## **Convolutional Neural Network (CNN)**

### **What is CNN?**

Since the 1980s, convolutional neural networks (CNNs) have been employed in image recognition as a result of research into the visual cortex of the brain. [1980'lerden beri, beynin görsel korteksine yapılan araştırmaların bir sonucu olarak görüntü tanımada evrişimli sinir ağları (CNN'ler) kullanılmıştır.]

A convolutional neural network (CNN, or ConvNet) is a type of artificial neural network used to interpret visual imagery in deep learning. [Evrişimsel sinir ağı (CNN veya ConvNet), derin öğrenmede görsel görüntüleri yorumlamak için kullanılan bir tür yapay sinir ağıdır.]

CNNs are different versions of multilayer perceptrons. [CNN'ler, çok katmanlı algılayıcıların farklı versiyonlarıdır.] Multilayer perceptrons are typically completely connected networks, meaning that each neuron in one layer is linked to all neurons in the following layer. [Çok katmanlı algılayıcılar tipik olarak tamamen bağlı ağlardır, yani bir katmandaki her bir nöron, bir sonraki katmandaki tüm nöronlara bağlıdır.] These networks' "complete connectedness" makes them vulnerable to data overfitting. [Bu ağların 'tam bağlılığı', onları aşırı veri uyumuna karşı savunmasız hale getirir.]

Image search services, self-driving cars, automatic video classification systems, and other applications rely on them. [Görsel arama hizmetleri, sürücüsüz arabalar, otomatik video sınıflandırma sistemleri ve diğer uygulamalar bunlara güveniyor.] Furthermore, CNNs are not limited to visual perception; they excel at a variety of other tasks as well, including speech recognition and natural language processing. [Ayrıca, CNN'ler görsel algı ile sınırlı değildir; konuşma tanıma ve doğal dil işleme dahil olmak üzere çeşitli diğer görevlerde de başarılıdırlar.]

# **Introducing convolutional neural networks (ML Zero to Hero - Part 3)**

<https://www.youtube.com/watch?v=x_VrgWTKkiM&t=7s>

### **Why Convolution? [Neden Konvolüsyon?]**

Concrete data points must be provided when using ANN. [YSA kullanılırken somut veri noktaları sağlanmalıdır.] The breadth of the snout and the length of the ears, for example, must be explicitly included as data points in a model attempting to distinguish between dogs and cats. [Örneğin, burnun genişliği ve kulakların uzunluğu, köpekler ve kediler arasında ayrım yapmaya çalışan bir modelde veri noktaları olarak açıkça dahil edilmelidir.]

These spatial properties are extracted from image input when CNN is used. [Bu uzamsal özellikler, CNN kullanıldığında görüntü girişinden çıkarılır.] When thousands of features need to be extracted, CNN is suitable. [Binlerce özelliğin çıkarılması gerektiğinde CNN uygundur.] CNN gathers these features on its own rather than having to measure each one individually. [CNN, her birini ayrı ayrı ölçmek zorunda kalmadan bu özellikleri kendi başına toplar.]

Because 2-dimensional images must be translated to 1-dimensional vectors when using ANN, image classification tasks become more challenging. [YSA kullanılırken 2 boyutlu görüntülerin 1 boyutlu vektörlere çevrilmesi gerektiğinden, görüntü sınıflandırma görevleri daha zor hale gelir.] This rapidly increases the number of trainable parameters. [Bu, eğitilebilir parametrelerin sayısını hızla artırır.] Increasing the number of trainable parameters necessitates more storage and processing power. [Eğitilebilir parametre sayısının arttırılması, daha fazla depolama ve işlem gücü gerektirir.]

