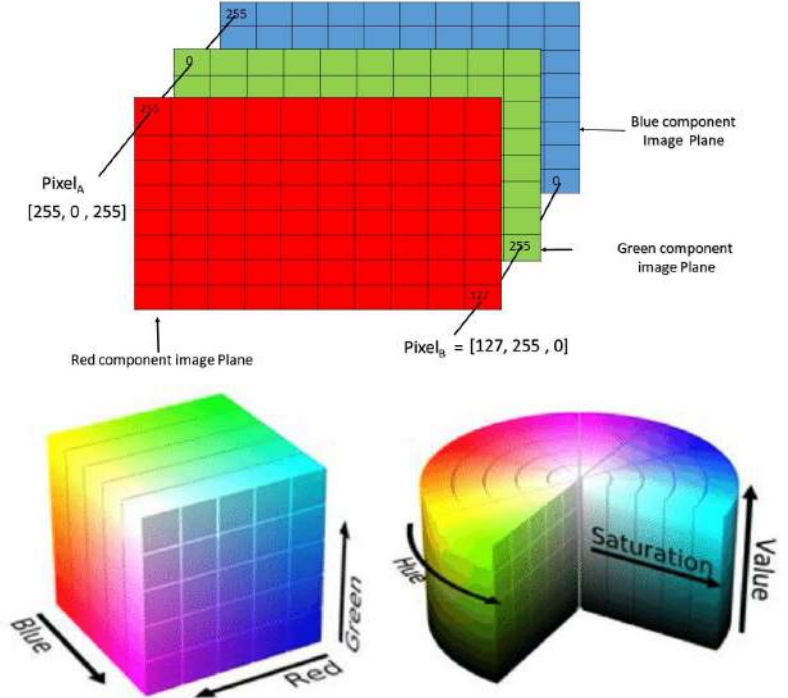
Görüntü İşleme ve Piksel

Bir donanım aygıtından bir görüntü elde edilir. Ardından bir yazılım sistemi, görüntünün içindeki renkleri (Renk Uzayları) sayısal olarak temsil eder.

Bir piksel, görüntüdeki bir noktanın **sayısal** değeridir.

Bir görüntü tipik olarak bir **piksel dizisi** olarak temsil edilir.

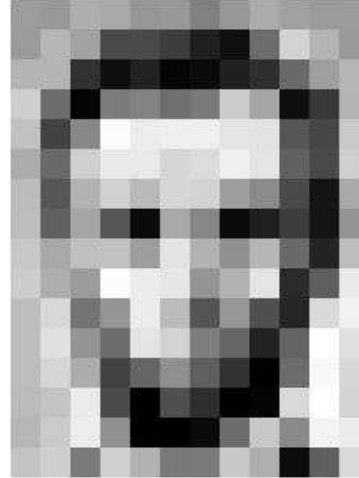
*En ünlü renk alanlarından ikisi **RGB** (Kırmızı, Yeşil, Mavi) ve **HSV'dir** (Ton, Doygunluk, Değer).*



Görüntü İşleme

Görüntü işleme, sayısal sinyal işlemenin bir alt konusu olarak kabul edilir.

Sayısal (Dijital) görüntüler iki boyuttan daha fazla boyutta tanımlanabildiğinden, sayısal görüntü işleme çok boyutlu sistemler şeklinde modellenebilmektedir.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	205
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	96	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

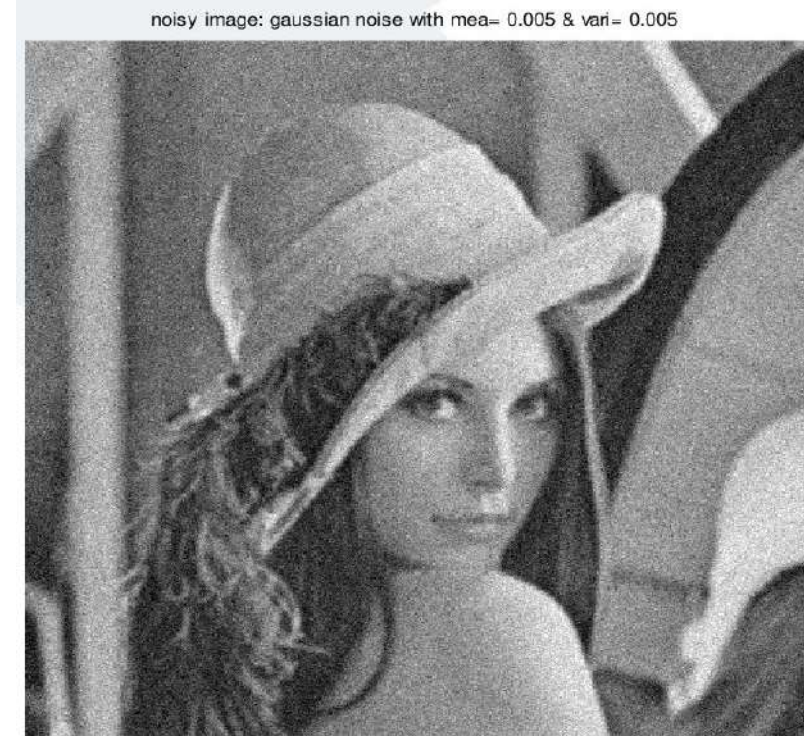
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	205
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	96	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Sinyallerde Gürültü Kavramı

Sinyal işlemede gürültü, bir sinyalin yakalanma, depolama, iletim gibi işlemler sırasında maruz kalabileceği istenmeyen değişiklikler sonucu oluşur.

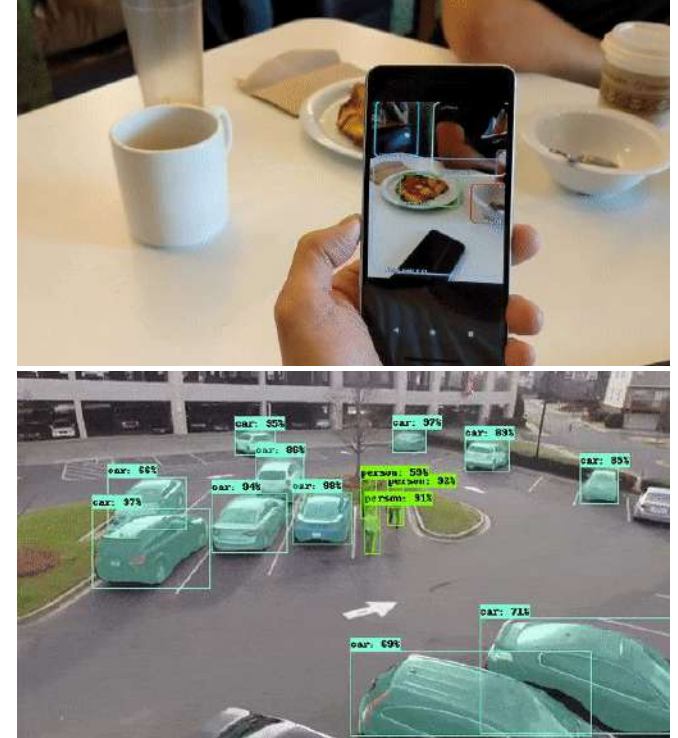
Birçok gürültü çeşidi vardır. Bu gürültüleri azaltmak için gürültü çeşidine özel filtreler uygulanabilir.

Görüntüde gürültü çeşitlerinden biri olan Gaussian noise; piksel düzeyinde, görüntünün Gauss dağılım fonksiyonuna göre karakterize edilerek gürültü oluşmuş biçimidir.



Bilgisayarlı Görü Kullanım Alanları

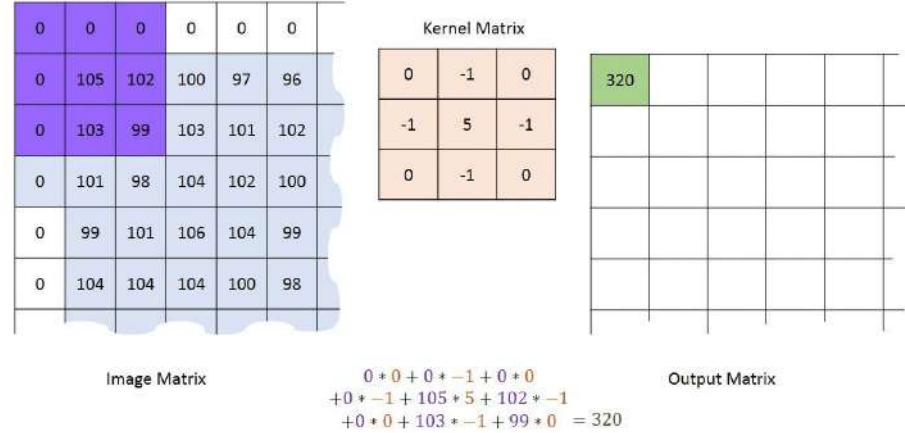
- Otonom Sürüş
- Sosyal Ağ Filtreleme
- Kamera ve Filtreler
- Bioinformatik
- Reklamcılık
- Görüntü İşleme
- Sağlık Hizmetleri
- Gerçek Zamanlı Spor Takibi
- Tarım
- Eğitim



Özellik Çıkarımı (Feature Extraction)

Özellik çıkarımı, görüntüdeki nesneleri net bir şekilde tanımlayan faydalı özellikleri çıkarma fikri etrafında çalışır.

Büyük bir veri kümesini tanımlamak için gereken **kaynak sayısını azaltmayı** içerir, **boyut azaltma** işlemidir.

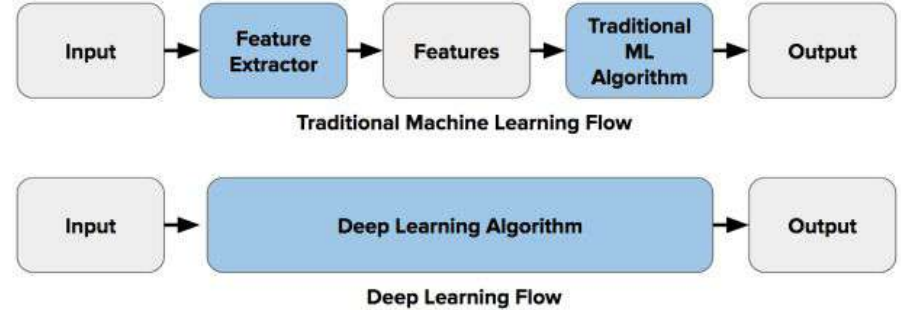


Özellik Çıkarımı (Feature Extraction)

Neden Özellik Çıkarımı Kullanıyoruz?

