

# **CNN ( EVRİŞİMLİ SİNİR AĞLARI )**

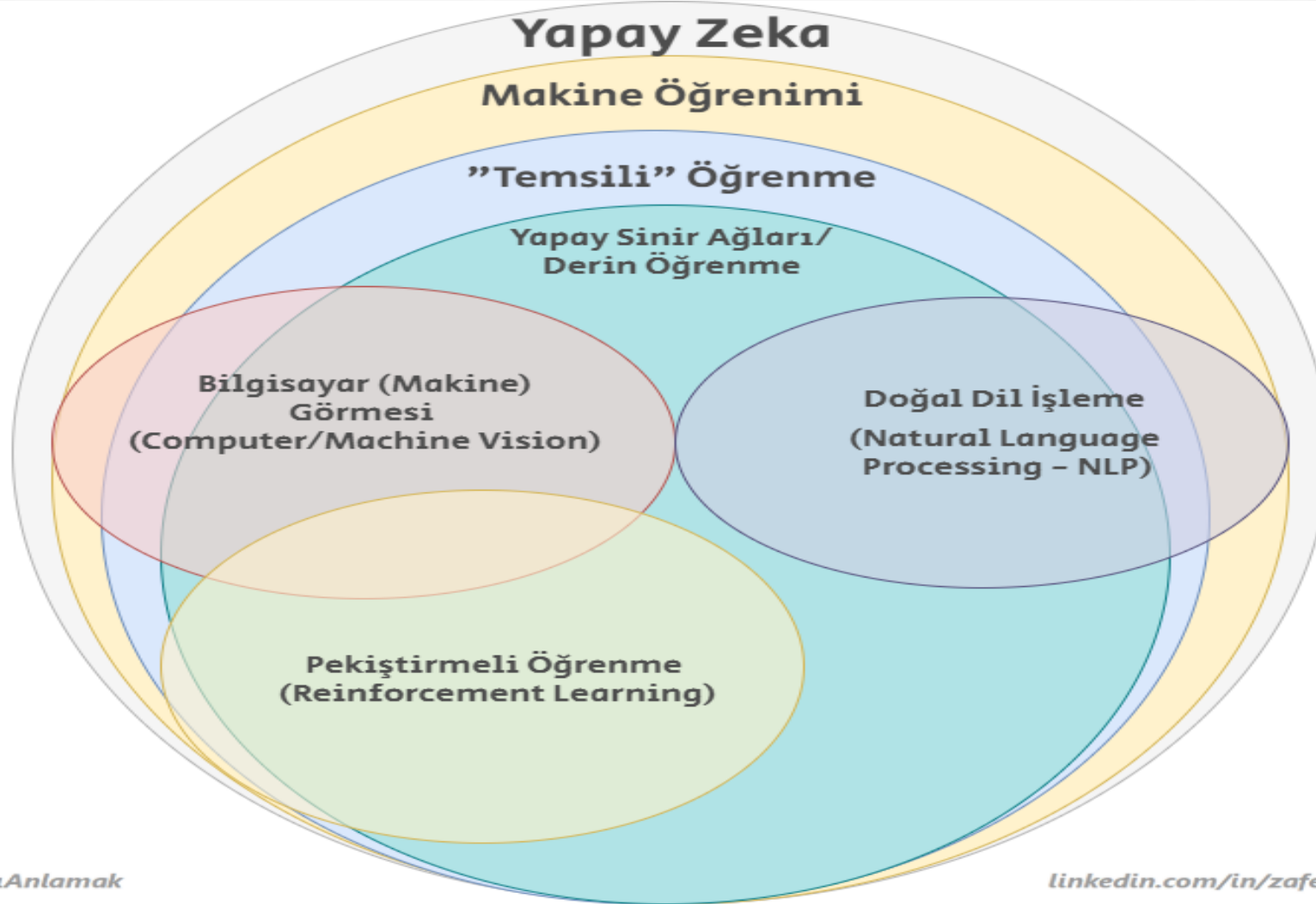


# BİLGİSAYARLAR NASIL GÖRÜR?

- ★ Bilgisayarın görebilmesi konusu ait çalışmalar 1960'lara dayanır.
- ★ İlk olarak görüntülerden anlam çıkaran programlar oluşturulmuştur. Zaman içinde derin öğrenme ile nesneler tanımlanabilmiştir.
- ★ İlk başlarda şekiller ,çizgiler ve kenarları algılayan algoritmalar geliştirilirdi . Daha sonra sinyal işleme , görüntü işleme ve nesne tanıma gibi alanlarda çalışmalar başlamıştır.
- ★ Üzerinde en çok yoğunlaşılan konu nesne tanımak olmuştur . İlk başlarda şablon eşleşme ve kural tabanlı algoritmalara odaklanıldı . Çok başarılı olunmadı . Sınırlı görünümleri sınırlı koşullarda algılanabildi . Yüz tanıma gibi..

# DEVAM..

- **2012 YILINDAN SONRA DERİN ÖĞRENME İLE YAPILAN ÇALIŞMALARDA ÇOK HIZLI YOL KAT EDİLDİ VE GÜNÜMÜZ BİLGİSAYAR GÖRMESİ BAŞARILARINA ULAŞILDI.**
- **BU İLERLEMELERDEN SONRA GÜZEL SONUÇLAR ALMAYA VE İNSANLARIN İLGİSİNİ YAPAY ZEKA SİNİR AĞLARINA ÇEKTİ.**
- **BİLGİSAYAR GÖRMESİ İLE ESKİDEN ZOR OLAN İŞLERİMİZİ ÇOK DAHA KOLAY HALE GETİRDİ.**



**Günümüzde bilgisayar görmesi , derin öğrenmenin bir kolu olarak değişik alt alanlarda farklı uygulamalar olarak karşımıza çıkar .**

- **1.Görüntü sınıflandırma**
- **2.Nesne algılama**
- **3.Görüntü tanıma**
- **4.Örnek bölümlleme**
- **5.Poz algılama**



# Bu alanlarda kendi içinde alt alanlara ayrılır.

Görüntü Tanıma (Image Recognition)



Özel Görüntü Tanıma  
(Custom Image Recognition)



Nesne Algılama (Object Detection)



Anlamsal Bölümleme  
(Semantic Segmentation)



Poz Tahmini (Pose Estimation)



Örnek Bölümleme  
(Instance Segmentation)

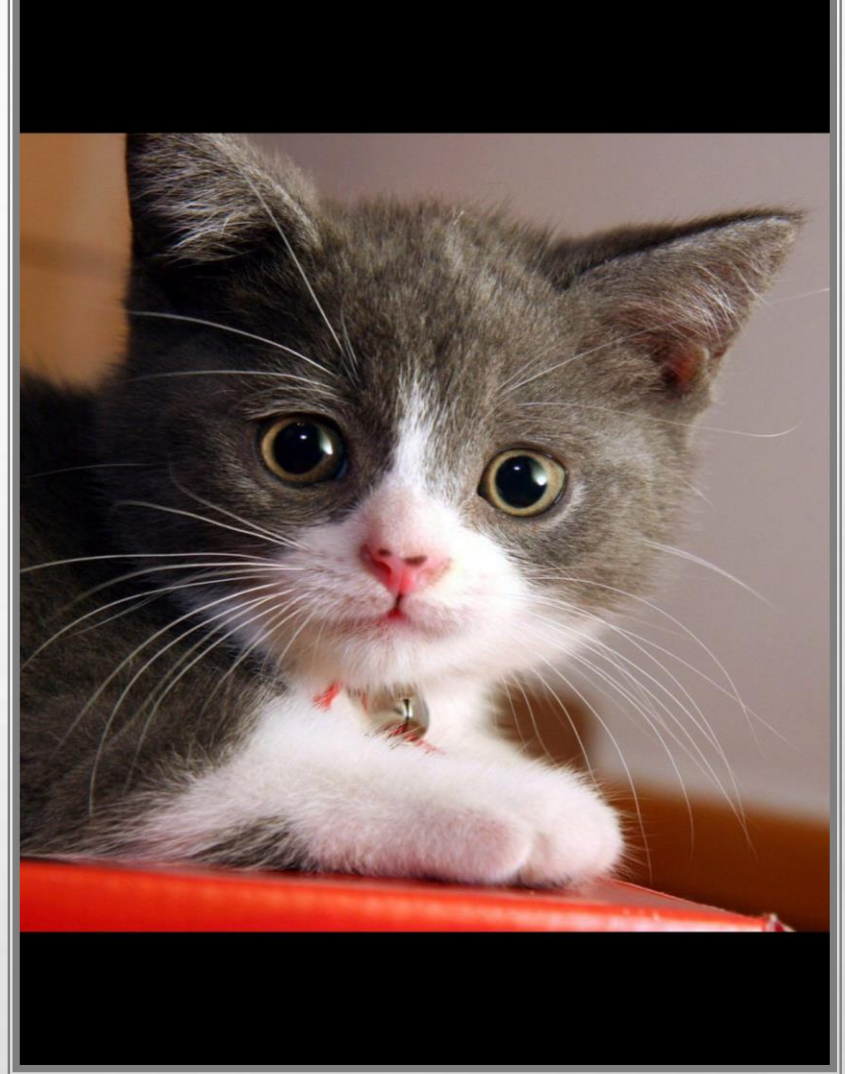


# **BİLGİSAYAR GÖRMESİNDEKİ ZORLUK NEDİR?**

- **En büyük zorluk görüntülerdeki dinamik yapıdır . Kural tabanlı sistemlerde paradigma buraya başarılı bir şekilde uygulanamaz**
- **Bunun anlamı şudur bir bilgisayar programı model olmadan bir görüntüyü tanıyamaz .Model olmadan tanıyabilmesi için o görüntünün aynısının veri tabanında olması ve piksel piksel karşılaştırarak bu görselin aynı görsel olduğunu tespit etmesi gerekir.**