Assume an image with dimensions of 68 X 68 X 3. [68 X 68 X 3 boyutlarında bir görüntü varsayalım.] The dimension of the input feature is now 12,288. [Giriş özelliğinin boyutu artık 12.288'dir.] If we have larger images, this will be even bigger. [Daha büyük resimlerimiz varsa, bu daha da büyük olacaktır.] Now, if we provide a neural network such a large input, the number of parameters will skyrocket (depending on the size of the input and neural network). [Şimdi, bir sinir ağına bu kadar büyük bir girdi sağlarsak, parametre sayısı (girişin ve sinir ağının boyutuna bağlı olarak) fırlayacaktır.]  More processing and memory requirements will follow, which most of us will be unable to handle. [Çoğumuzun üstesinden gelemeyeceği daha fazla işlem ve bellek gereksinimi takip edecek.]

# **C4W1L11 Why Convolutions**

<https://www.youtube.com/watch?v=ay3zYUeuyhU>

### **Types of CNNs [CNN Türleri]**

1. **1D CNN**: Kernel (filter) moves in only one direction with them. [1D CNN: Çekirdek (filtre) onlarla yalnızca bir yönde hareket eder.] These kinds of CNNs are typically applied to time-series data. [Bu tür CNN'ler tipik olarak zaman serisi verilerine uygulanır.]
2. **2D CNN**: Kernels moves in two directions. [2D CNN: Çekirdekler iki yönde hareket eder.] These are commonly used in picture labeling and processing. [Bunlar genellikle resim etiketleme ve işlemede kullanılır.]
3. **3D CNN**: Kernel travels in three directions. [3D CNN: Çekirdek üç yönde hareket eder.] This form of CNN commonly used on 3D images such as CT scans and MRIs. [Bu CNN formu, BT taramaları ve MRI'lar gibi 3B görüntülerde yaygın olarak kullanılır.]

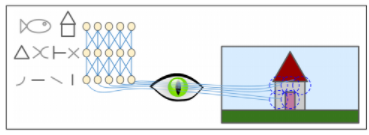
Most of the time, you'll run into 2D CNNs because they're frequently used with image data. [Çoğu zaman, görüntü verileriyle sıklıkla kullanıldıkları için 2D CNN'lerle karşılaşırsınız.] Some of the applications of CNNs are listed below. [CNN'lerin bazı uygulamaları aşağıda listelenmiştir.]

* Image recognition with little preprocessing [Küçük ön işleme ile görüntü tanıma]
* Recognize handwriting [el yazısını tanımak]
* Applications for computer vision [Bilgisayarla görme uygulamaları]
* In banking, it is used to read the numbers on checks. [Bankacılıkta çeklerdeki rakamları okumak için kullanılır.]
* In postal services, it is used to read zip numbers on envelopes. [Posta hizmetlerinde, zarfların üzerindeki zip numaralarını okumak için kullanılır.]

## **Architecture of CNNs**

### **Inspiration from Visual Cortex Architecture [Görsel Korteks Mimarisinden İlham]**

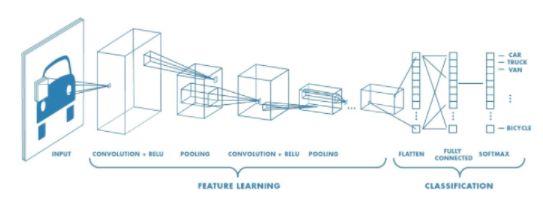
It is discovered that many neurons in the brain’s visual cortex have a tiny local receptive field, which means that they only respond to visual stimuli in a specific portion of the visual field. [Beynin görsel korteksindeki birçok nöronun küçük bir yerel alıcı alana sahip olduğu keşfedildi, bu da onların görsel alanın yalnızca belirli bir bölümündeki görsel uyaranlara tepki verdiği anlamına geliyor.] Different neurons' receptive fields may overlap, and when they do, they tile the entire visual field. [Farklı nöronların alıcı alanları örtüşebilir ve örtüştüklerinde tüm görsel alanı döşerler.]