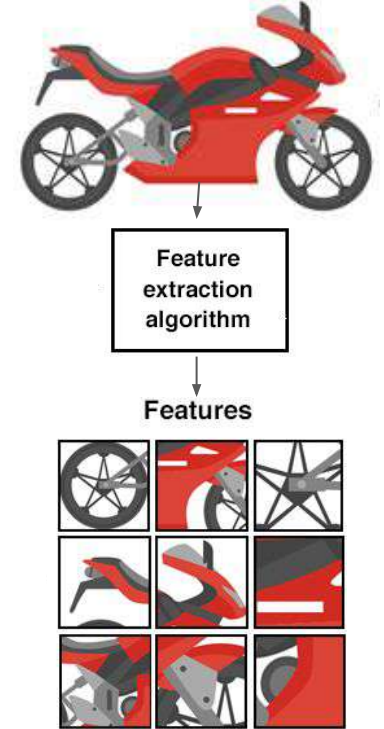
- Çok sayıda değişken ile analiz yapmak genellikle büyük miktarda bellek ve hesaplama gücü gerektirir
- Çok sayıda özellik, modelimizin karmaşıklığını arttıracığından overfite yol açabilir

Özellik çıkarımı; verileri yeterli accuracy ile tanımlar ve bu problemlerin üstesinden gelmek için görüntü ile dönüştürülmek üzere farklı filtre türleri kullanır.



Özellik Çıkarımında 4 Ana Morfoloji

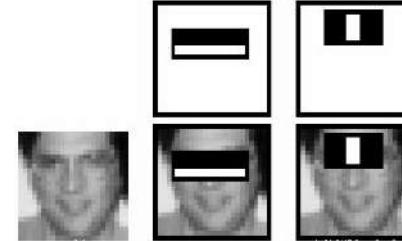
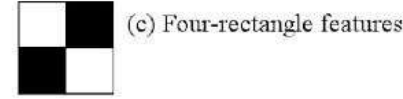
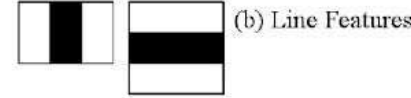
- **Global Özellikler:** Tüm görüntü tek bir özellik olarak analiz edilir ve öznitelik çıkarıcıdan tek bir öznitelik vektörü çıkar
- **İzgara veya Blok Tabanlı Özellikler:** Görüntü farklı bloklara bölünür ve özellikler farklı blokların her birinden çıkarılır
- **Bölge Tabanlı Özellikler:** Görüntü farklı bölgelere bölünür ve bu bölgelerin her birinden bir özellik çıkarılır
- **Yerel Özellikler:** Görüntüde birden çok ilgi noktası tespit edilir ve ilgi noktalarına komşu pikseller analiz edilerek özellikler çıkarılır



Alfred Haar ve Haarcascade Özellikleri

Haar Benzeri Özellikler (Haar-like Features):

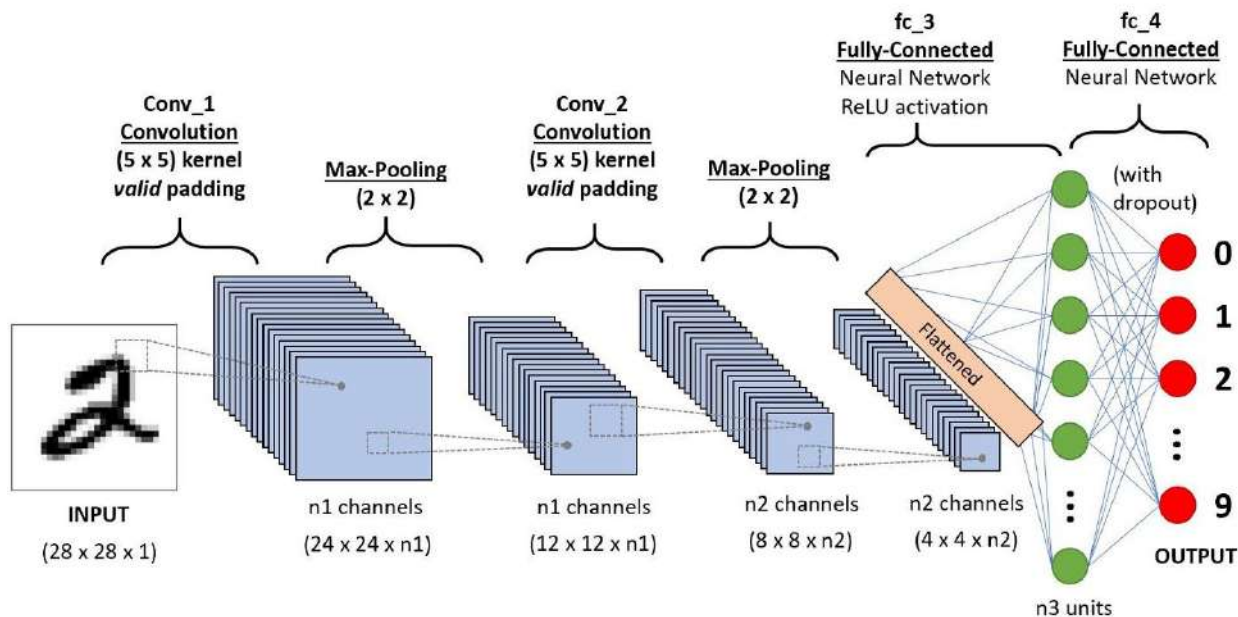
- **Kenar Özellikleri;** Görüntü üzerinde belirli bir alan koyu alandan oluşuyor ve belirli bir alan açık renklerden oluşuyor ise kenar özelliği olduğunu belirtmektedir
- **Çizgi Özellikleri;** Görüntü üzerinde sırasıyla açık, kapalı, açık renklerden oluşuyor ise çizgi özelliği vardır
- **Dört Kare Özellikleri;** Çaprazlama olarak kare şeklinde koyu ve açık tonlar çapraz bir şekilde bulunuyor ise dört kare özelliğini belirtmektedir



Convolutional Neural Network

CNNs

Convolutional Neural Network

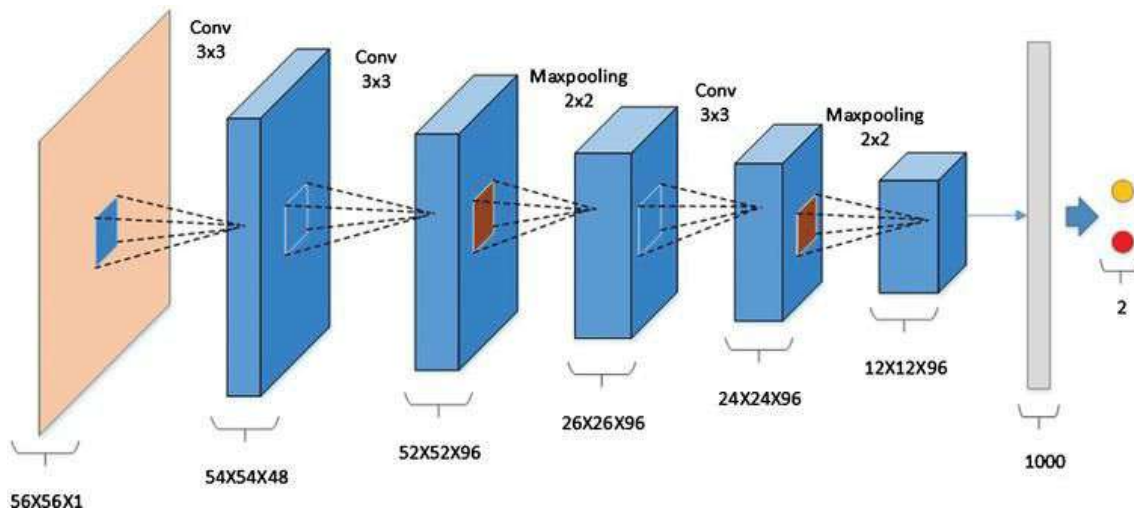


Simple Convolutional Neural Network Notation

CNN ve Katmanlar

CNN katmanları;

- Convolutional Layer
- Non-Linearity Layer
- Pooling
(Downsampling Layer)
- Stride
- Padding
- Flattening Layer
- Fully-Connected Layer

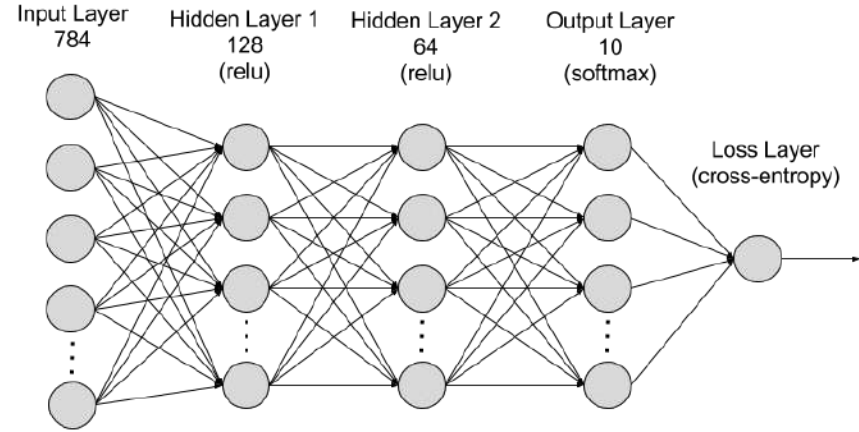


Giriş Katmanı (Input Layer)

CNN'deki giriş katmanı, görüntü verilerini içermelidir. Görüntü verileri çok boyutlu matris ile temsil edilir ve input layer içinde **tek bir vektöre** yeniden şekillendirilir.

Örneğin 28x28 boyutunda bir görüntünüz var. Bu görüntü işlemlere girmeden önce $28 \times 28 = 784$ şeklinde 784x1 vektör haline getirilir.