✧ **Örneğin kedinizin bir fotoğrafı olsun . bu fotoğrafın kediye ait olduğunu anlaması için bilgisayarın önünde iki yol vardır.**

✧ **Birinci yol aynı fotoğrafın bir kopyasının veri tabanında olması ve siz fotoğrafı bilgisayara gösterdiğinizde veri tabanındaki tüm fotoğraflarla piksel piksel karşılaştırılıp veri tabanında kopyayı bulması. Pratik bir yol değildir.**

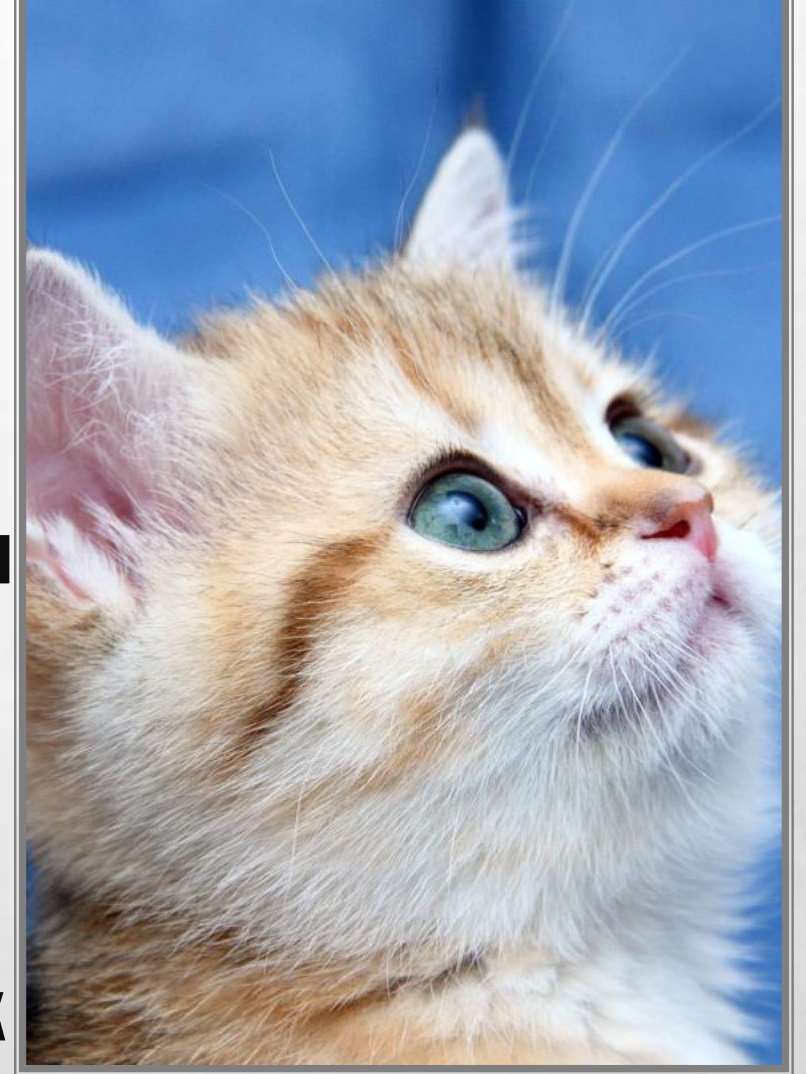




❖ **İkinci yol ise günümüz bilgisayar görmesi teknolojisinin ,görüntülerden ortak özellikler bulmasıdır.**

❖ **Örneğin bir kedi kendine özel karakteristikler taşır. Boyu , kafası ,gövdesi ,gözleri gibi.İşte bu özellikler derin ağda bu karakteristiklere özel ağırlıklar oluşturur.**

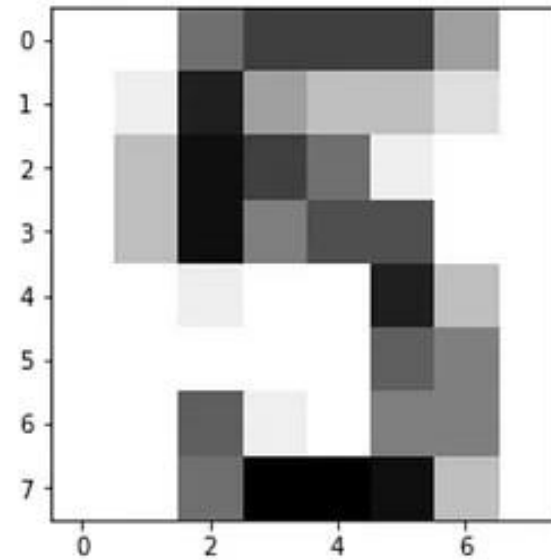
❖ **Yeni bir kedi resminde bu ağırlıklara yakın değerlere sahiptir ve derin ağ onu kedi olarak sınıflandırır.**



# **BAKMAK , GÖRMEK : BİLGİSAYARLAR BİR GÖRSELİ NASIL ALGILAR?**

- **“Bakmakla görmek arasında fark vardır ” şeklinde klasik deyimimiz burada uygulanabilir.**
- **Bilgisayarlar bakabilirler ama göremezler.**
- **Bilgisayarlar analog çalışmazlar, onlar sayılardan sıfır ve birlerden anlarlar.**
- **Bir bilgisayar (makine) için bir resim ,fotoğraf ve bunlardan oluşan videolar pikseller topluluğudur.**

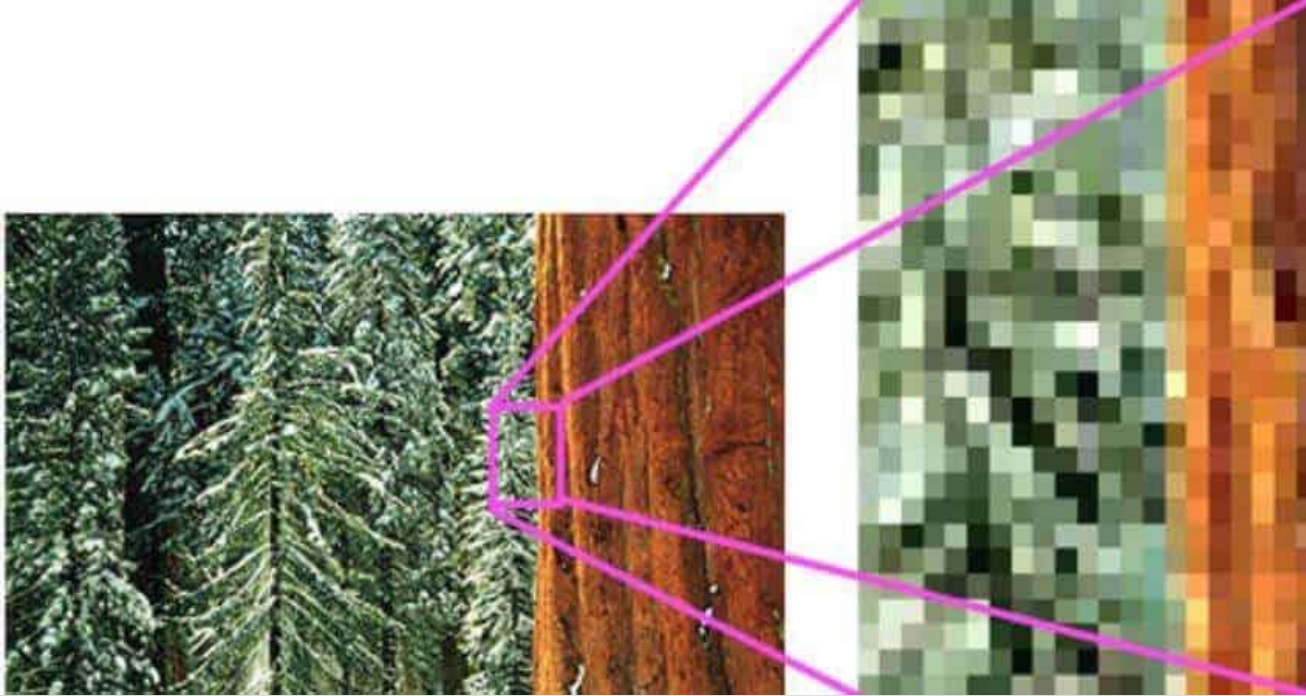
- Burada piksellerden gelen sayısal değerleri alıp hesaplamalar yaparız ve bilgisayarın görmesini sağlarız.
- Örneğin bu 5 rakamının görselinin noktalarında farklı sayısal değerler vardır. Bu rakamlar bir araya gelir ve 5 rakamını oluşturur. Bilgisayarın bir görsele bakış açısıda böyledir.
- Burdaki sayılar topluluğundan bir model oluşturulur. Buda zaten makine öğrenmesi, derin öğrenme algoritmaları ve hesaplamaları yapılan süreçtir.



```
1 digits.images[1010]
array([[ 0.,  0.,  9., 12., 12., 12.,  6.,
        [ 0.,  1., 14.,  6.,  4.,  4.,  2.,
        [ 0.,  4., 15., 12.,  9.,  1.,  0.,
        [ 0.,  4., 15.,  8., 11., 11.,  0.,
        [ 0.,  0.,  1.,  0.,  0., 14.,  4.,
        [ 0.,  0.,  0.,  0.,  0., 10.,  8.,
        [ 0.,  0., 10.,  1.,  0.,  8.,  8.,
        [ 0.,  0.,  9., 16., 16., 15.,  4.,
```



**Bir resmi zoom yapınca daha net bir şekilde pikselleri görebiliriz  
piksellerin sayısal deęerleri belli renklere karřılık gelir.  
RGB matrisi renkli gorselleri oluřturmak için kullanılır. Bu üç rengın  
karıřımı dięer renkleri oluřturur.**