Furthermore, the researchers discovered that some neurons only respond to images of horizontal lines, while others only respond to lines of varying orientations. [Ayrıca araştırmacılar, bazı nöronların yalnızca yatay çizgilerin görüntülerine tepki verdiğini, diğerlerinin ise yalnızca farklı yönlerdeki çizgilere tepki verdiğini keşfettiler.] They also discovered that some neurons have bigger receptive fields and respond to more complicated patterns that are a mixture of lower-level patterns. [Ayrıca bazı nöronların daha büyük alıcı alanlara sahip olduğunu ve daha düşük seviyeli kalıpların bir karışımı olan daha karmaşık kalıplara tepki verdiğini keşfettiler.] These findings lead to the hypothesis that higher-level neurons are predicated on the outputs of lower-level neurons. [Bu bulgular, üst düzey nöronların alt düzey nöronların çıktılarına dayandığı hipotezine yol açar.]

### **CNN Architecture**

Deep Learning CNN models takes each input image and passes it through a sequence of convolution layers using filters(kernels) , Pooling layers, fully connected layers (FC) and finally uses a classification function to identify the object.



A typical CNN architecture consists of a few convolutional layers (each followed by a ReLU layer), a pooling layer, , another few convolutional layers (+ReLU), another pooling layer, and so on. Because of the convolutional layers, the image shrinks as it passes through the network, but it also gets deeper and catches more features. After convolution and pooling layers  a feedforward neural network with a few fully connected layers (+ReLUs) is added, and the prediction is made by the final layer of a classification function.

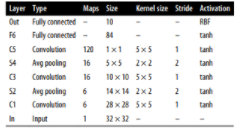
# **5. CNN architecture in 5 minutes - A Quick Learning**

<https://www.youtube.com/watch?v=R9CSbcGxKzo>

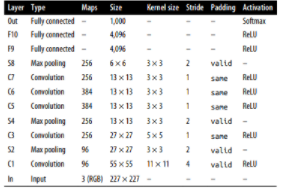
### **Classical CNN Models [Klasik CNN Modelleri]**

Variations of this core architecture have been evolved over time, resulting in incredible advancements in the discipline. [Bu çekirdek mimarinin varyasyonları zaman içinde gelişti ve bu da disiplinde inanılmaz gelişmelere yol açtı.]  Understanding how CNNs work can be learned by looking at those models. 5 famous models are introduced below. [CNN'lerin nasıl çalıştığını anlamak, bu modellere bakarak öğrenilebilir. Aşağıda 5 ünlü model tanıtılmıştır.]

1. Perhaps the most well-known CNN architecture is the **LeNet-5** architecture. [Belki de en iyi bilinen CNN mimarisi LeNet-5 mimarisidir.] Yann LeCun invented it in 1998, and it's been frequently utilized for handwritten digit recognition since then (MNIST). [Yann LeCun bunu 1998'de icat etti ve o zamandan beri (MNIST) elle yazılmış rakam tanıma için sıklıkla kullanılıyor.]



1. **AlexNet**was the first to stack convolutional layers directly on top of one another, rather than stacking a pooling layer on top of each conv layer. [AlexNet, her bir dönüşüm katmanının üzerine bir havuz katmanı yerleştirmek yerine, evrişim katmanlarını doğrudan birbirinin üzerine istifleyen ilk kişiydi.]