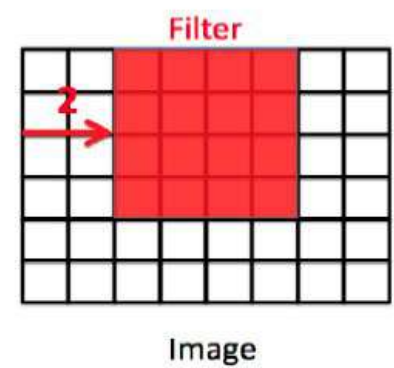
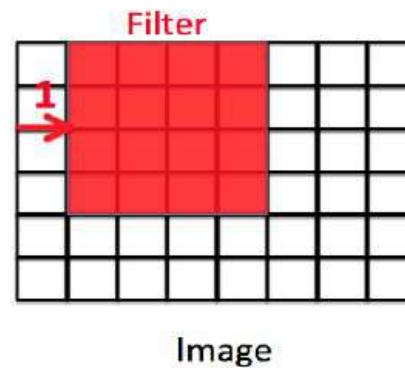
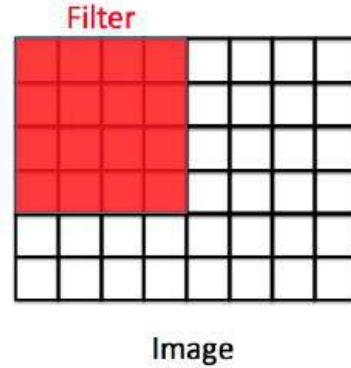
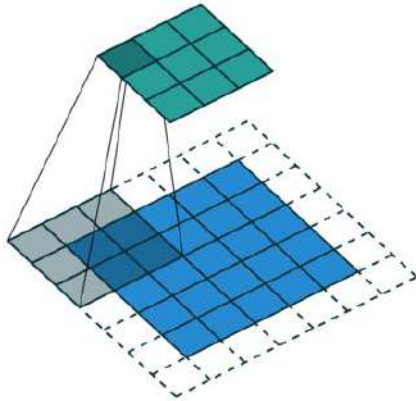
m eğitim örneğiniz varsa, boyutunuz $784 \times m$ olacaktır.



Adım (Stride)

Kayma başına kat edilen satır ve sütun sayısını **adım** olarak ifade ederiz.

Adım sayısı 1 olmak zorunda değildir. Bazen, ya hesaplama verimliliği için ya da altörnekleme yapmak istediğimiz için, ara konumları atlayarak penceremizi aynı anda **birden fazla** öğeyi hareket ettiririz.

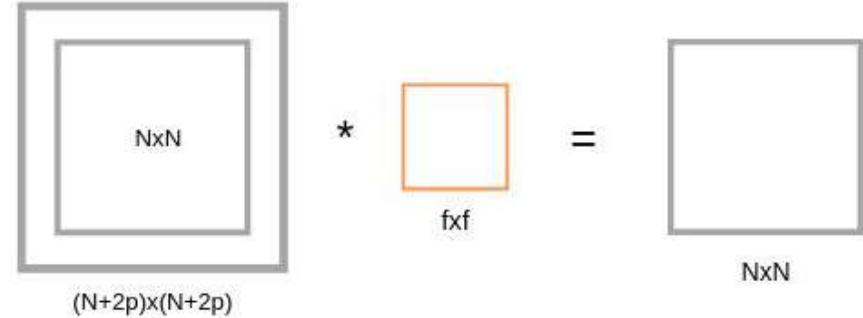
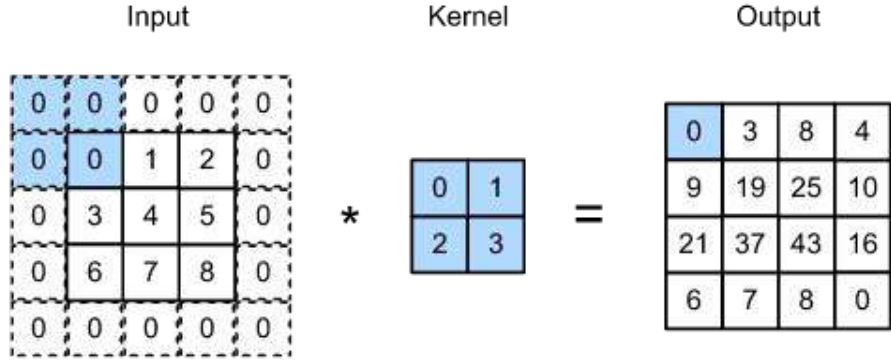


Padding

Evrişimli katmanları uygularken problem olan bir konu, görüntümüzün çevresinde piksel kaybetme eğiliminde olmamızdır.

Özellikle birçok ardışık evrişim katmanı uyguladığımızda piksel kayıpları yoğun şekilde yaşanabilir.

Bu soruna basit bir çözüm, girdi görüntümüzün sınırının etrafına fazladan dolgu pikselleri eklemek, böylece görüntünün etkin boyutunu artırmaktır.

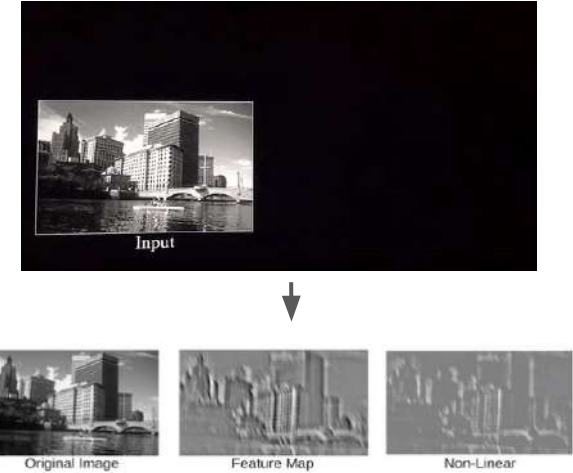
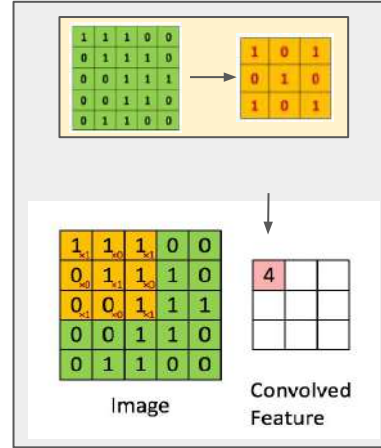


Convolution Layer

Evrişim katmanı (CONV), giriş katmanını boyutlarına göre tararken evrişim işlemlerini gerçekleştiren filtreler kullanır.

Elde edilen çıktı, öznitelik haritası veya aktivasyon haritası olarak adlandırılır.

*Matematikte ve özellikle fonksiyonel analizde **konvolüsyon ya da evrişim**, bir fonksiyonun şeklinin başka fonksiyon tarafından nasıl modifiye edildiğini gösteren bir integral işlemdir.*



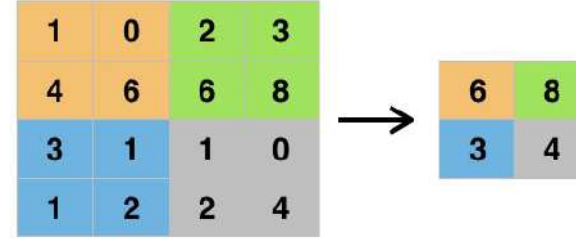
$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$

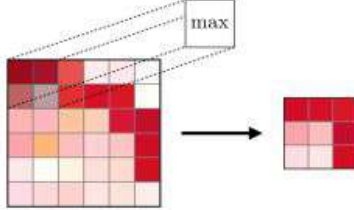
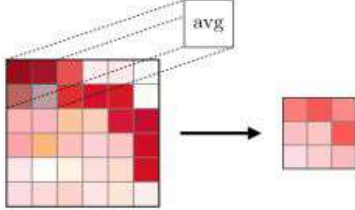
Pooling Katmanı

Özellik haritalarının boyutlarını azaltmak için kullanılır. Böylece ağda öğrenilecek parametre sayısını ve yapılan hesaplama miktarını azaltır.

Pooling katmanı Konvülasyon işlemi sonrası özellik haritasının özelliklerini özetler.

Maksimum pooling maksimum değerleri alırken, Ortalama pooling ortalama değerleri alır.

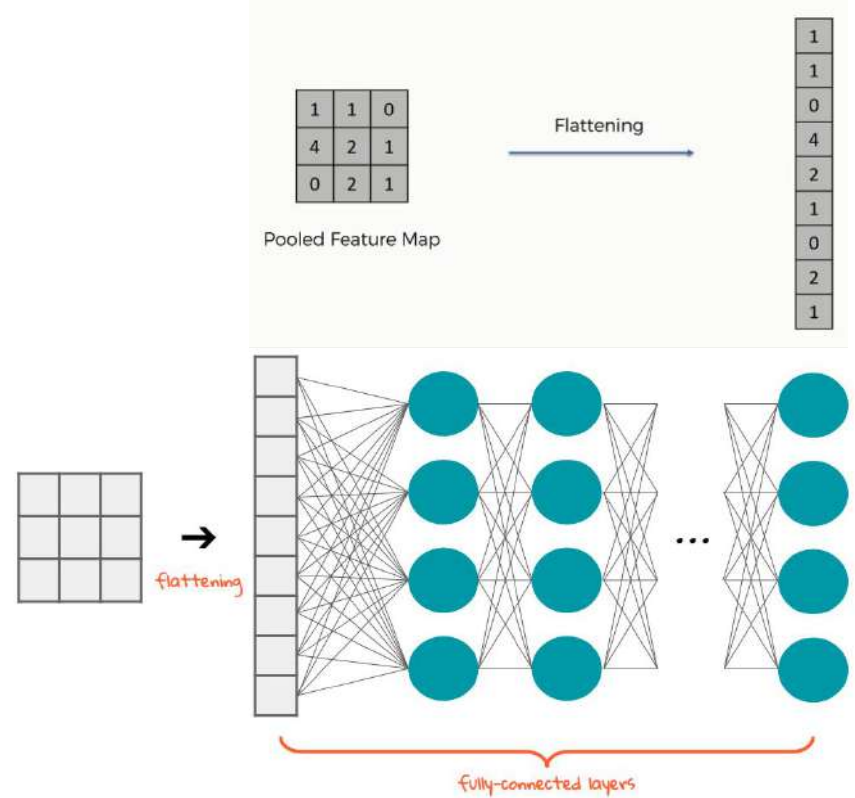


	Max pooling	Average pooling
Purpose	Each pooling operation selects the maximum value of the current view	Each pooling operation averages the values of the current view
Illustration		
Comments	<ul style="list-style-type: none"> - Preserves detected features - Most commonly used 	<ul style="list-style-type: none"> - Downsamples feature map - Used in LeNet

Flattening Katmanı

Özellik Haritası elde edildikten sonra, bir sonraki adım onu düzleştirmektir.

Düzleştirme, pooling işlemine girmiş özellik haritası matrisinin tamamının tek bir sütuna dönüştürülmesini ve daha sonra işlenmek üzere sinir ağını beslenmesini içerir.