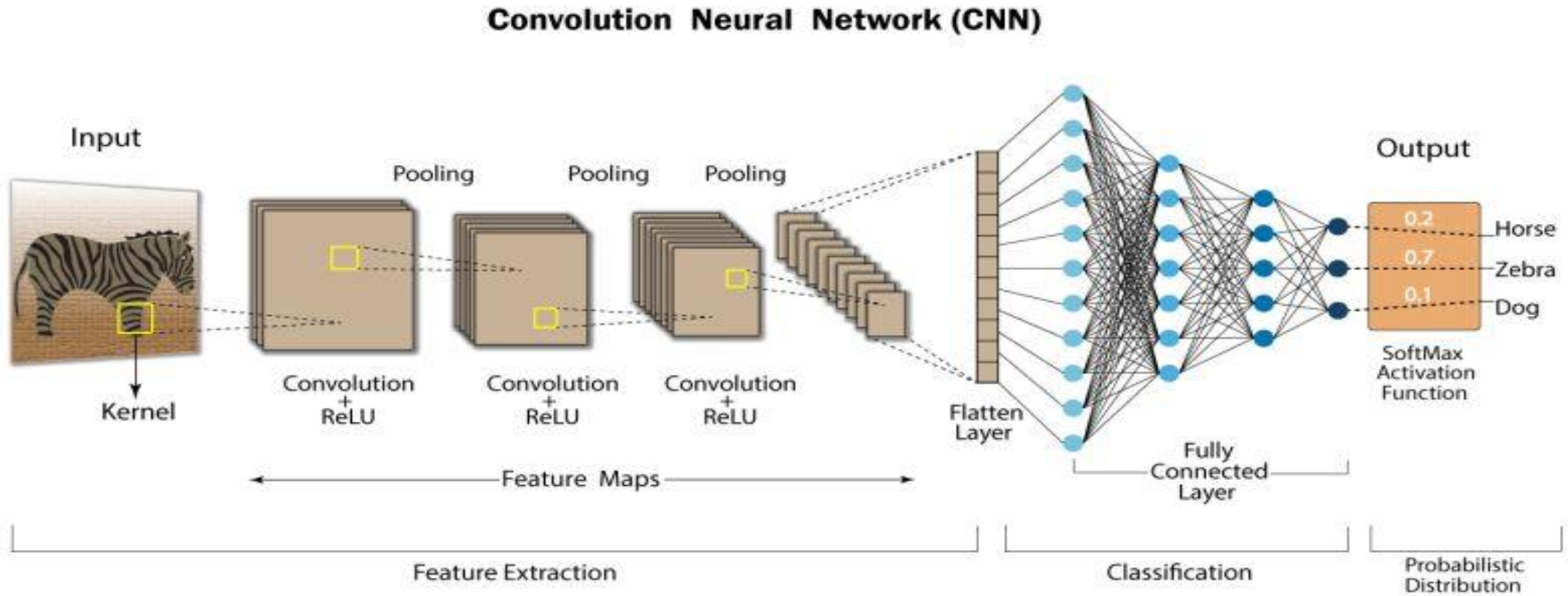
# **BİLGİSAYAR GÖRMESİNİN ALTYAPISI: CNN (EVİRİŞİMLİ SİNİR AĞLARI)**

**Bilgisayar görmesi için kullanılan ağ mimarisidir.**

**Bir makinenin görebilmesi için , yeni nesneleri , canlıları algılayıp tanıyabilmesi için yapay zeka tarihi boyunca değişik yöntemler , algoritmalar , teknolojiler geliştirilmiştir . Hiçbiri cnn kadar başarılı olamamıştır.**

**Mimarinin alt yapısı yapay sinir ağına dayanır , yalnız tam bağlantılı yapay sinir ağına gelmeden önce görsel veriler çok özel metot ve algoritmalarla oluşan yine bir sinir ağ mimarisine geçerler. Bu işlemlerde evrişimli sinir ağ mimarisini oluşturur.**

# EVRIŞİMSEL SİNİR AĞI



# MİMARİSİ

- **Temel olarak iki yapıdan oluşur.İlk bölüm özellik öğrenme ,ikinci bölüm de sınıflandırma bölümüdür**
- **Sınıflandırma bölümü tam bağlantılı sinir ağlarındaki işleyiştir.**
- **Özellik öğrenme ise tam bağlantılı ağa gelene kadar görsel hakkındaki karakteristik özelliklerin ortaya çıkarılmasıdır.Buna özellik haritasıda denir.**

# DEVAM..

- **Görsel tam bağlantılı ağa gelince birtakım işlemler yapılarak karakteristik özelliği ortaya çıkarılır ve veri boyutlarında azaltmalar yapılır.**
- **Özellik öğrenme veya çıkarmadaki bu işlemler ise bir takım teknikler ile yapılır.**
- **Bu tekniklerin başında filtreleme gelir. Daha sonra havuzlama işlemi uygulanır**
- **Bu işlemler bir çok tekrar ile yapılır sonunda tam bağlantılı ağa daha karakteristik , damıtılmış veriler gelir.**



# EVRIŞİM KATMANI

- **Evrişim katmanı CNN algoritmalarında görüntüyü ele alan ilk katmandır. Bilindiği üzere görseller aslında içlerinde belirli değerler taşıyan piksellerden oluşan matrislerdir.**
- **Evrişim katmanı görüntünün kaydırıldığı ,filtrenin girdileri ile girdi görüntüsü arasında bir dizi filtreden oluşur. Bu filtreler eğitilir.**
- **Her görüntü kaydırılması işlemi sonucunda bu görüntü ile filtre arasında evrişim işlemi uygulanır. Her görüntü ayrı bir öznitelik olarak kabul edilir.Tüm kaydırma işlemleri sonucunda oluşan topluluk öznitelik haritası olarak adlandırılır.**

# FİLTRELER

- **Öncelikle kedi görselini tekrardan hatırlayalım. Bilgisayarın bir kedi görselini tanıyabilmesi için daha önceden kedi görseliyle eğitilmiş bir modeli kullanması gerekiyordu.**
- **Bu modelin oluşabilmesi için de binlerce , milyonlarca kedi görseli ile yapılan model eğitimi gerçekleştirilmesi gerekiyor.**
- **Bu eğitim esnasında kedilerin karakteristik , ortak özelliklerin tespit edilip buna göre sinir ağı ağırlıkları oluşturulup ,modellenmesi lazım.**
- **İşte bu karakteristiklerin ,özelliklerin tespit edilmesinde filtreler kullanılır.**

# DEVAM..

- **FİLTRELER BAZEN “KERNEL”(ÇEKİRDEK ) OLARAK ADLANDIRILIR.**
- **FİLTRELEME MANTIĞINDA GÖRSELİ BELİRLİ ÖZELLİKLERE GÖRE FİLTRELEMEK KALIPLARA AYIRMAK,AYRIŞTIRMAK YATAR.**
- **GÖRSELİMİZ HEPSİNİN FARKLI GÖRSEL KARAKTERİSTİKLERİ OLAN BU FİLTRELERE DAYALI BÖLÜMLERE**

**Burada ki evin bilgisayar tarafından algılanabilmesi matematiksel işlemlerle olur.Bu da sayılarla çalışmak demektir.Burada sadece sıfır ve bir değerini almıştır.**

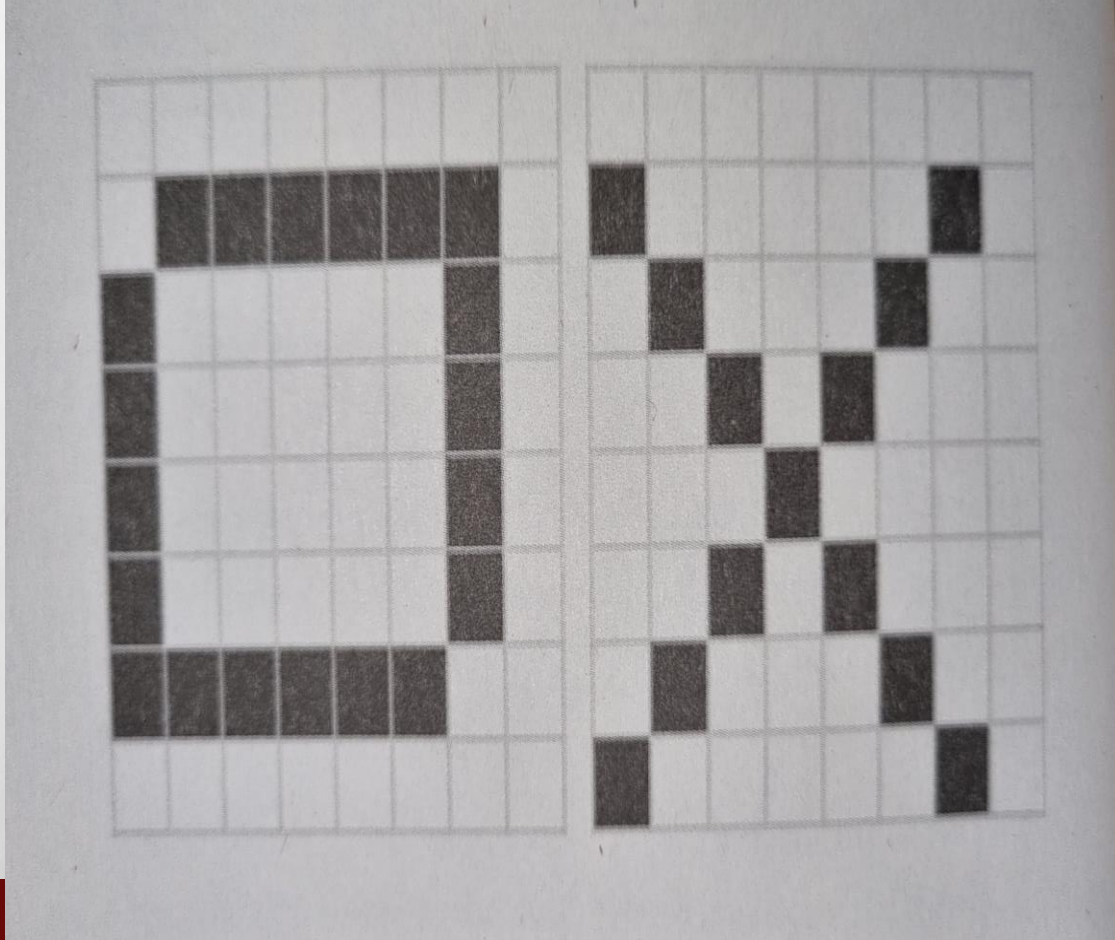
**Bunun için bunlara ait sayısal değerleri tespit edip karakteristik özelliklerini yine sayısal olarak ortaya koyup medellememiz gerekir.**