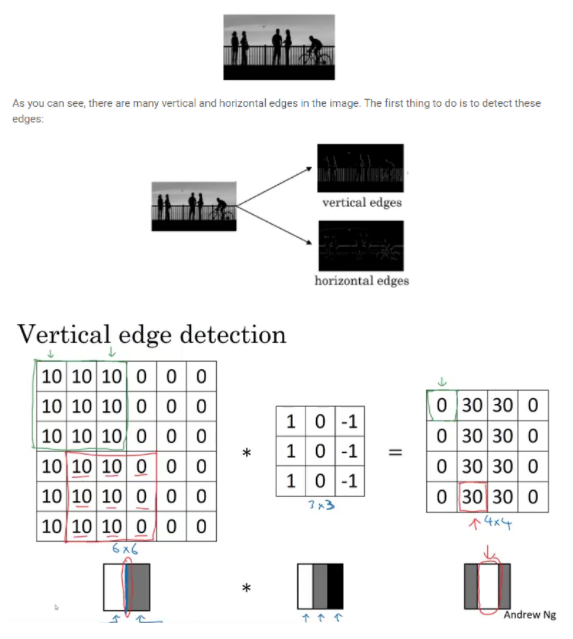


1. **GoogLeNet** uses inception modules, which are subnetworks that allow it to use parameters considerably more efficiently than earlier architectures: In fact, GoogLeNet has ten times fewer parameters than AlexNet (roughly 6 million instead of 60 million). [GoogLeNet, parametreleri önceki mimarilerden çok daha verimli kullanmasına izin veren alt ağlar olan başlangıç ​​modüllerini kullanır: Aslında, GoogLeNet, AlexNet'ten on kat daha az parametreye sahiptir (60 milyon yerine yaklaşık 6 milyon).]
2. **VGGNet**has a very simple and classic architecture, 2/3 convolutional layers and a pooling layer, then 2/3 convolutional layers and a pooling layer, and so on (16/19 convolutional layers in total depending on the variant). [VGGNet, çok basit ve klasik bir mimariye, 2/3 evrişim katmanlarına ve bir havuzlama katmanına, ardından 2/3 evrişim katmanlarına ve bir havuzlama katmanına vb. sahiptir (versiyona bağlı olarak toplamda 16/19 evrişim katmanı).] Finally there is a dense network with 2 hidden layers and the output layer. [Son olarak, 2 gizli katman ve çıktı katmanından oluşan yoğun bir ağ vardır.] It uses a lot of but only 3x3 size filters. [Çok fazla ancak yalnızca 3x3 boyutunda filtre kullanır.]
3. **Residual Network (or ResNet)**composed of 152 layers (other variants had 34, 50, and 101 layers). [152 katmandan oluşan Artık Ağ (veya ResNet) (diğer varyantlarda 34, 50 ve 101 katman vardı).]  It confirmed a common trend: models are becoming more complex and containing fewer parameters. [Ortak bir eğilimi doğruladı: modeller daha karmaşık hale geliyor ve daha az parametre içeriyor.] The use of skip connections (or shortcut connections) is important to being able to train such a deep network: the signal entering into one layer is also added to the output of a layer a little higher up the stack. [Bu kadar derin bir ağı eğitebilmek için atlama bağlantılarının (veya kısayol bağlantılarının) kullanılması önemlidir: bir katmana giren sinyal aynı zamanda yığının biraz yukarısındaki bir katmanın çıkışına eklenir.]

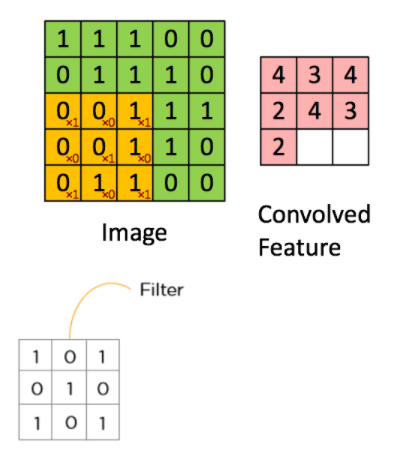
## **Filters**

### **What is a filter? [Filtre nedir?]**

Filters in Convolutional Neural Networks identify spatial patterns such as edges in an image by detecting changes in the picture's intensity values. [Evrişimli Sinir Ağlarındaki filtreler, resmin yoğunluk değerlerindeki değişiklikleri algılayarak bir görüntüdeki kenarlar gibi uzamsal kalıpları tanımlar.]