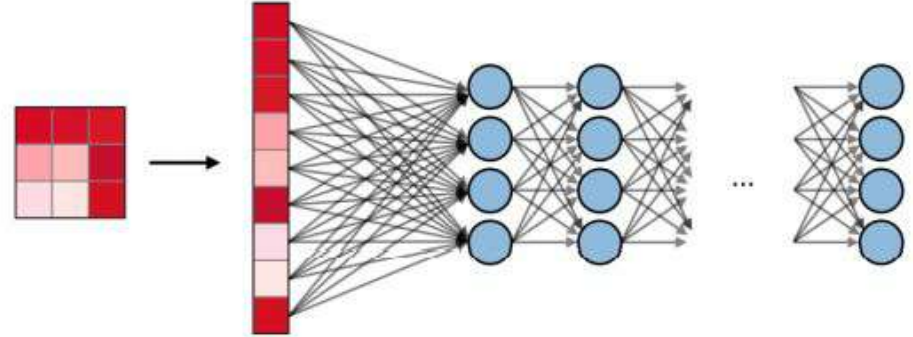
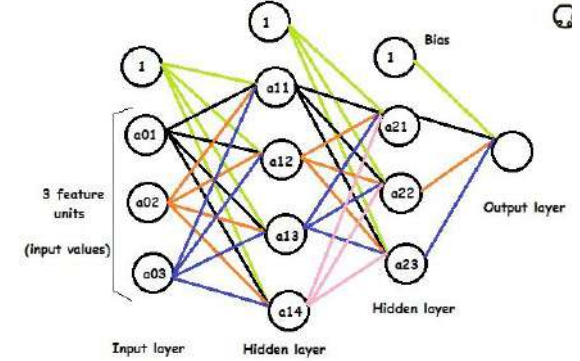


Tam Bağlı Katman (Fully Connected Layer)

Tam bağlı katman (FC), bir katmandaki tüm girdilerin, bir sonraki katmanın her birimine bağlandığı katmanlardır. Standart bir yapay sinir ağı katmanıdır.

Tam bağlı katmanlar sınıf skorları gibi hedefleri optimize etmek için kullanılabilir.

Tam bağlı katmanların en büyük avantajı; “yapıdan bağımsız” olmalarıdır. **Girdi** hakkında yapılması gereken özel varsayımları yoktur

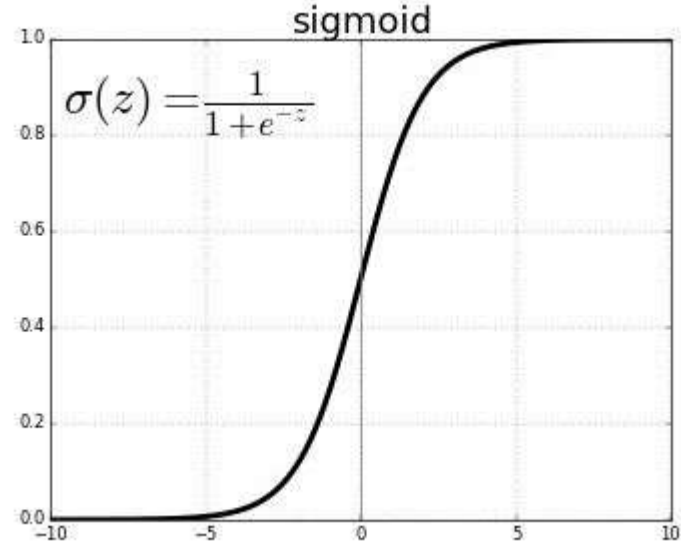


Yaygın Kullanılan Aktivasyon Fonksiyonları

Sigmoid Fonksiyonu

Doğrusal olmayan(**non-linear**) bir fonksiyondur. Karar vermeye yönelik olasılıksal bir yaklaşıma sahip bir yapıda olan bu fonksiyonun değer aralığı $[0,1]$ arasındadır.

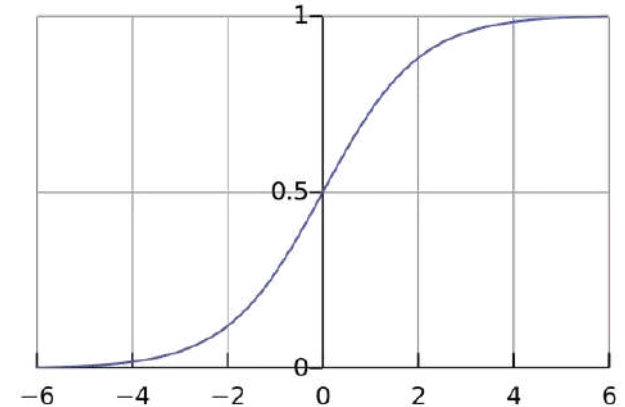
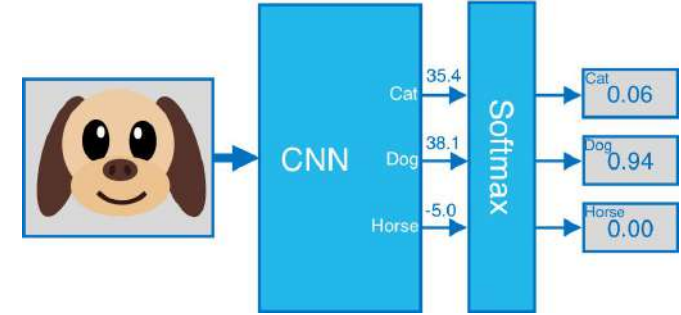
- Bu fonksiyonlardaki x değerinin değişimleri y değeri üzerinde ciddi değişikliğe neden olacaktır
- x (-sonsuz, +sonsuz) arasında ne kadar değişirse değişsin aktivasyon değeri her zaman 0 ile 1 arasında olacaktır



Softmax

Softmax, Çok sınıflı bir sınıflandırma probleminde, çıktı olarak her sınıfa ait olasılık sonucu döndürür.

Çok sınıflı işlemlerde ağınız bir **softmax aktivasyon fonksiyonu** ile bitmelidir, böylece N çıkış sınıfı üzerinde bir olasılık dağılımı verecektir.



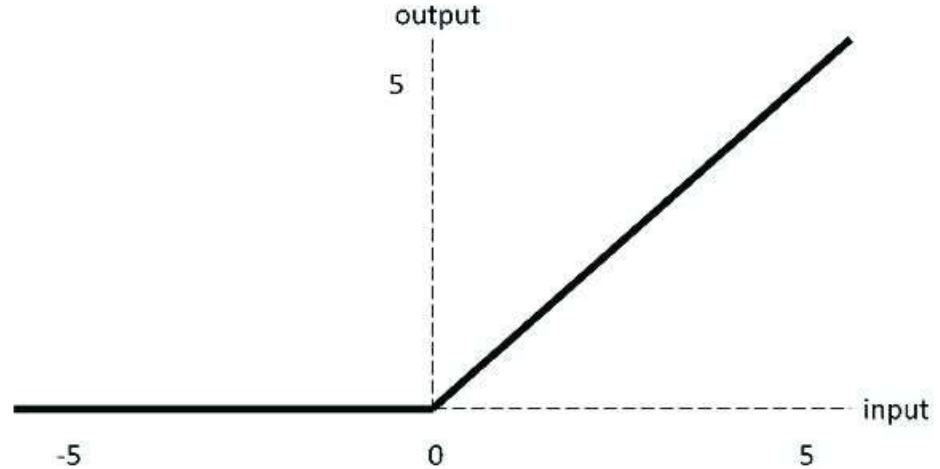
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

ReLU (Düzeltilmiş Doğrusal Birim)

Doğrusal olmayan(**non-linear**) bir fonksiyondur. ReLU fonksiyonunun diğer aktivasyon fonksiyonlarına göre ana avantajı, tüm nöronlarını aynı anda aktive etmemesidir.

Aralık: [0 - sonsuz)

ReLU aktivasyon fonksiyonuna verilen herhangi bir negatif giriş değerini sıfıra çevirir.



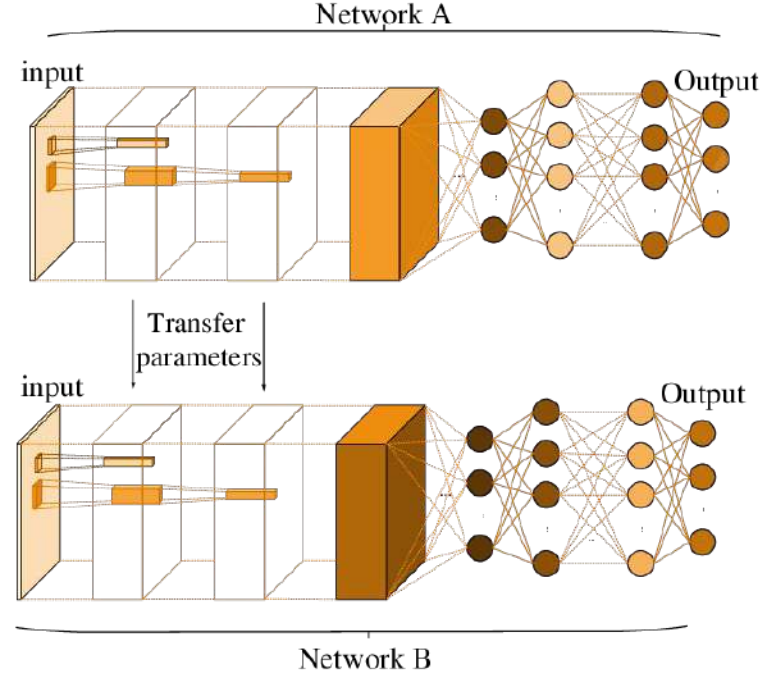
Ön Eğitilmiş CNN Modelleri

Ön Eğitilmiş Model Nedir?

Önceden eğitilmiş bir model, daha önce büyük bir veri kümesinde, tipik olarak büyük ölçekli bir görüntü sınıflandırma görevinde eğitilmiş kayıtlı bir ağıdır. Ya önceden eğitilmiş modeli olduğu gibi kullanırsınız ya da bu modeli belirli bir göreve göre özelleştirmek için transfer öğrenimini kullanırsınız.

- Xception
- Inception V3
- ResNet50
- VGG16
- VGG19
- MobileNet

Listelenen modellerin tümü **ImageNet** veri kümesinde önceden eğitilmiştir

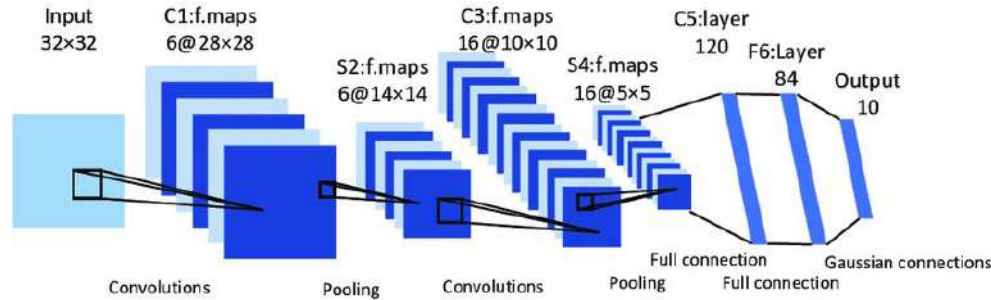


LeNet-5

Yann LeCun ve ekibi 1998 yılında yayınladığı makale, görüntü işleme yöntemlerinin nasıl Derin Ağlar üzerinde kullanabileceğini anlattı. LeNet-5 olarak adlandırılan bu yapı **ilk başarılı sonucu veren Evrişimli Sinir Ağı** olmuştur.