**Bir filtre belirli bir boyuttaki görseldir.**

0	0	0	1	0	1	0
0	0	1	0	1	1	0
0	1	0	0	0	1	0
1	0	0	0	0	0	1
1	0	1	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1
1	0	0	0	0	0	1
1	1	1	1	1	1	1



**8x8 piksel 2 görüntü vardır.**



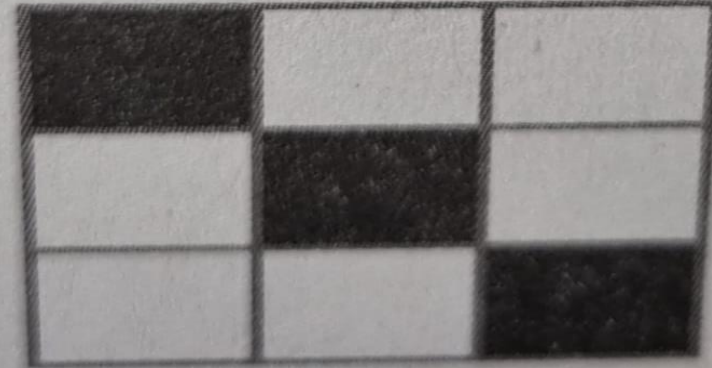
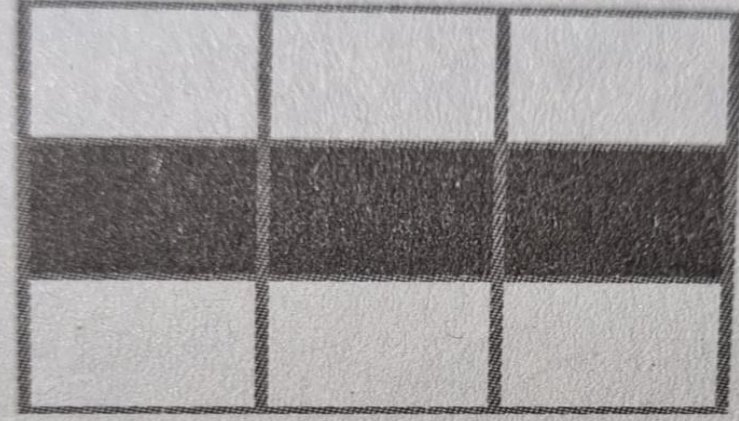
**Görüntüleri sayısal matrislerle gösterimi**

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	1	0
1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0
0	1	0	0	0	1	0	0
0	0	1	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	0

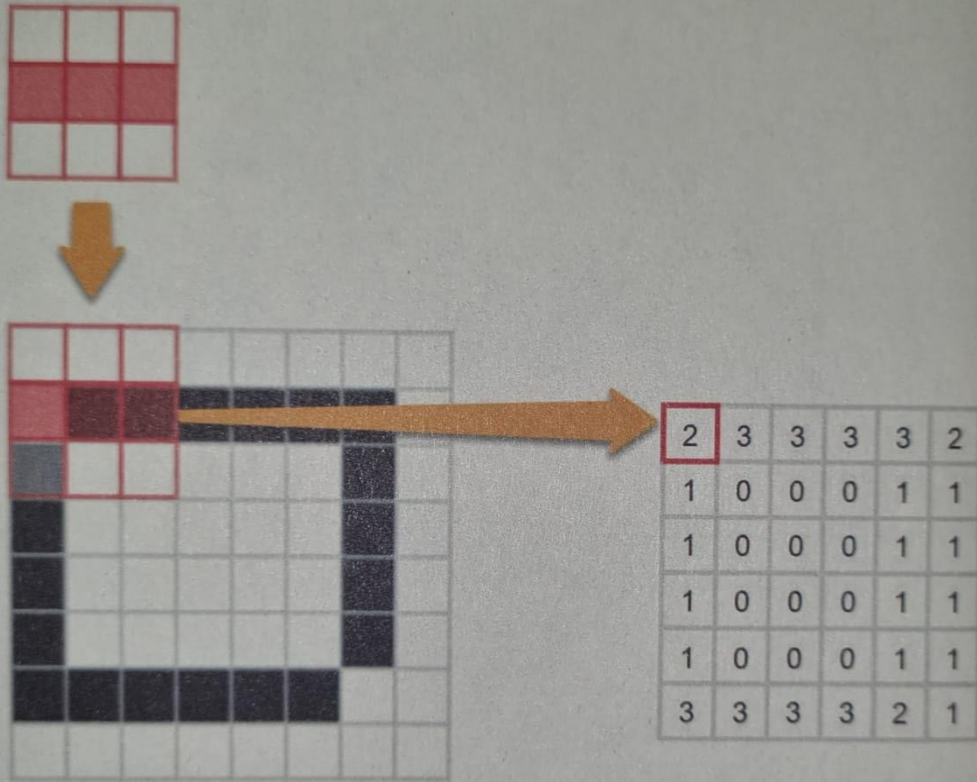
✦ **3x3 pikselde bir filtre örneđi**

✦ **Bir filtrenin girdi görseline uygulanması adım adım olur.**

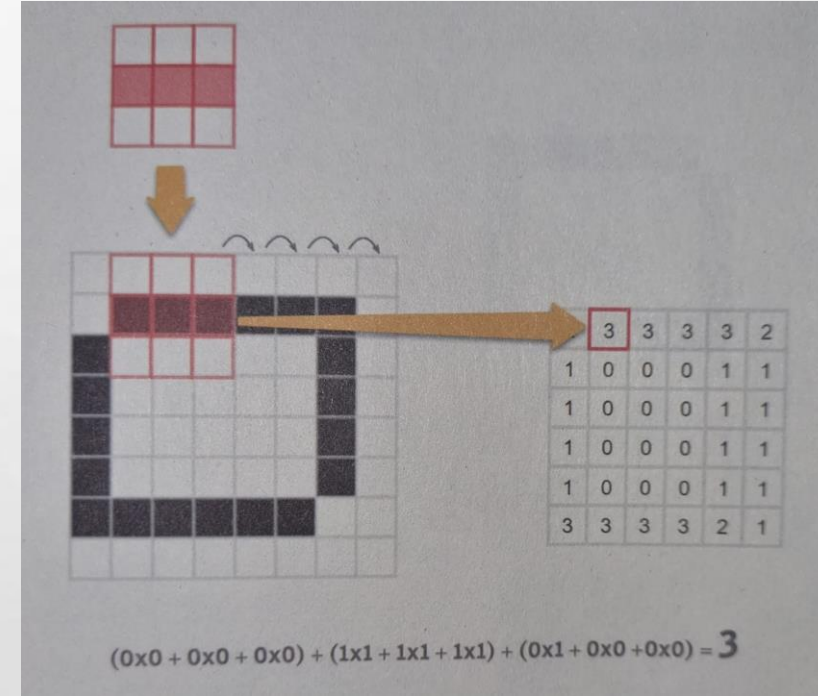




**Ana görselin sol üst köşesinden başlayarak yan tarafa doğru adım adım kaydırılır ve her bir pozisyon için ana görsel ile filtre değerlerinin çarpımın toplamı hesaplanır. Bu işlem filtre kapsamındaki her satır ve sütun örtüşmesi için yapılıyor ve sonunda toplam bir değer çıkıyor. Bu filtrelenmiş matrisnin ilk satır ve sütun değeri oluyor**