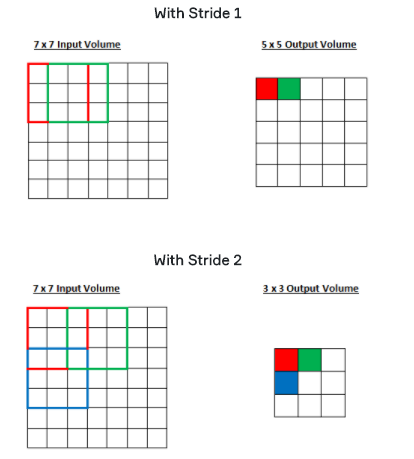


Convolution of an image with various filters can be used to accomplish tasks such as edge detection, blurring, and sharpening. [Kenar algılama, bulanıklaştırma ve keskinleştirme gibi görevleri gerçekleştirmek için çeşitli filtrelerle bir görüntünün evrişimi kullanılabilir.]

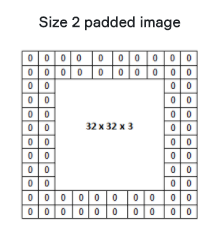


**Stride and Padding**

**Stride** determines how the filter convolves around the input volume. In the first portion of the example, the filter convolved around the input volume by moving one unit at a time. The stride is the amount by which the filter shifts. Programmers will raise the stride if they want the receptive fields to overlap less and the spatial dimensions to be smaller.



**Padding,** which is a feature that adds blank or empty pixels to the image frame works in conjunction with stride, to allow for a minimal reduction in size in the output layer. It is a method of enlarging the size of a picture to compensate for the fact that stride lowers the size.



# **Convolution padding and stride | Deep Learning Tutorial 25 (Tensorflow2.0, Keras & Python)**

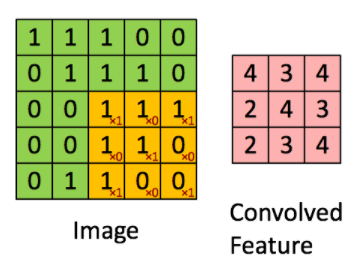
<https://www.youtube.com/watch?v=oDAPkZ53zKk>

## **Convolutional Layers**

### **Convolution Layer [Evrişim Katmanı]**

Most of the original images have the feature of being “stationary,” which means that the statistics of one portion of the image are the same as the statistics of any other section of the image. [Orijinal görüntülerin çoğu, “durağan” olma özelliğine sahiptir; bu, görüntünün bir bölümünün istatistiklerinin görüntünün başka herhangi bir bölümünün istatistikleriyle aynı olduğu anlamına gelir.] This implies that the characteristics we learn in one portion of the image may be transferred to other sections of the image, and that the same features can be used in all places. [Bu, görüntünün bir bölümünde öğrendiğimiz özelliklerin görüntünün diğer bölümlerine aktarılabileceğini ve aynı özelliklerin her yerde kullanılabileceği anlamına gelir.]

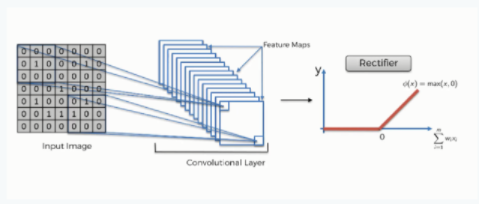
More specifically, after learning features across tiny (say, 5x5) patches randomly selected from the overall picture, we may use this learnt 5x5 feature detector everywhere in the image. [Daha spesifik olarak, genel resimden rastgele seçilen küçük (örneğin 5x5) yamalar üzerindeki özellikleri öğrendikten sonra, bu öğrenilmiş 5x5 özellik dedektörünü görüntünün her yerinde kullanabiliriz.] Specifically, we may “convolve” the learnt 5x5 features with the bigger picture, resulting in a new feature activation value at each place in the image. [Spesifik olarak, öğrenilen 5x5 özellikleri daha büyük resimle 'dönüştürebiliriz', bu da görüntüdeki her yerde yeni bir özellik etkinleştirme değeriyle sonuçlanır.]