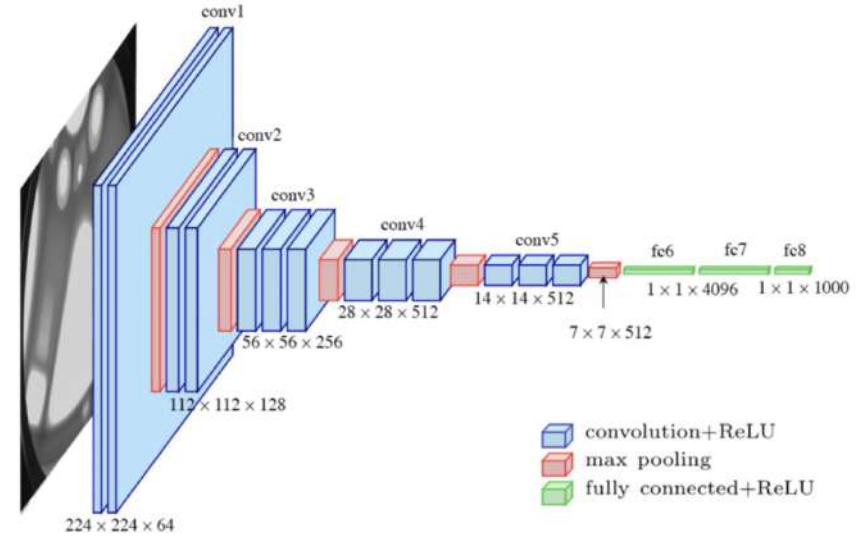
LeNet-5 diğer modellerden farklı olarak;

- Pooling aşamasında max-pooling yerine average-pooling işlemi yapmaktadır
- Aktivasyon fonksiyonu olarak sigmoid ve hiperbolik tanjant kullanılmaktadır



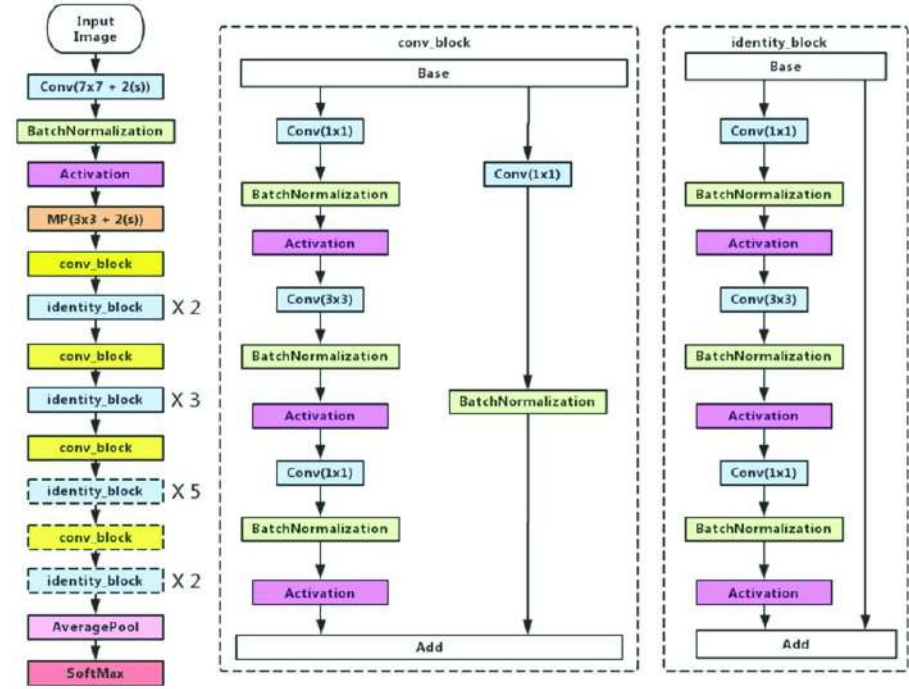
VGG16

- Basit bir ağı model olup diğer ağlardan en önemli farkı evrişim katmanlarının 2'li ya da 3'lü kullanılmasıdır
- İki FC katmanı çıkışında 1000 sınıflı softmax başarımı hesaplanır
- Yaklaşık 138 milyon parametre hesabı yapılmaktadır
- Diğer modellerde de olduğu gibi girişten çıkışa doğru matrislerin yükseklik ve genişlik boyutları azalırken derinlik değeri artmaktadır



ResNet50

- ResNet50, 50 katman derinliğine sahip bir Evrişimli Sinir ağıdır
- Ağ 224x224 giriş boyutuna sahiptir
- ResNet diğer ağlardan farklı olarak artık değerlerin (residual value) sonraki katmanları besleyen blokların (residual block) modele eklenmesiyle oluşmaktadır

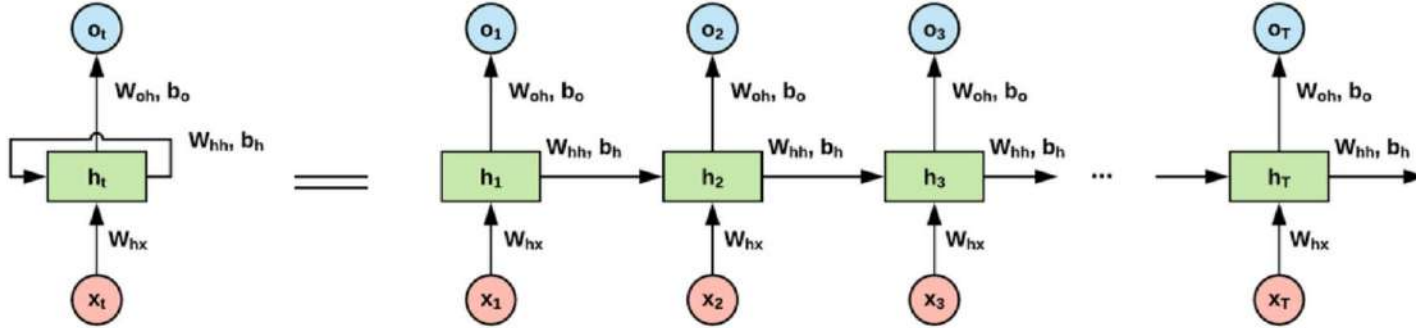


Recurrent Neural Networks

RNNs

Recurrent Neural Network

- RNN, dahili bir döngüye sahip bir tür sinir ağıdır
- Karşılıklarına çıkan her girdiyi birbirinden bağımsız olarak işlerler ve girdiler arasında durum bilgisi tutmazlar
- RNN ile çıktığı başka bir sinir ağı besleyebilir ve görüntü verilerini dizilerle çalıştırabiliriz

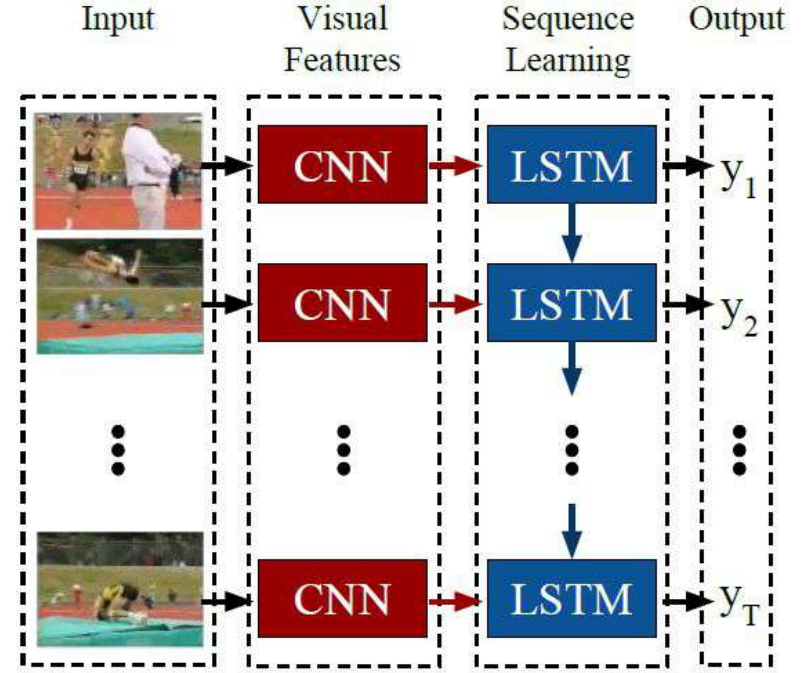


RNN ve Bilgisayarlı Görü

Videolardaki eylemleri anlamak için, yalnızca bir tane değil, birden çok görüntüyü analiz etmek gereklidir.

CNN'leri ve RNN'leri birleştirmek, videolar üzerine çalışmamıza olanak sağlar.

CNN ve RNN mimarileri farklı sorunları ele almak için kullanılıyor gibi görünebilirler, ancak bazı veri türlerinin her iki mimari tarafından da işlenebilir. Örneğin, RNN'ler dizileri girdi olarak kullanır. Diziler, bir dizi görüntü olan videolar da olabilir.



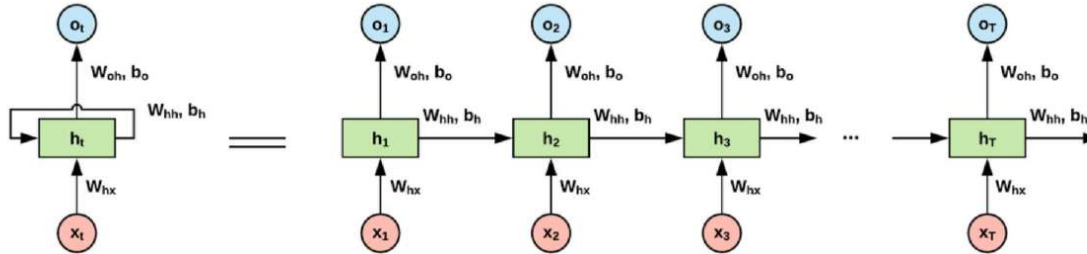
Recurrent Neural Network

Avantajlar;

- Herhangi bir uzunluktaki girdiyi işleme imkanı
- Model boyutunun, girdi boyutuyla artması
- Hesaplama, geçmiş bilgileri hesaba katar
- Ağırlıklar zaman içinde paylaşılır

Dezavantajlar;

- Hesaplamanın yavaş olması
- Uzun zaman öncesine ait bilgilere erişim zorluğu
- Mevcut durum için gelecekteki herhangi bir girdi düşünülemez



Konvolüsyonel Ağlarda Kullanılan Diğer Teknikler

Data Augmentation

Data Augmentation

Veri analizinde veri arttırma, halihazırda var olan verilerin biraz değiştirilmiş kopyalarını veya mevcut verilerden yeni oluşturulan sentetik verileri ekleyerek veri miktarını arttırmak için kullanılan bir tekniktir.

