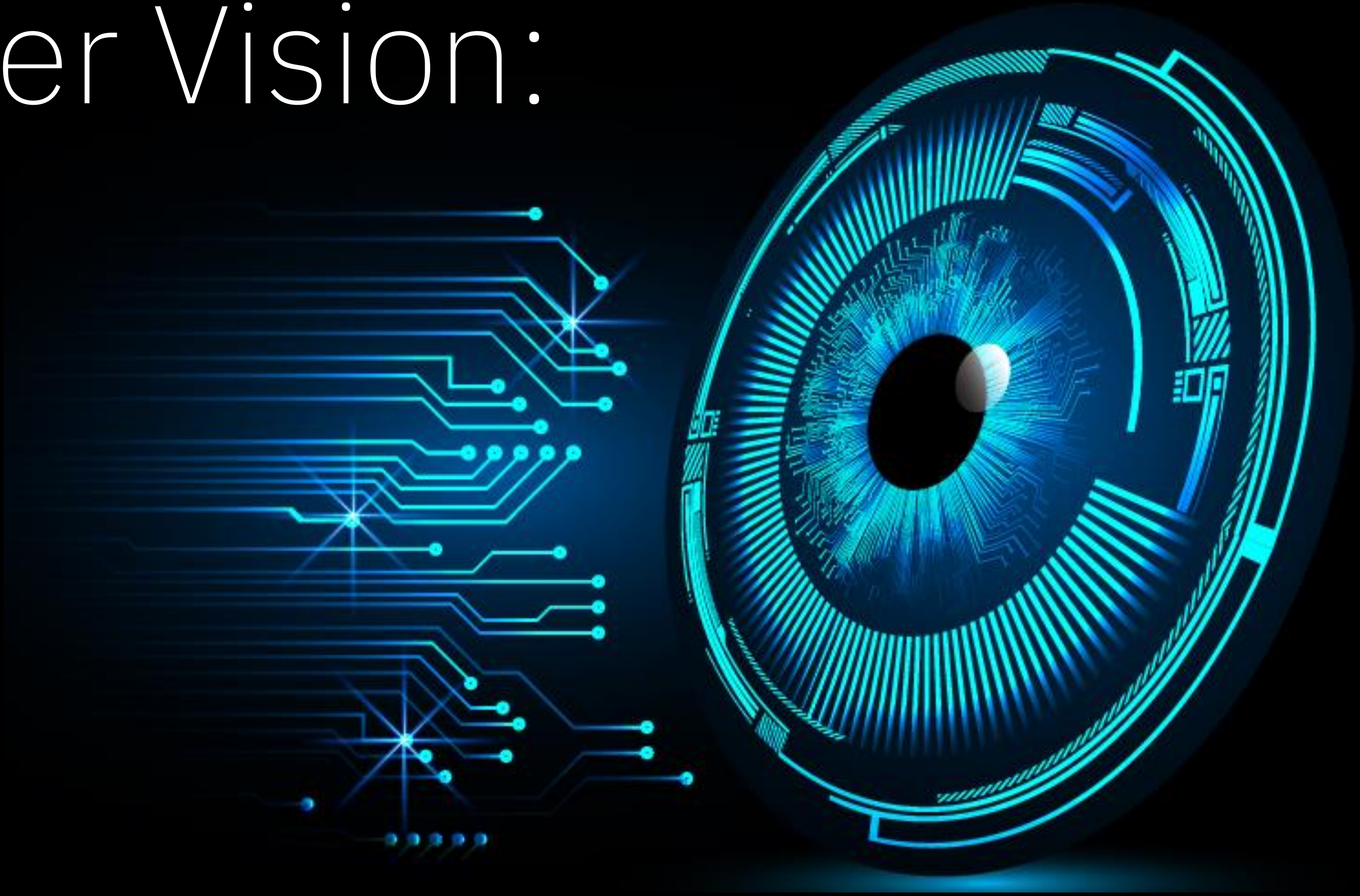
Deep Learning for Computer Vision: CNN



Computer Vision: Explained

Training computer vision models is done in much the same way as teaching children about objects visually.

Instead of a person being shown physical items and having them identified, however, the computer vision algorithms are provided many examples of images that have been tagged with their contents.

In addition to these positive examples, negative examples are also added to the training.

For example, if we're training for images of cars, we may also include negative examples of airplanes, trucks, and even boats.