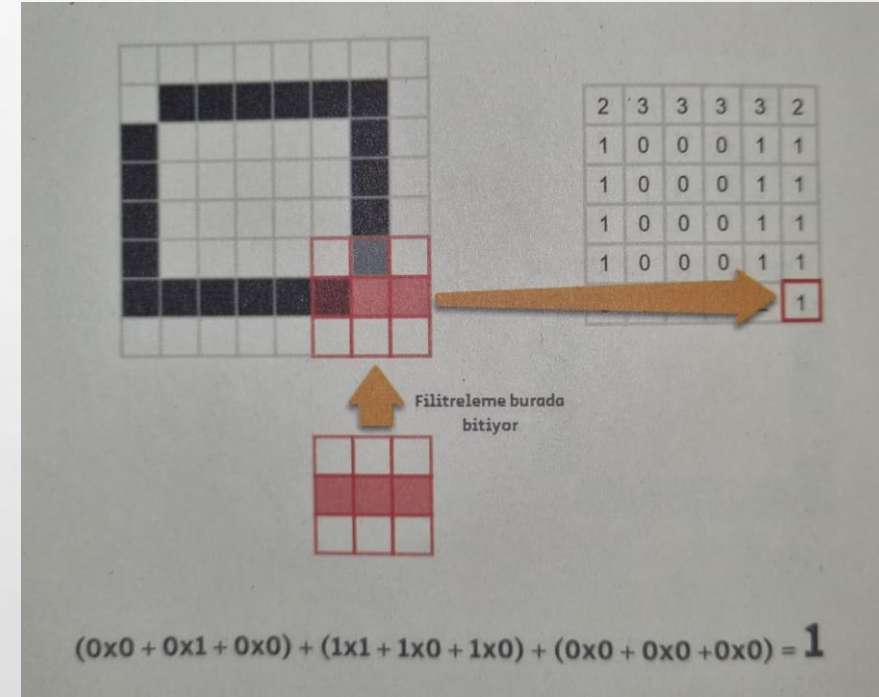
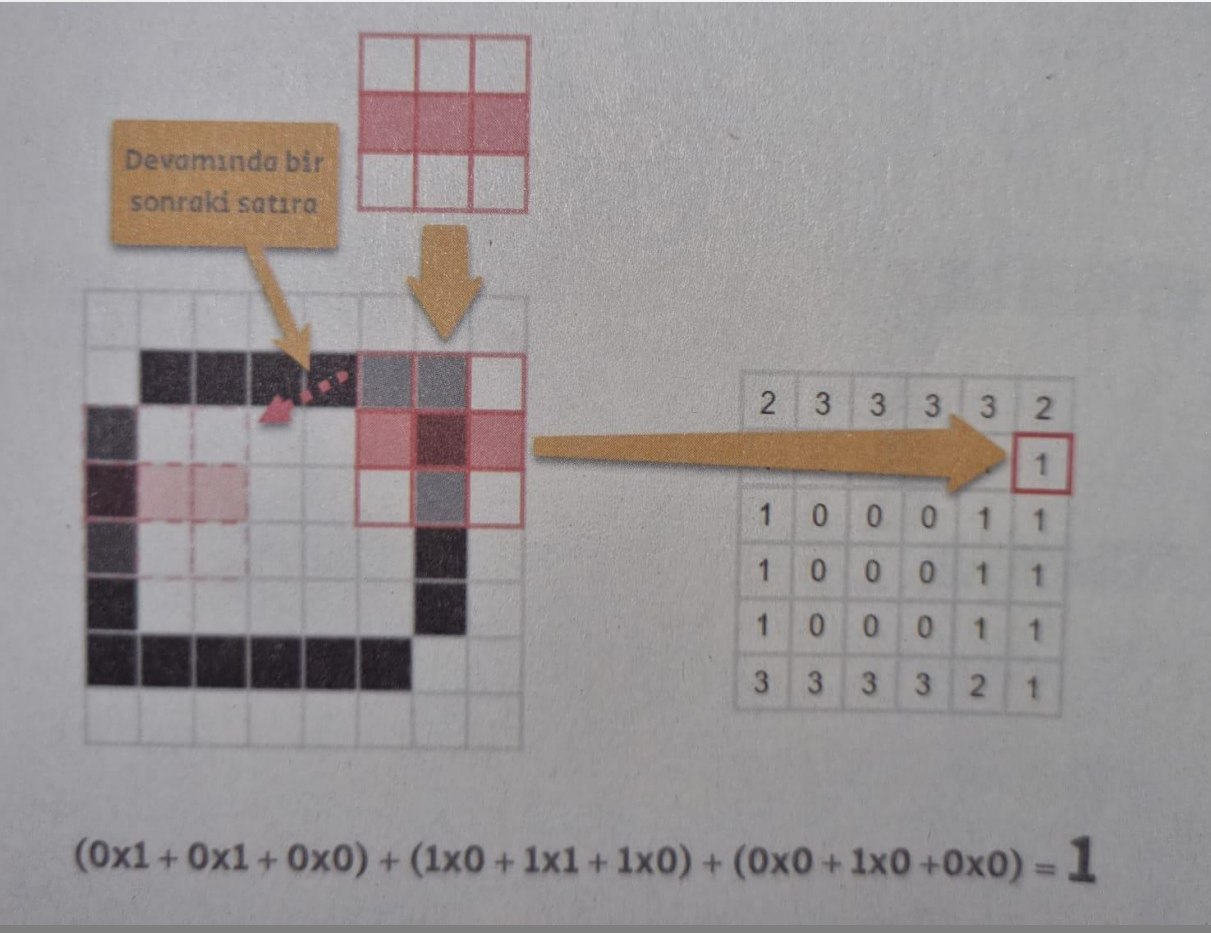


$$(0 \times 0 + 0 \times 0 + 0 \times 0) + (1 \times 0 + 1 \times 1 + 1 \times 1) + (0 \times 1 + 0 \times 0 + 0 \times 0) = 2$$



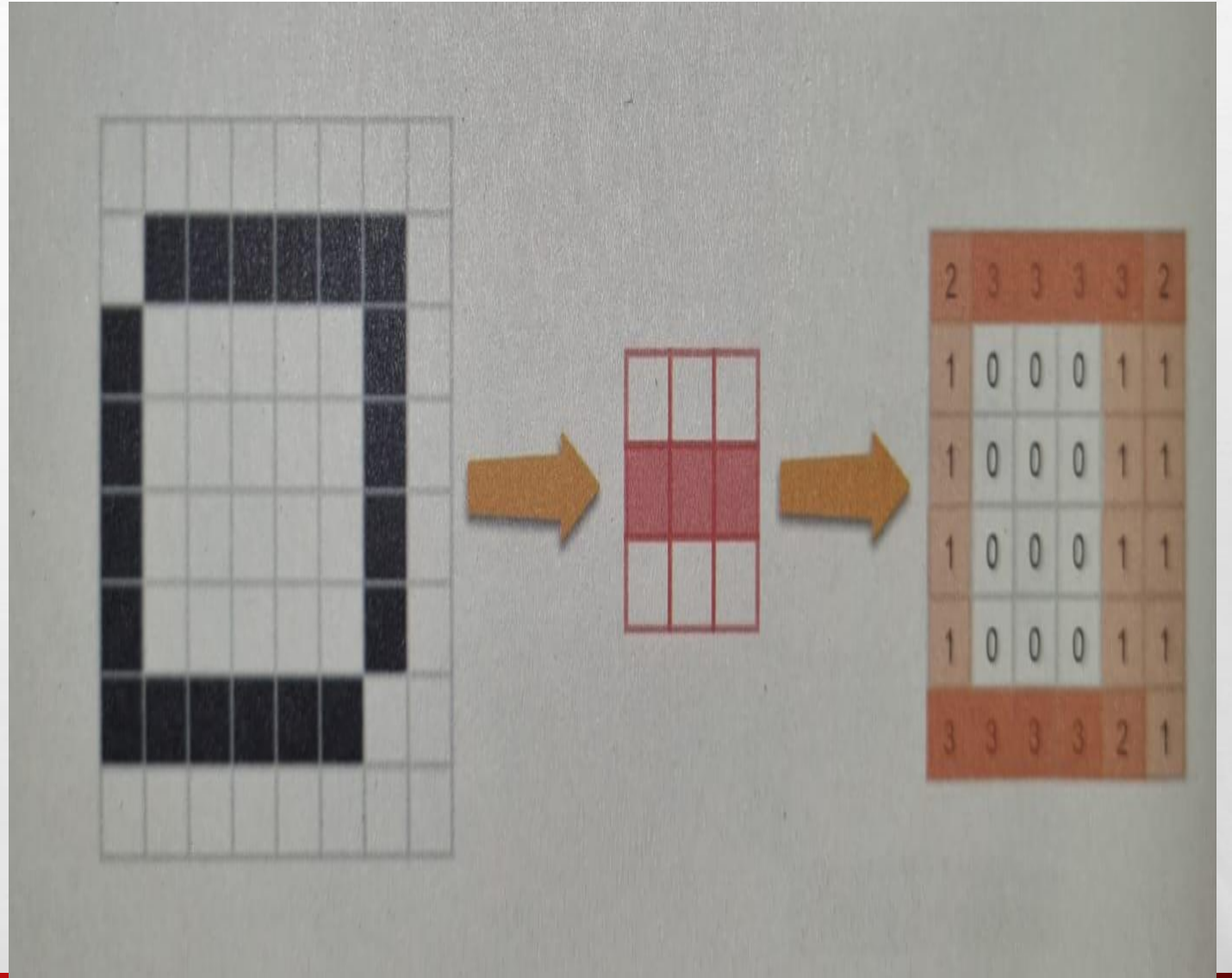
$$(0 \times 0 + 0 \times 0 + 0 \times 0) + (1 \times 1 + 1 \times 1 + 1 \times 1) + (0 \times 1 + 0 \times 0 + 0 \times 0) = 3$$

**Sütun kaydırması filtremizin son sütunu ile ana görselimizin son sütunu örtüşünce bitiyor ve bir alt satıra geçiyor ve bir alt satıra geçilip aynı işlem tekrar yapılıyor. Bu işlem ana matristeki bütün satır ve sütunlar içim tekrar edilir.**