# **Convolutional Neural Networks Explained (CNN Visualized)**

<https://www.youtube.com/watch?v=pj9-rr1wDhM&t=28s>

### **RELU Layer [RELU Katmanı]**



The rectifier function is used to augment the nonlinearity of our pictures. [Doğrultucu işlevi, resimlerimizin doğrusal olmama durumunu artırmak için kullanılır.] We want to do this since pictures are naturally non-linear. [Resimler doğal olarak doğrusal olmadığı için bunu yapmak istiyoruz.]  When you look at any image, you will see that it has a number of non-linear characteristics (e.g. the transition between pixels, the borders, the colors, etc.). [Herhangi bir görüntüye baktığınızda, bir dizi doğrusal olmayan özelliğe sahip olduğunu göreceksiniz (örn. pikseller arası geçiş, kenarlıklar, renkler vb.).]

The rectifier breaks up the linearity even further to compensate for the linearity that we could impose on a picture when we run it through the convolution process. [Doğrultucu, bir resme evrişim sürecinden geçirdiğimiz zaman yükleyebileceğimiz doğrusallığı telafi etmek için doğrusallığı daha da bozar.] To illustrate how it works in practice, look at the image below and see the changes that occur as it goes through the convolution process followed by rectification. [Pratikte nasıl çalıştığını göstermek için aşağıdaki resme bakın ve evrişim sürecinden geçerken meydana gelen değişiklikleri ve ardından düzeltmeyi görün.]



**After applying to a feature detector (**Pixels vary with many shades of gray**) [Bir özellik algılayıcıya uygulandıktan sonra (Pikseller grinin birçok tonuyla değişir)]**



The rectifier function removes all of the dark components from this picture, leaving only those with a positive value (the grey and white colors). [Doğrultucu işlevi, bu resimdeki tüm koyu bileşenleri kaldırır ve yalnızca pozitif değere sahip olanları (gri ve beyaz renkler) bırakır.]  The colors will change more abruptly once we correct the picture. [Resmi düzelttiğimizde renkler daha ani bir şekilde değişecektir.] The progressive transformation is no longer present. [Aşamalı dönüşüm artık mevcut değil.] This implies that the linearity has been eliminated. [Bu, doğrusallığın ortadan kaldırıldığı anlamına gelir.]



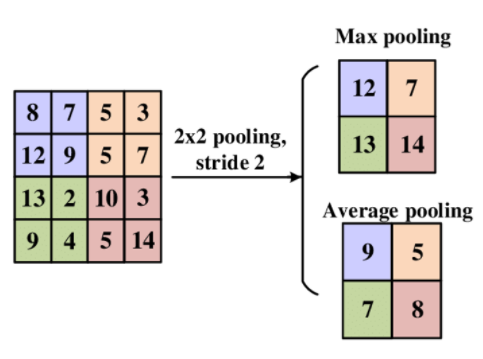
## **Pooling Layer, Dropout Layer and Fully Connected Layer**

### **Pooling Layer [Havuzlama Katmanı]**

We would like to use the features obtained using convolution for classification after they have been extracted. [Evrişim kullanılarak elde edilen öznitelikleri, çıkarıldıktan sonra sınıflandırma için kullanmak istiyoruz.] In theory, all of the collected features may be used with a classifier, such as a softmax classifier, but this can be computationally challenging. [Teoride, toplanan özniteliklerin tümü, softmax sınıflandırıcı gibi bir sınıflandırıcı ile kullanılabilir, ancak bu, hesaplama açısından zorlayıcı olabilir.]

To solve this, remember that we chose convolved features because pictures have the “stationarity” trait, which means that features that are beneficial in one location are likely to be useful in other regions as well. [Bunu çözmek için, resimlerin 'durağanlık' özelliğine sahip olması nedeniyle kıvrımlı özellikleri seçtiğimizi unutmayın; bu, bir konumda yararlı olan özelliklerin diğer bölgelerde de yararlı olacağı anlamına gelir.] As a result, one simple method to describing a big image is to aggregate statistics of these characteristics at multiple places. [Sonuç olarak, büyük bir görüntüyü tanımlamanın basit bir yöntemi, bu özelliklerin istatistiklerini birden çok yerde toplamaktır.] For example, one might compute the mean (or maximum) value of a certain feature over an image region. [Örneğin, bir görüntü bölgesi üzerinden belirli bir özelliğin ortalama (veya maksimum) değeri hesaplanabilir.] These summary statistics have a reduced dimension (when compared to utilizing all of the collected characteristics) and can enhance outcomes (less over-fitting). [Bu özet istatistiklerin boyutu küçültülür (toplanan tüm özelliklerin kullanılmasıyla karşılaştırıldığında) ve sonuçları iyileştirebilir (daha az fazla uydurma).] This aggregation operation is known as "pooling"', or "average pooling" or "max pooling". [Bu toplama işlemi 'havuzlama' veya 'ortalama havuzlama' veya 'maksimum havuzlama' olarak bilinir.]