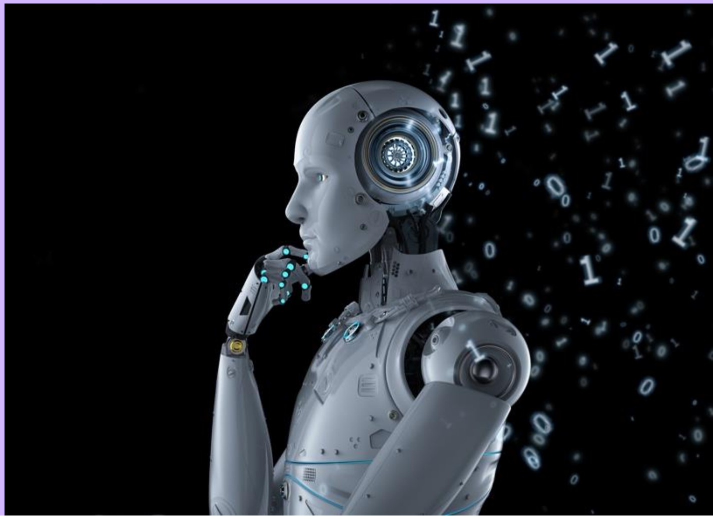


Image Classification and Tagging

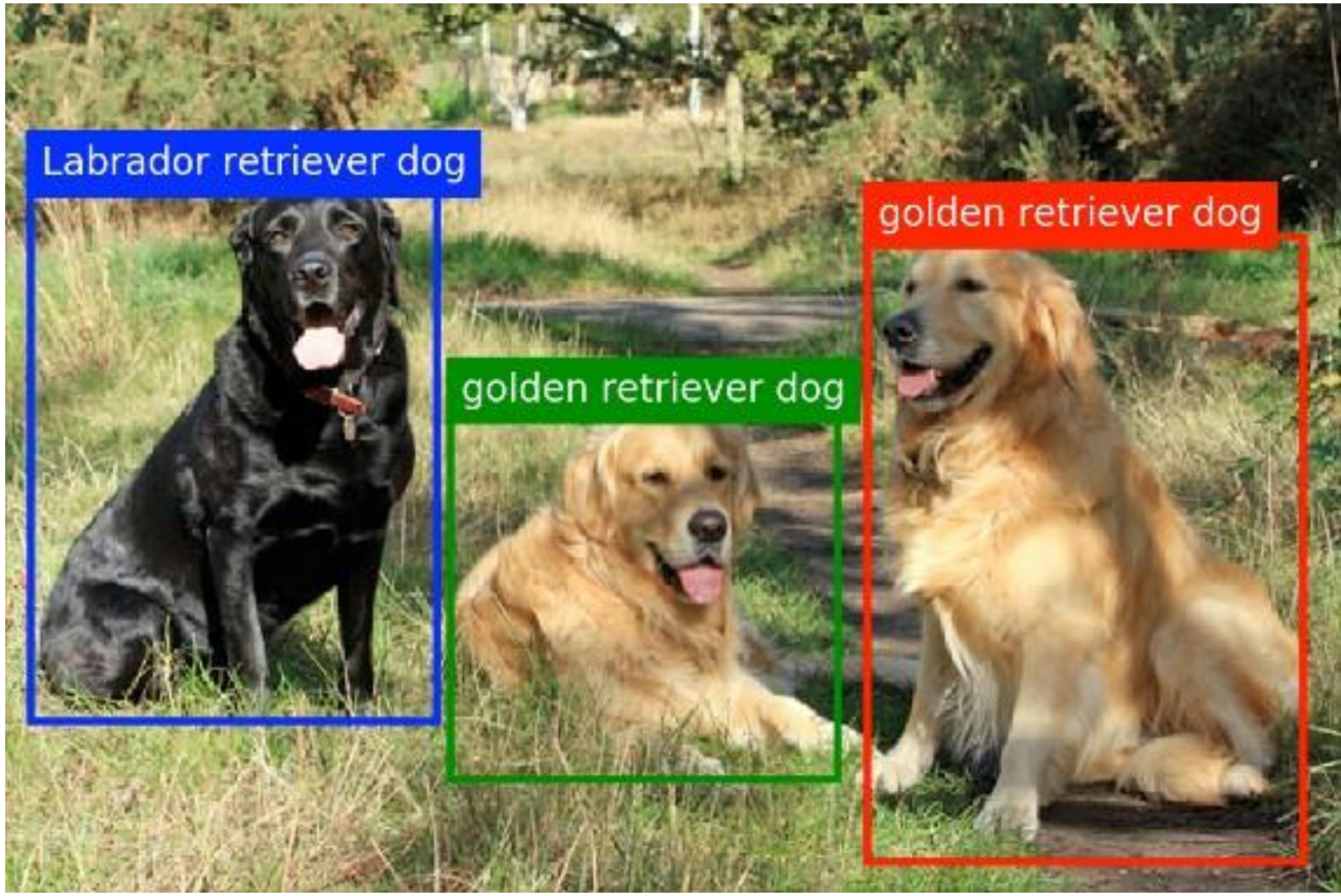
Computer vision's most core functionality, *general image tagging and classification*, allows users to understand the content of an image.