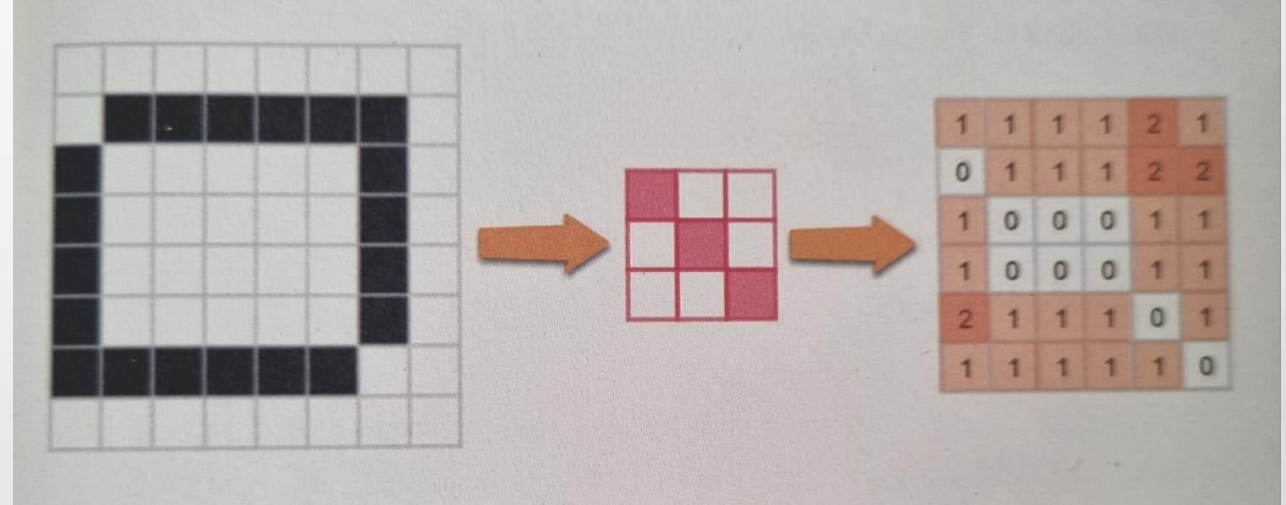




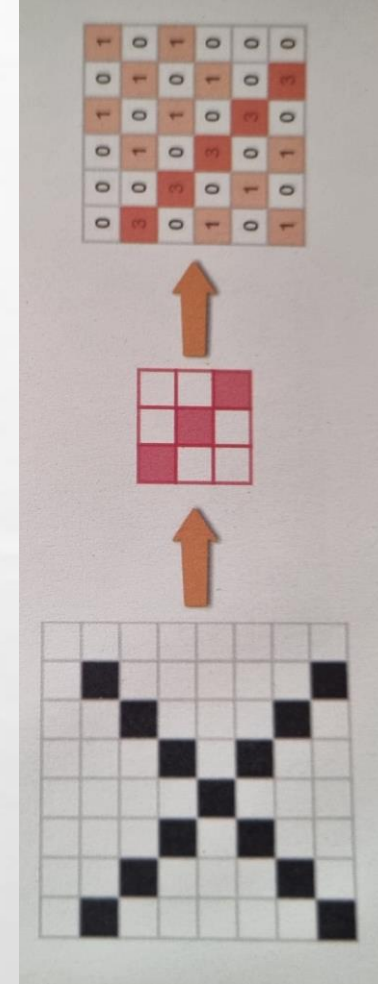
- **Sonuç olarak 8x8 bir resme 3x3 filtrelemesi yapılırsa yeni bir resim ortaya çıkar.**
- **Yeni görseli ağırlıklara göre renklendirilmiştir.**
- **Yatay alanların ağırlığı fazla çıktığına göre yatay alanları ön plana çıkarmıştır.**



**0 harfine apraz filtre  
uygularsak ıktısında  
apraz zellikler ağırlık  
oluşmamıştır.Filtreye ait  
zellikler tespit  
edilememiştir**

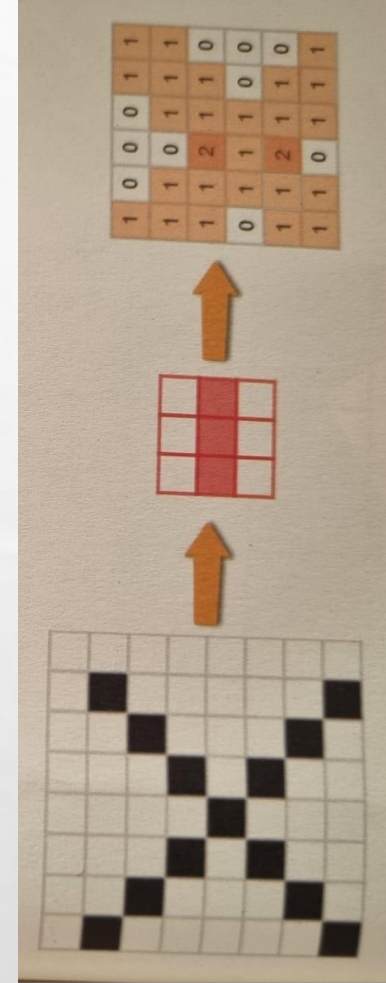


**Aynı apraz filtreyi X  
harfine uygularsak kendi  
eđim ynnde alanların  
ađırlıklarını en fazla  
ıkarmıřtır.  
apraz zelliđi algılamıřtır.**



**Tam tersi durumdaysa bu x  
görselimize yatay filtreyi  
uygularsak böyle bir özellik  
seçilmediğini görürüz.**








**Bir CNN ağında bir çok filtre  
kullanılır. Her bir filtre  
görseldeki belli alanları ön  
plana çıkarır, o özellikleri  
yakalar.**

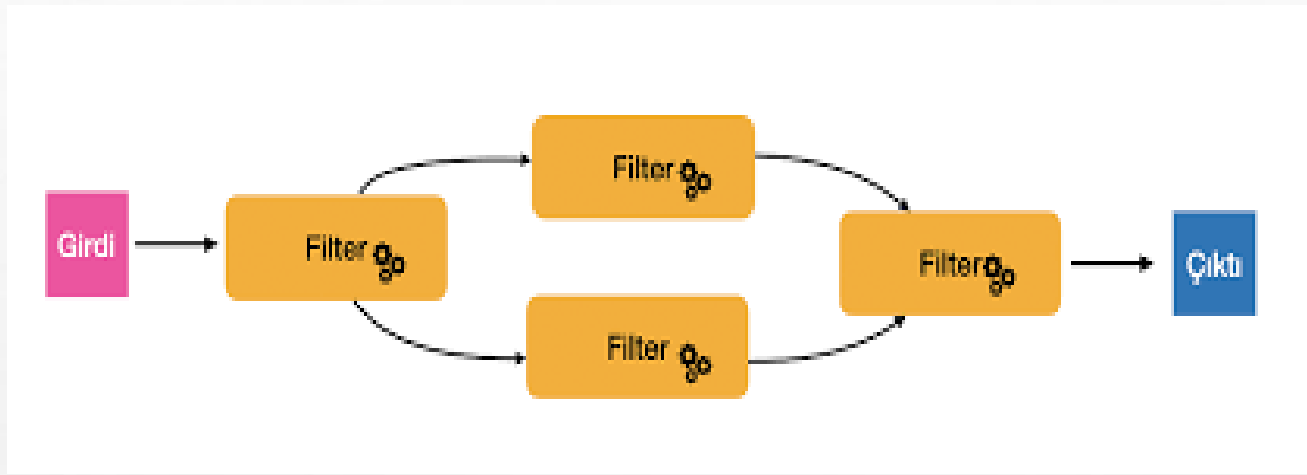




**Farklı filtre tiplerine  
örnek..**

**Ağ ne kadar derin olursa  
filtreler o kadar karmaşık  
olur.**

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	



**Evrişimli sinir ağında , giriş görseline bir çok filtre uygulanır.**

**Daha sonra bunlara havuzlama denilen işlem uygulanır ve belirli bir aktivasyon fonksiyonundan sonra bir sonraki katmana geçilir.**

**Bir sonraki katmanlarda benzer yapıya sahiptir, burada yine filrelemeler yapılır ve ağ mimarisi bu şekilde devam eder.**

# HAVUZLAMA (POOLING)

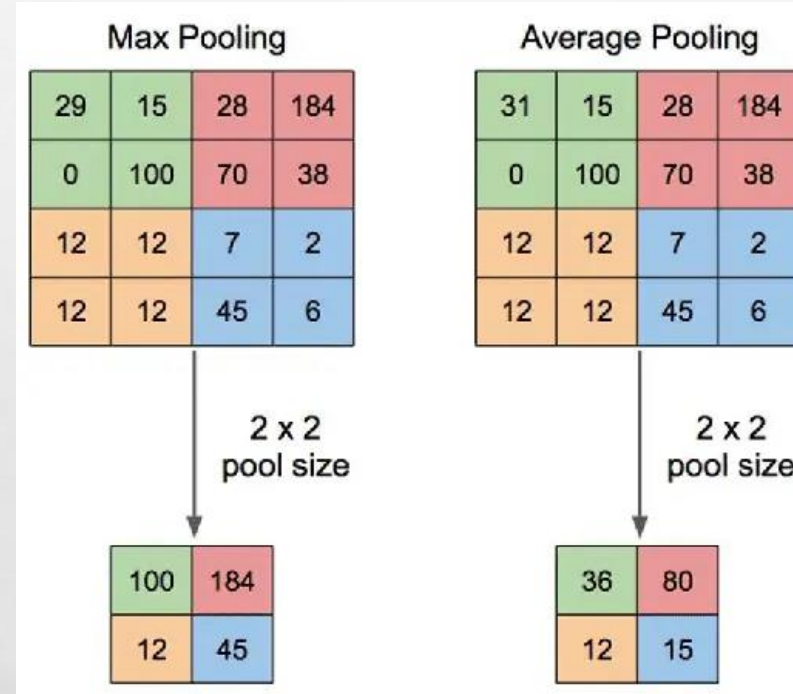
- **Filtreleme işleminden sonra uygulanan işlem havuzlamadır.**
- **Havuzlamanın ana fikri görselleri oluşturan ve filtre uygulanmış matrisleri, daha küçük boyutlara indirgemektir. Böylece remin ana karakteristiği kaybolmadan bir sonraki katmana iletilmesi sağlanır ve boyutuda küçülmüş olur.**