Bir makine öğrenimi modelini eğitirken overfitting oluşmamasına yardımcıdır.

Veri analizinde aşırı örnekleme (oversampling) ile yakından ilgilidir.

Orjinal	Çevirme	Rotasyon (Yönlendirme)	Rastgele kırpmak/kesme
			
<ul style="list-style-type: none"> Herhangi bir değişiklik yapılmamış görüntü 	<ul style="list-style-type: none"> Görüntünün anlamının korunduğu bir eksene göre çevrilmiş görüntü. 	<ul style="list-style-type: none"> Hafif açılı döndürme Yanlış yatay kalibrasyonu simüle eder 	<ul style="list-style-type: none"> Görüntünün bir bölümüne rastgele odaklanma Arka arkaya birkaç rasgele kesme yapılabilir

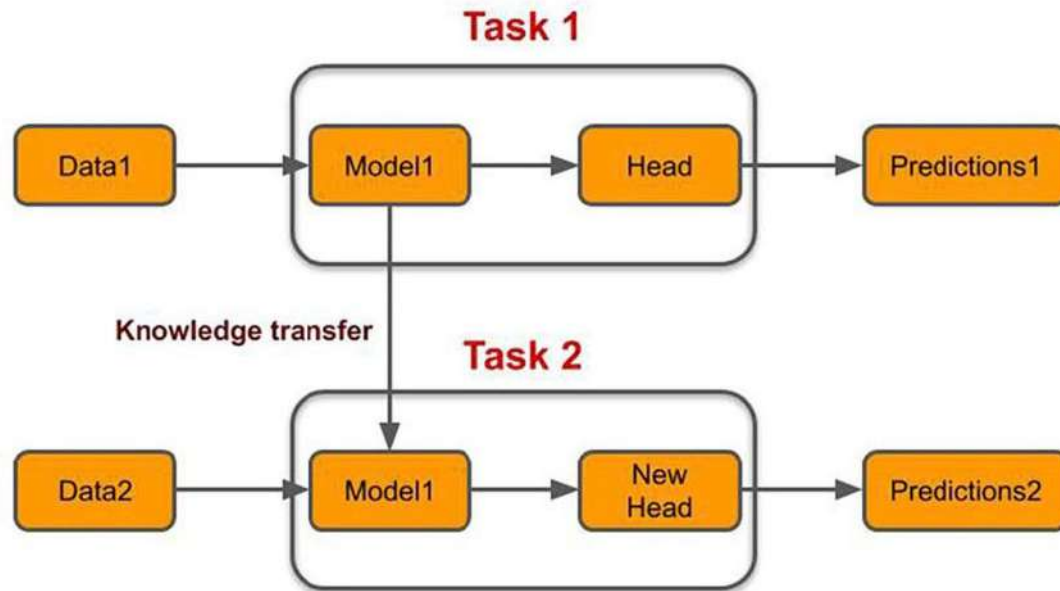
Renk değişimi	Gürültü ekleme	Bilgi kaybı	Kontrast değişimi
			
<ul style="list-style-type: none"> RGB'nin nüansları biraz değiştirilmesi Işığa maruz kalırken oluşabilecek görüntü 	<ul style="list-style-type: none"> Gürültü ekleme Girdilerin kalite değişkenliğine daha fazla toleranslı olması 	<ul style="list-style-type: none"> Yok sayılan görüntüler Görüntünün parçalardaki olası kayıplarını kopyalanması 	<ul style="list-style-type: none"> Gün içindeki ışık ve renk değişimim kontrolü

Transfer Learning

Transfer Learning

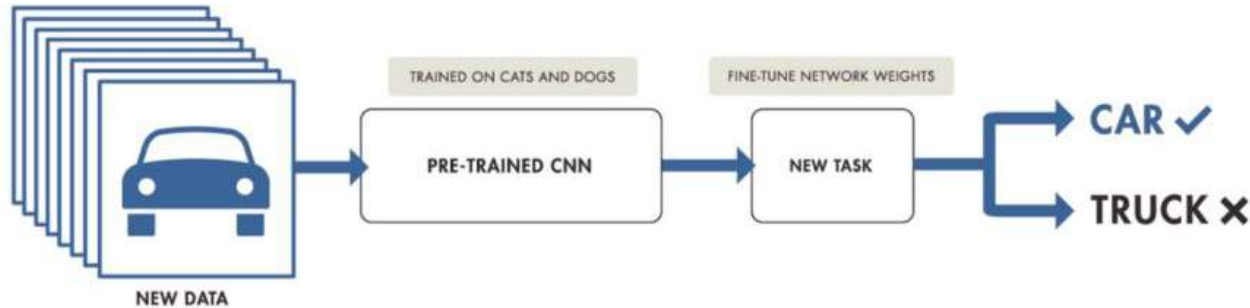
Transfer öğrenimi, bir problemde öğrenilen özellikleri almak ve bunları yeni, **benzer** bir problem üzerinde kullanmaktan oluşur.

Transfer öğrenimi genellikle, veri kümenizin tam ölçekli bir modeli sıfırdan eğitmek için çok az veriye sahip olduğu görevler için yapılır.



Transfer Learning İş Akışı

- Önceden eğitilmiş bir modelden katmanlar alınır
- Gelecekteki eğitim turlarında içerdikleri herhangi bir bilgiyi yok etmemek için katmanlar dondurulur
- Dondurulmuş katmanların üzerine yeni, eğitilebilir katmanlar eklenir. Bu işlem ile, eski özellikleri yeni bir veri kümesinde tahminlere dönüştürmeyi öğreneceklerdir
- Veri kümesindeki yeni katmanlar eğitilir



Transfer Learning ve Katman Dondurma

Transfer Learning'i kullanarak, önceden eğitilmiş olanlar olmak üzere bazı katmanları dondurmaları ve yalnızca eklenenleri eğitmeli ve ağ için anlamlarını karıştırmadan ağırlıkları ayarlamak için öğrenme oranını azaltmalısınız.

Bir katmanı dondurmak, ağırlıklarının değiştirilmesini önler. Bu teknik genellikle, temel modelin (başka bir veri kümesi üzerinde eğitilmiş) dondurulduğu transfer öğreniminde kullanılır.

```
# Make a model with 2 layers
layer1 = keras.layers.Dense(3, activation="relu")
layer2 = keras.layers.Dense(3, activation="sigmoid")
model = keras.Sequential([keras.Input(shape=(3,)), layer1, layer2])

# Freeze the first layer
layer1.trainable = False

# Keep a copy of the weights of layer1 for later reference
initial_layer1_weights_values = layer1.get_weights()

# Train the model
model.compile(optimizer="adam", loss="mse")
model.fit(np.random.random((2, 3)), np.random.random((2, 3)))

# Check that the weights of layer1 have not changed during training
final_layer1_weights_values = layer1.get_weights()
np.testing.assert_allclose(
    initial_layer1_weights_values[0], final_layer1_weights_values[0]
)
np.testing.assert_allclose(
    initial_layer1_weights_values[1], final_layer1_weights_values[1]
)
```

1/1 [=====] - 0s 1ms/step - loss: 0.0846

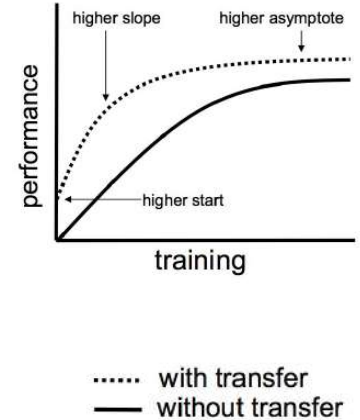
Transfer Learning ve Yararları

Daha iyi başlangıç modeli: Diğer öğrenme türlerinde, herhangi bir bilginiz olmadan bir model oluşturmanız gerekir. Transfer öğrenimi daha iyi bir başlangıç noktası sunar ve bazı seviyelerde görevleri eğitime bile gerek duymadan gerçekleştirebilir.

Daha yüksek öğrenme oranı: Problem zaten benzer bir görev için eğitildiğinden, transfer öğrenme eğitim sırasında daha yüksek bir öğrenme oranı sunar.

Eğitimden sonra daha yüksek doğruluk: Daha iyi bir başlangıç noktası ve daha yüksek öğrenme oranı ile transfer öğrenimi, daha yüksek bir performans düzeyinde yakınsayan ve daha doğru çıktı sağlayan bir makine öğrenimi modeli sağlar.

Daha hızlı eğitim: Öğrenme, önceden eğitilmiş bir modelden yararlandığı için geleneksel öğrenme yöntemlerinden daha hızlı istenen performansı elde edebilir.



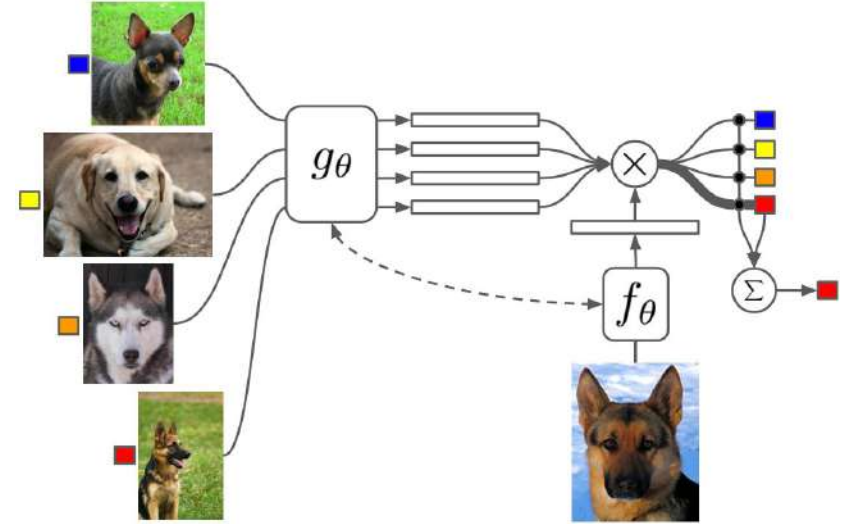
One-shot Learning

One-shot Learning

Tek seferde öğrenme, her sınıf için bir örneğin (veya çok az sayıda örneğin) verildiği, bir model hazırlamak için kullanılır.