In summary to decrease the size of the feature maps, pooling layers are employed. [Özetle, özellik haritalarının boyutunu azaltmak için havuzlama katmanları kullanılır.] As a result, it decreases the number of parameters to learn as well as the amount of computation done in the network. [Sonuç olarak, ağda yapılan hesaplama miktarı kadar öğrenilecek parametre sayısını da azaltır.]



### **Dropout Layer [Bırakma Katmanı]**

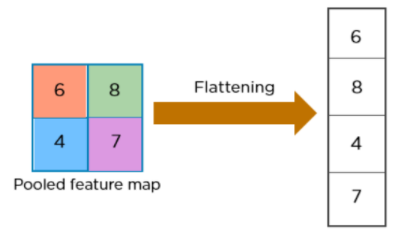
**Adding Dropout [Bırakma Ekleme]**

To some extent, you may need to overcome the problem of overfitting by adding a dropout layer. [Bir dereceye kadar, bir bırakma katmanı ekleyerek fazla takma sorununun üstesinden gelmeniz gerekebilir.] Dropout switches off a proportion of neurons at random throughout the training process, lowering the reliance on the training data by a small amount. [Bırakma, eğitim süreci boyunca rastgele nöronların bir kısmını kapatır ve eğitim verilerine olan güveni küçük bir miktar azaltır.] A hyperparameter determines how many fractions of neurons you want to turn off and may be adjusted accordingly. [Bir hiperparametre, nöronların kaç fraksiyonunu kapatmak istediğinizi belirler ve buna göre ayarlanabilir.] In this manner, turning off random neurons will prevent the network from memorizing the training data since not all neurons will be active at the same time, and the inactive neurons would be unable to learn anything. [Bu şekilde, rastgele nöronların kapatılması, tüm nöronlar aynı anda aktif olmayacağından ve aktif olmayan nöronlar hiçbir şey öğrenemeyeceğinden, ağın eğitim verilerini ezberlemesini önleyecektir.]

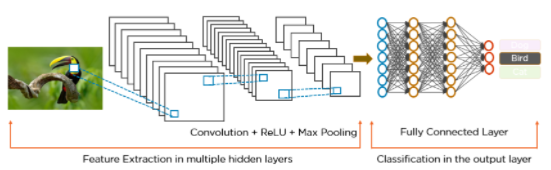
Dropout layer is usually used after pooling layers and dense layers in ConvNets. [Bırakma katmanı genellikle ConvNets'te katmanları ve yoğun katmanları bir araya getirdikten sonra kullanılır.]

### **Fully Connected Layer [Tam Bağlantılı Katman]**

Flattening is the next step after pooling layer in the procedure. [Düzleştirme, prosedürde katmanın havuzlanmasından sonraki adımdır.] Flattening is a technique for combining all of the resulting 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. [Düzleştirme, havuzlanmış özellik haritalarından elde edilen tüm 2 Boyutlu dizileri tek bir uzun sürekli doğrusal vektörde birleştirmek için bir tekniktir.]



To identify the image, the flattened matrix is supplied as input to the fully connected layer. [Görüntüyü tanımlamak için düzleştirilmiş matris, tam bağlı katmana girdi olarak sağlanır.]



# **Convolutional Neural Networks Explained (CNN Visualized)**

<https://www.youtube.com/watch?v=pj9-rr1wDhM&t=309s>