# DEVAM..

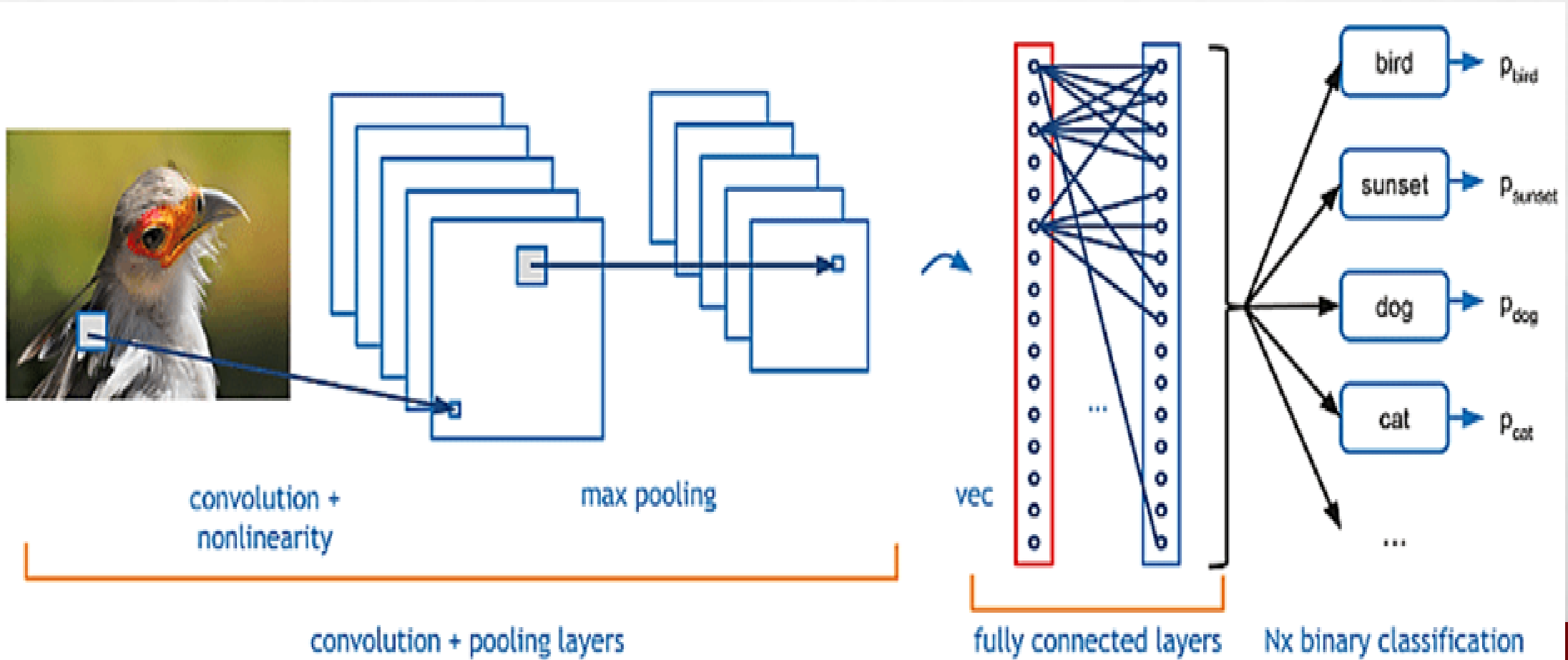
- **Belirli kriterlere göre yapılan havuzlamada giriş değerini temsil eden ,daha az veri ile yoğunlaştırılmış çıktılar elde edilir. Mantık giriş karakteristiğinin kaybolmadığı varsayımına dayanır. Böylece daha az veri ile daha a hesaplatmalar veya işlemler yapılarak ağın daha verimli çalışması sağlanır.**
- **CNN modellerinde genellikle kullanılan iki farklı pooling tekniği vardır. Bunlardan biri max (maksimum) diğeri de average (ortalama) pooling'tir.**



**Eğer maximum havuzlama uyguluyorsanız filtrenin kapsadığı alandaki en büyük değeri, ortalama havuzlama uyguluyorsanız ise filtredeki değerlerin ortalamasını alır. Bu sayede boyut azalır ve önemli özellikler elimizde kalır.**



**Bir görselin girdisi evrilme katmanı ve havuzlama katmanından sonra tam bağlantılı katmana geçer.**



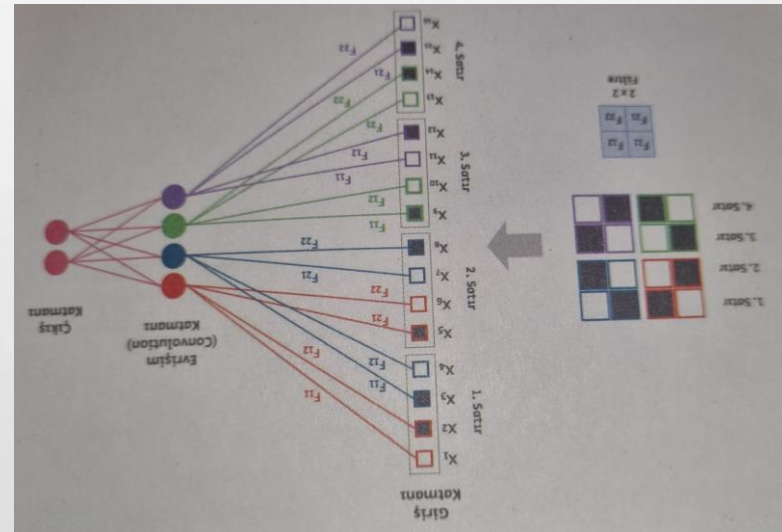
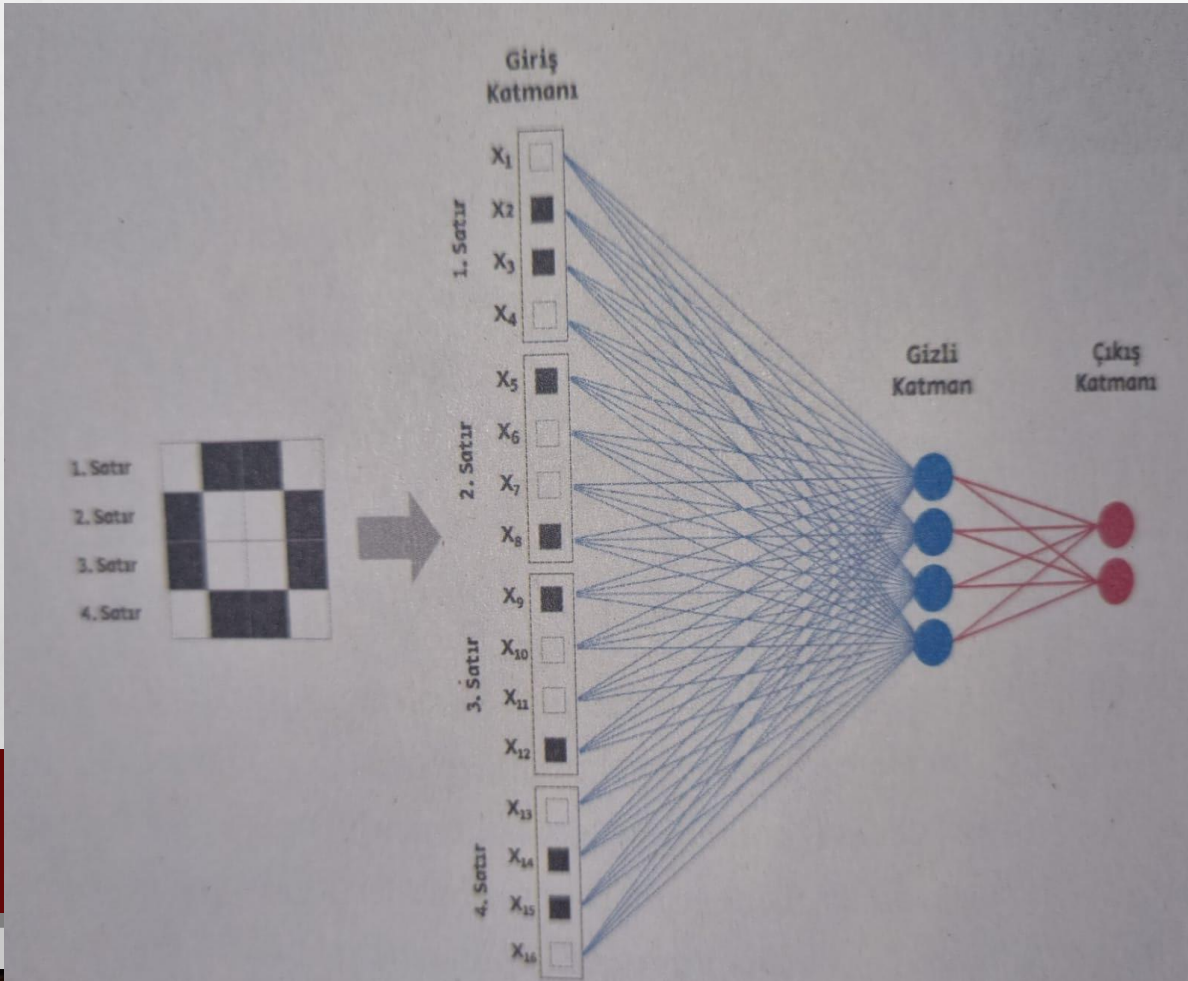
# **TAM BAĞLANTI KATMANI**

- **Bir evrişim sinir ağı , bir önceki slaytta gördüğümüz filtreleme , havuzlama ve aktivasyondan oluşan katmanların bir araya gelmesinden oluşur.**
- **Bu mimari filtreleme sayısı ,yöntemi ,havuzlama teknikleri vs. Gibi bölümlerde farklı yapılarda Ve sayılarda olabilir ama genel mantık değişmez.**
- **Bütün bu işlemlerden geçildikten sonra ağın nihai katmanı tam bağlantılı yapay sinir ağına gelinir .**
- **Öncesinde epey damıtılmış yoğunlaştırılmış özellikleri karakterize edilmiş veriler , son aşamada sınıflandırmanın yapılması için bu ağ sokulur.**

# EVRIŞİMLİ SİNİR AĞLARI BAĞLANTI FARKLİLİĞİ

**TAM BAĞLANTILI SİNİR AĞI :** Bağlantı sayısı fazladır

**EVRIŞİMLİ SİNİR AĞI:** Bağlantı sayısı filtrelemeden dolayı azalır





# **TARİHSEL SIRALAMASINDA GÖRE EN BİLİNER CNN AĞLARI**

- **LeNet-5 (1998)**
- **AlexNet-2012**
- **VGG-16-2014**
- **Inception-v1-2014**
- **Inception-v3-2015**
- **ResNet -50-2015**
- **Xception-2016**
- **Inception-v4-2016**
- **Inception-ResNets-V2-2016**
- **ResNext-50-2017**