One-Shot Learning (Tek seferde öğrenme) çoğunlukla nesne sınıflandırma problemi ile ilgilenir.

Çoğu makine öğrenimi tabanlı nesne sınıflandırma algoritması, yüzlerce görüntü ve çok büyük veri kümeleri üzerinde eğitim gerektirirken, one-shot learning, bir veya yalnızca birkaç eğitim örneğinden nesne kategorileri hakkında bilgi öğrenmeyi amaçlar.



Nesne Tanıma (Object Detection)

Nesne Tanıma Nedir?

Nesne algılama yaparak bir görüntüdeki sınırlayıcı kutuya sahip nesnelerin varlığını ve konumlanmış nesnelerin türlerini veya sınıflarını bulunabilir.

Girdi: Fotoğraf gibi bir veya daha fazla nesne içeren bir görüntü.

Çıktı: Bir veya daha fazla sınırlayıcı kutu ve her sınırlayıcı kutu için bir sınıf etiketi.



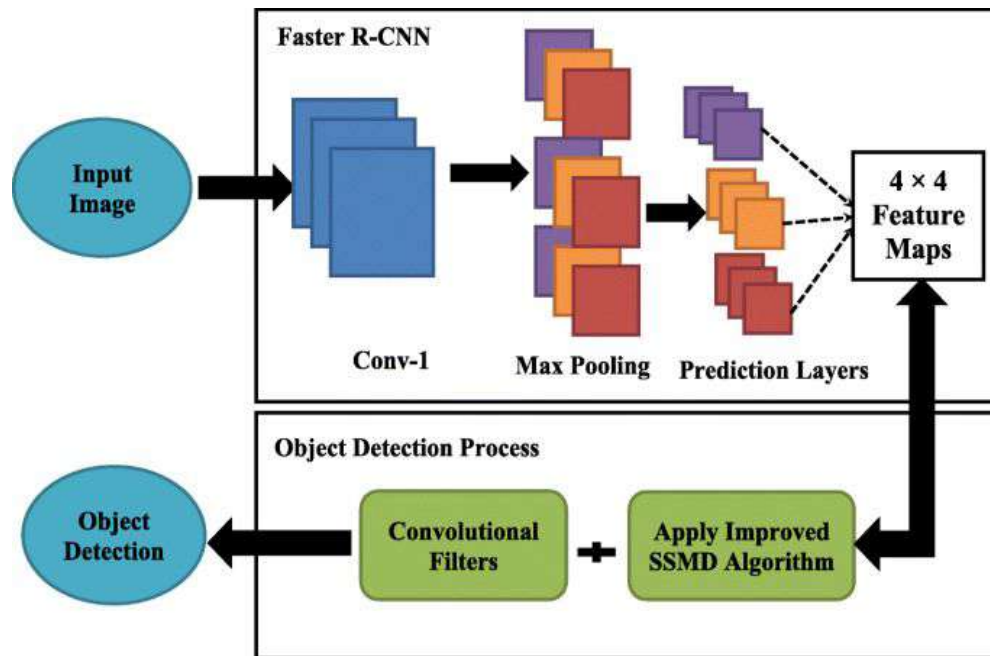
Nesne Tanıma ve Kullanım Alanları

- **Reklamcılık:**
Brand & Logo detection, Human Behavior Analysis
- **Otonom Sürüş:**
Pedestrian Detection, Traffic Light Detection
- **Defense & Space Exploration:**
Remote Sensing Target Detection
- **Sağlık:**
Medical Imaging & Diagnostics
- **Robotik:**
Robot Vision & Manipulation
- **Güvenlik:**
Video Surveillance, Traffic Monitoring

Nesne Tanıma Mimarileri

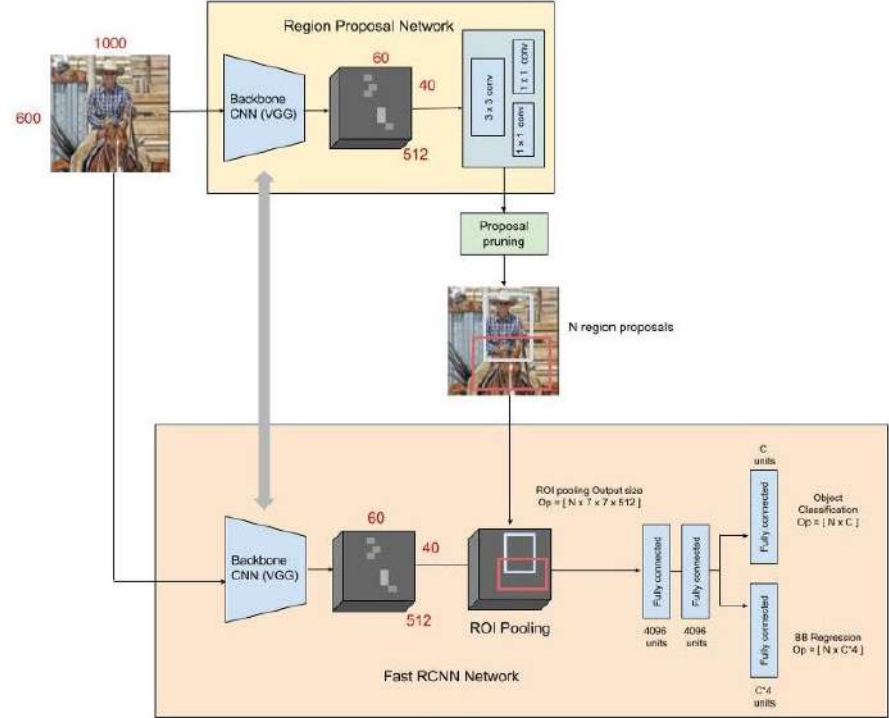
Nesne Tanıma Mimarileri

- R-CNN
- Fast R-CNN
- Faster R-CNN
- YOLO
- SSD
- RefineNet
- RefineDet



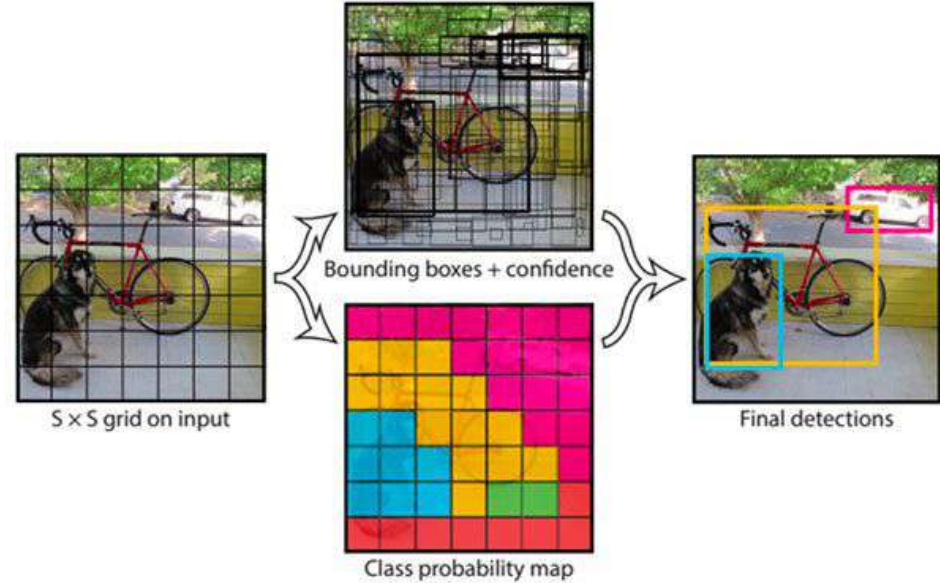
Faster R-CNN

- **Region Proposal Network** algoritmasını kullanarak bölgeler oluşturur
- Her bölgede nesne olduğunu varsayarak işlem yapar ve bölgelere tarafsız puanlar atar
- Bu işlemleri yaparken yoğunluk, renk, pixel sırası gibi kriterlere bakar
- ReLu aktivasyon fonksiyonu kullanarak Özellik Haritası çıkartır



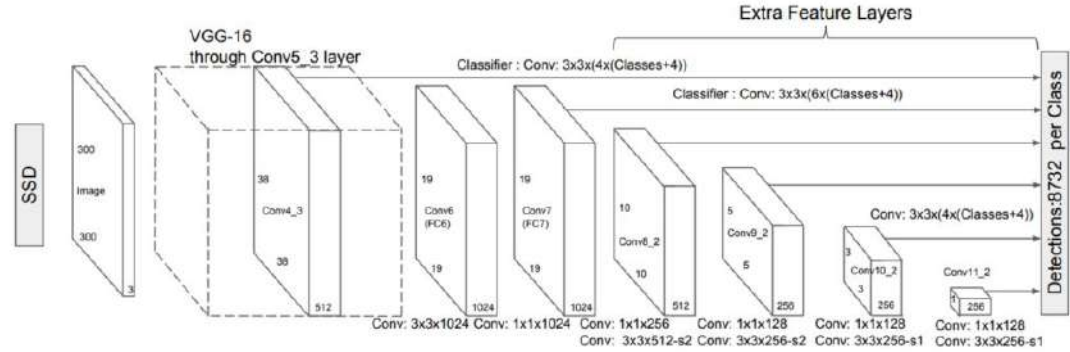
YOLO (You Only Look Once)

- Açılımı “Sadece Bir Kez Bak” olan YOLO algoritması; nesne tespiti tek seferde yapabilen, hızlı bir algoritmadır
- YOLO algoritması, görüntüyü gridler aracılığıyla bakarak tarama yapar
- Nesneleri ve nesne koordinatlarını aynı anda ve tek seferde belirleyebilir



SSD - Single Shot Detector

- Özellik haritası çıkartıp ardından nesneleri algılamak için Evrişim filtreleri uygular
- SSD'de yüksek doğruluk ile algılamanın sebebi, farklı en boy oranlarına sahip birden çok kutu kullanılmasıdır
- Sınırlayıcı kutu, konumlarındaki nesne kategorilerini tahmin etmek için giderek azalan bir evrişim filtresi içerir



Nesne Tanıma Datasetleri

COCO

COCO, büyük ölçekli bir nesne algılama, segmentasyon ve görsel veri kümesidir.