# CNN TARİHSEL GELİŞİMİ

- **LeNet (1998):**
- **İlk CNN 1998 yılında Yann LeCun tarafından geliştirildi. LeCun Amerikan posta ofisindeki görevinden dolayı özellikle zarfların üzerine el ile yazılan yazılara odaklandı ve LeNet geliştirdi.**
- **2000'li yıllarda bu konuda kayda değer fazla bir gelişme olmadı.**

- **AlexNet(2012):**
- **LeNet ile aynı prensipler üzerine inşa edilmiş olsada AlexNet çok daha derin bir mimaridir. AlexNet' teki eğitilebilir parametrelerin toplam sayısı 60 milyondur, bu sayı LeNet'in 1000 katıdır, bunun haricinde 650.000 nörona sahipti ve 5 evrişim katmanından oluşuyordu.**

- **VGG16(2014)**
- **“Büyük Ölçekli Görüntü Tanıma İçin Çok Derin Evrişimli Ağlar” makalesinde önerilen evrişimli bir sinir ağı modelidir. Büyük filtre boyutlarını kullanan ilk CNN ‘lerden biriydi.**
- **Dezavantajı , eğitilmesi gereken çok daha fazla parametrenin olması ve önemli bir eğitim süresi gerektirmektedir.**



- **GOOGLNET/INCEPTION(2014)**

- **Bu mimarinin ana özelliği derin bir sinir ağı oluştururken ,ağ kaynaklarını daha verimli kullanmaktı."Inception" parametre sayısını 12 kat azaltırken (~138 milyondan~13 milyona) 22 katman derinliğinde bir ağ kullanır.**
- **Ağ klasik ağlardan esinlenen bir CNN kullandı , ancak başlangıç modülü olarak adlandırılan yeni bir öğe uyguladı**

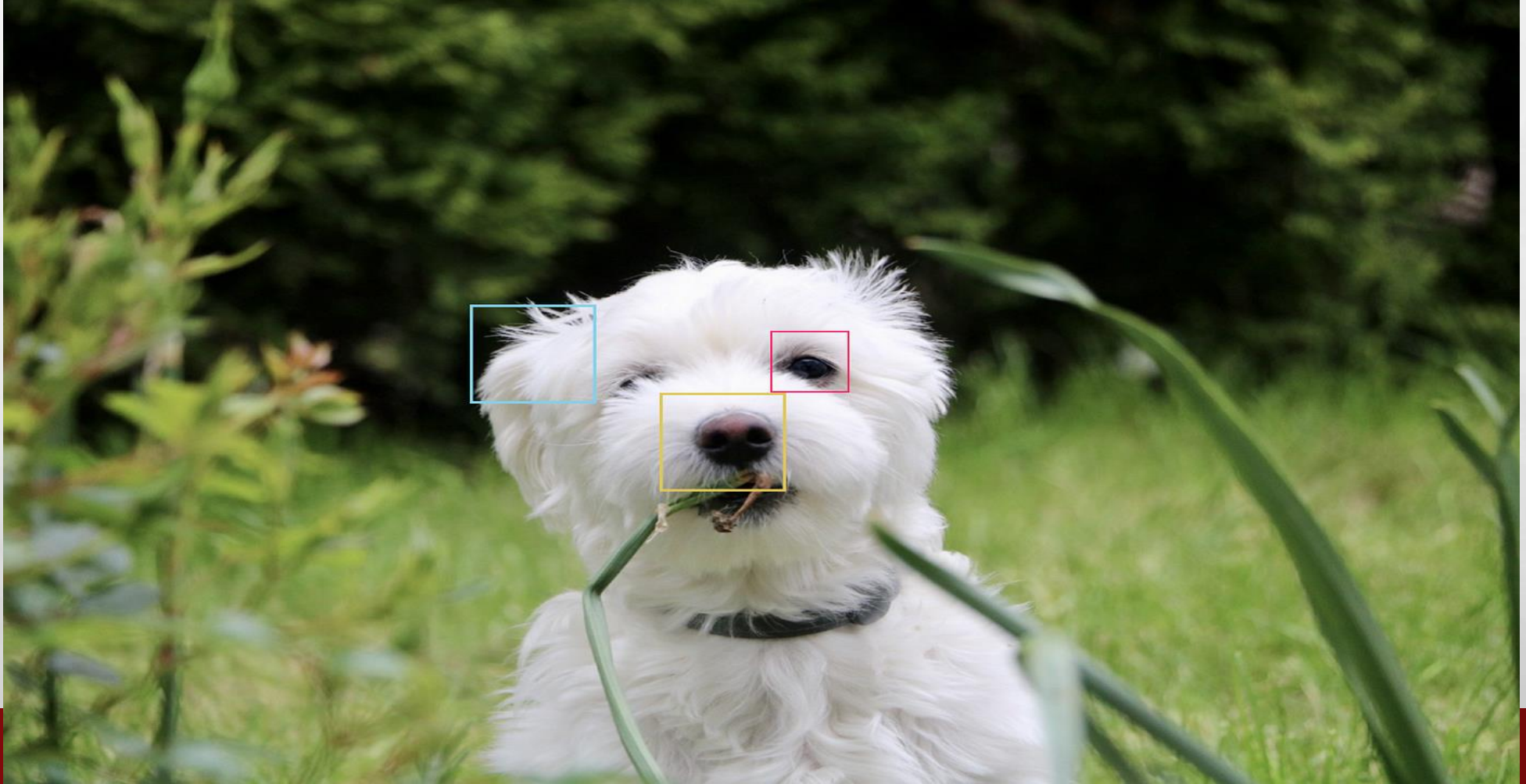
- **RESNET(2015)**

- **Microsoft araştırma ekibinden bir grup tarafından 2015 yılında geliştirilmiştir.Ağın özelliği parametre sayısını çok artırmadan,sinir ağının daha derin olmasını sağlayan,"residual" blok tekniğini uygulamasıdır.Böylece vggnet gibi daha düşük katmanlı ağlardan bile daha az karmaşıklığa sahiptir.**

# ÖZELLİK ÇIKARIMI (FEATURE EXTRACTION)

- **Özellik mühendisliği (feature engineering) veri bilimi ve makine öğrenmesi süreçlerinde oldukça önemlidir. Özellik çıkarımı ise özellik mühendisliğinin bir alt kümesidir. Özellik çıkarımı: ham veriden değişken, özellik (feature) türetmek demektir.**
- **Özellikler/öznitelikler, makine öğrenimi modellerinde bağımsız değişkenlerden başka bir şey değildir.**

**Köpeğin öne çıkan özellikleri gösterilmiştir.**





# DEVAM..

- **Bir sisteme giren girişlerin bütün bir bilgi olarak değil de bu bilgiyi oluşturan vasıflardan bazılarının çıkarılması ve sistemin bu vasıflar üzerine kurulması durumudur.**
- **Özellik çıkarımı (vasıflandırma, feature extraction) işlemi bir boyut azaltma (dimension reduction, dimensionality reduction) işlemidir. Buna göre karmaşık olan bir verinin boyutları azaltılarak daha basit bir problem haline indirgenir.**

# DEVAM..

- **Doğru yapılmış bir özellik çıkarımı ve bu özelliklere uygun bir sistem tasarımı sonucun başarılı olması ve performansını etkileyen unsurlardır.**
- **Ayrıca özellik çıkarımı sonucunda elde edilen birden fazla özelliğin karşılığını tutan veri yapısına özellik vektörü ( feature vector ) adı da verilmektedir.**

# DEVAM..

- **Özellik çıkarma, mevcut özelliklerden yeni özellikler oluşturarak (ve ardından orijinal özellikleri atarak) bir veri kümesindeki özelliklerin sayısını azaltmayı amaçlar.**
- **1. Boyutsallık azaltma ihtiyacı:** gereken zaman ve depolama alanını azaltır. Makine öğrenimi modelinin parametrelerinin yorumlanmasını iyileştiren çoklu doğrusallığı kaldırmaya yardımcı olur.

# DEVAM..

- **2. PCA (temel bileşen analizi — lineer):** veriler analiz edilirken özellik çıkarımı yapılmalıdır. Temel amacı yüksek boyutlu verilerde en yüksek varyans ile veri setini tutmak ancak bunu yaparken boyut indirgemeyi sağlamak olan bir tekniktir.
- **3. Çekirdek PCA:** kernel PCA, karar sınırları doğrusal olmayan fonksiyon tarafından tanımlanan verilerin sınıflandırılmasına yardımcı olmak amacıyla geliştirilmiştir.



# DEVAM..

- **4. LDA(lineer diskriminant analizi):** Lineer diskriminant analizi, önceden bilinen  $p$  sayıdaki özelliklerine göre birimleri, doğadaki gerçek sınıflarına en doğru şekilde atamayı amaçlayan çok değişkenli istatistiksel bir yöntemdir.

# KAYNAKÇA

- CONVOLUTIONAL NEURAL NETWORK (CONVNET YADA CNN) NEDİR, NASIL ÇALIŞIR? | BY TUNCER ERGİN | MEDIUM
- **HERKES İÇİN YAPAY ZEKA –ZAFER DEMİRKOL**
- CNN (CONVOLUTIONAL NEURAL NETWORKS) NEDİR? - TEKNOLOJİ.ORG
- VERİ BİLİMİNDE ÖZELLİK ÇIKARIMI (FEATURE EXTRACTION) | BY SERDAR TAFRALI | MEDIUM
- ÖZELLİK ÇIKARIMI (FEATURE EXTRACTION) – BİLGİSAYAR KAVRAMLARI (BİLGİSAYARKAVRAMLARI.COM)