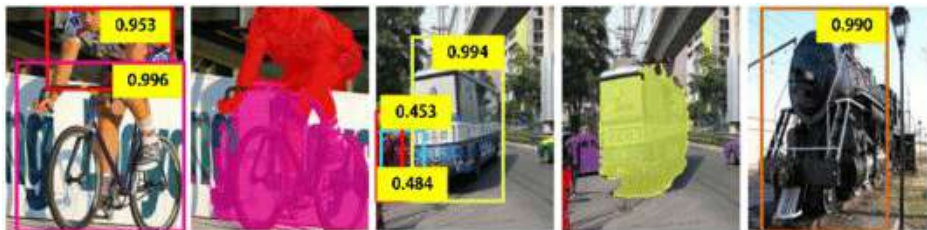
- 330K görsel (>200K etiketlenmiş)
- 1.5 milyon nesne örneği
- 80 nesne kategorisi
- 91 eşya kategorisi
- Verileri tutarken JSON formatı kullanır



```
{  
  "info": info,  
  "licenses": [license],  
  "categories": [category],  
  "images": [image],  
  "annotations": [annotation]  
}
```

Pascal VOC()

- Image processing ve object detection için kullanılan bir veri setidir
- Görüntü dosya ismi, label ve bounding box koordinat bilgilerini barındırır
- 2 GB ana görüntü, 1,9 GB metadata boyutunda gelir
- Veriler işlemler sırasında XML formatında tutulur

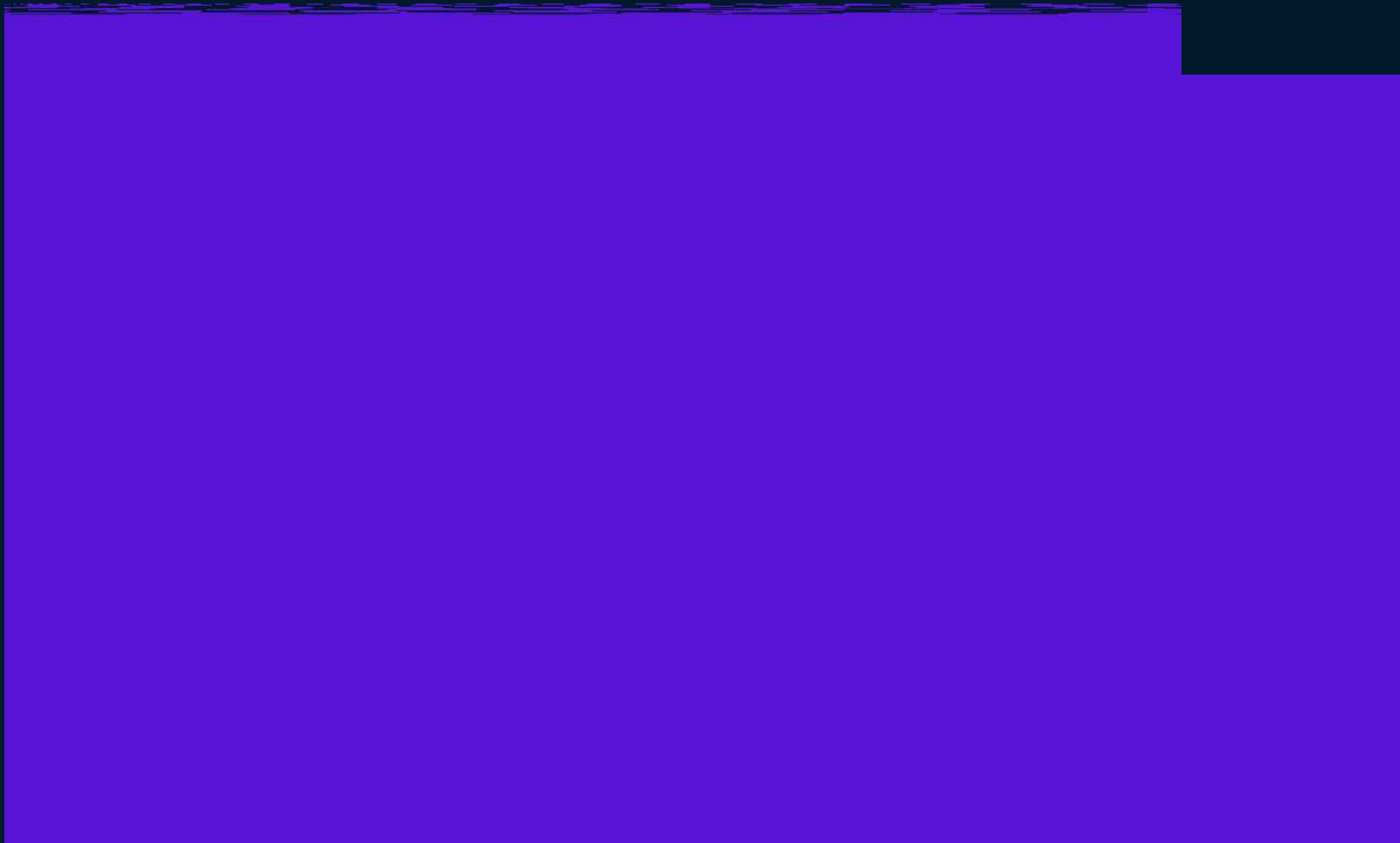


```
<annotation>
  <folder>Kangaroo</folder>
  <filename>00001.jpg</filename>
  <path>./Kangaroo/stock-12.jpg</path>
  <source>
    <database>Kangaroo</database>
  </source>
  <size>
    <width>450</width>
    <height>319</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>kangaroo</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>233</xmin>
      <ymin>89</ymin>
      <xmax>386</xmax>
      <ymax>262</ymax>
    </bndbox>
  </object>
</annotation>
```

Object Detection'i Anlamak için 12 Makale

1. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (2013)
2. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation
3. Fast R-CNN (2015)
4. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
5. You Only Look Once: Unified, Real-Time Object Detection
6. SSD: Single Shot MultiBox Detector
7. Feature Pyramid Networks for Object Detection
8. YOLO9000: Better, Faster, Stronger
9. Focal Loss for Dense Object Detection (RetinaNet)
10. YOLOv3: An Incremental Improvement
11. Objects As Points(CenterNet)
12. EfficientDet: Scalable and Efficient Object Detection

Görüntü Sınıflandırma (Image Classification)



Görüntü Sınıflandırma & Lokalizasyonu

Görüntü sınıflandırma, bir görüntünün göreve göre bir veya daha fazla sınıfa ayrıldığı bir görevdir.

Girdi: bir resim

Çıktı: bir veya birden fazla sınıf(lar)

Görüntü/Nesne lokalizasyonu; çıktının sınırlayıcı kutular çizerek ilgilenilen nesne etrafında x,y koordinatları ile işlem yaptığı bir regresyon problemidir.



MNIST

- 28x28 piksel
- 60.000 Eğitim Veri Eğitim
- 10.000 Test Veri Seti
- El ile yazılmış, 0-9 tüm rakamlar için 70.000 görüntüden oluşan public bir datasettir

label = 3



label = 5



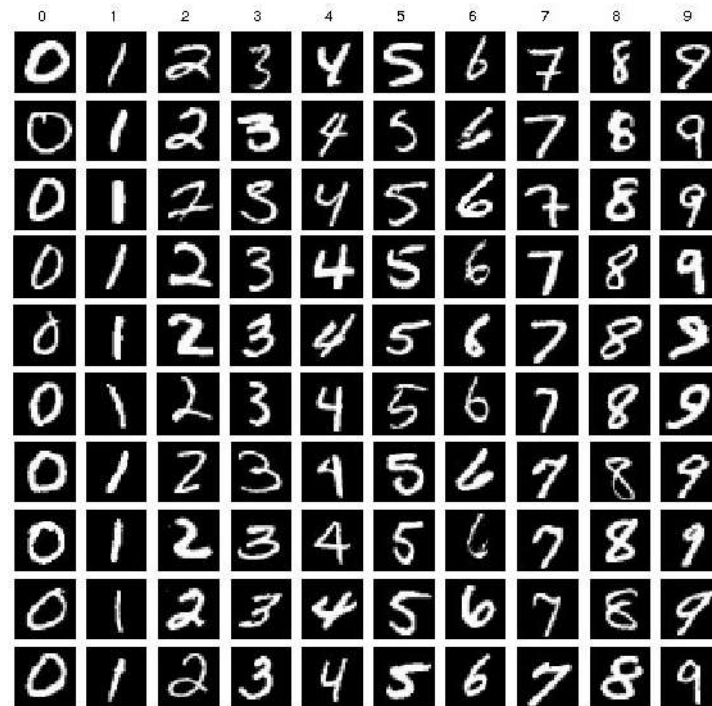
label = 3



label = 6

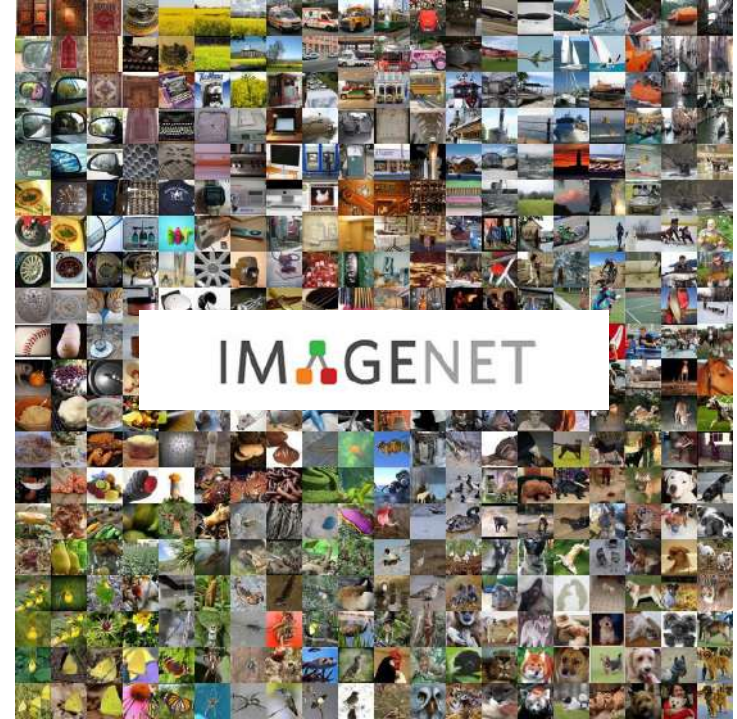


label = 1



ImageNet

- ImageNet Dataseti 14 milyondan fazla görüntü içerir
- Yaklaşık boyutu 150 GB'dir
- Görüntülerin her biri birden fazla sınırlayıcı kutuya ve ilgili sınıf etiketine sahiptir
- Ücretsiz ve public şekilde sunulan bir veri setidir



CIFAR-10

- 32x32 çözünürlüğünde
- 50.000 eğitim görüntüsü
- 10.000 test görüntüsü
- 10 farklı sınıf ve her sınıf için 6000 görüntü

Sağda, 10 sınıf için random şekilde dağılmış görüntü kümesinin bir kısmı yer almaktadır.

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



Colab Zamanı!