For example
You may need to find images with contents of "male playing soccer outside" or organize images into visual themes such as cars, sports, or fruits.

Object Localization

Sometimes your application's requirements will include not just classifying what is in the image

But also understanding the position of the particular object in relation to everything else

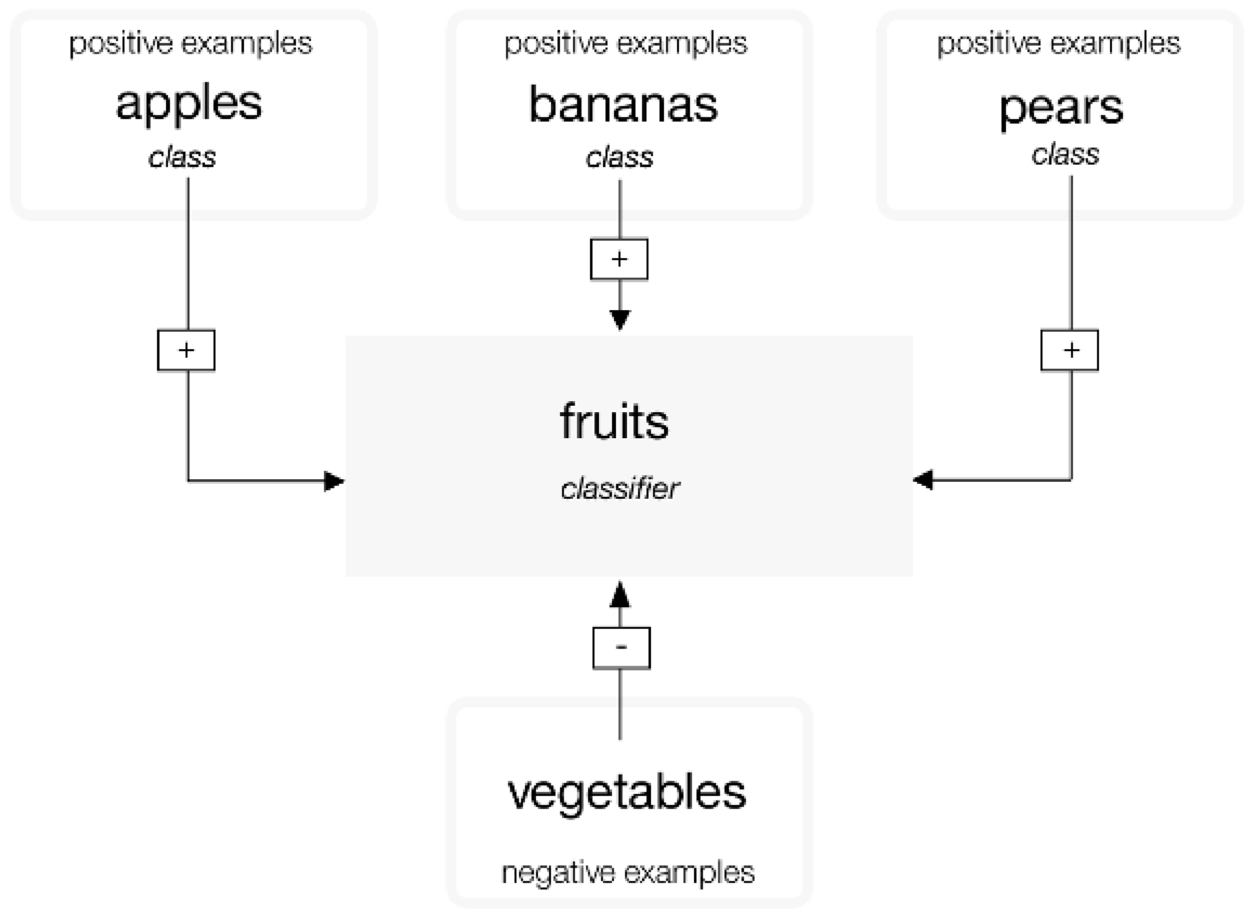


Custom Classifiers

If you're looking to identify or classify only a small set of objects, custom classifiers could be the right tool.

Most of the large third-party platforms provide some mechanism for building custom visual classifiers, allowing you to train the computer vision algorithms to recognize specific content within your images

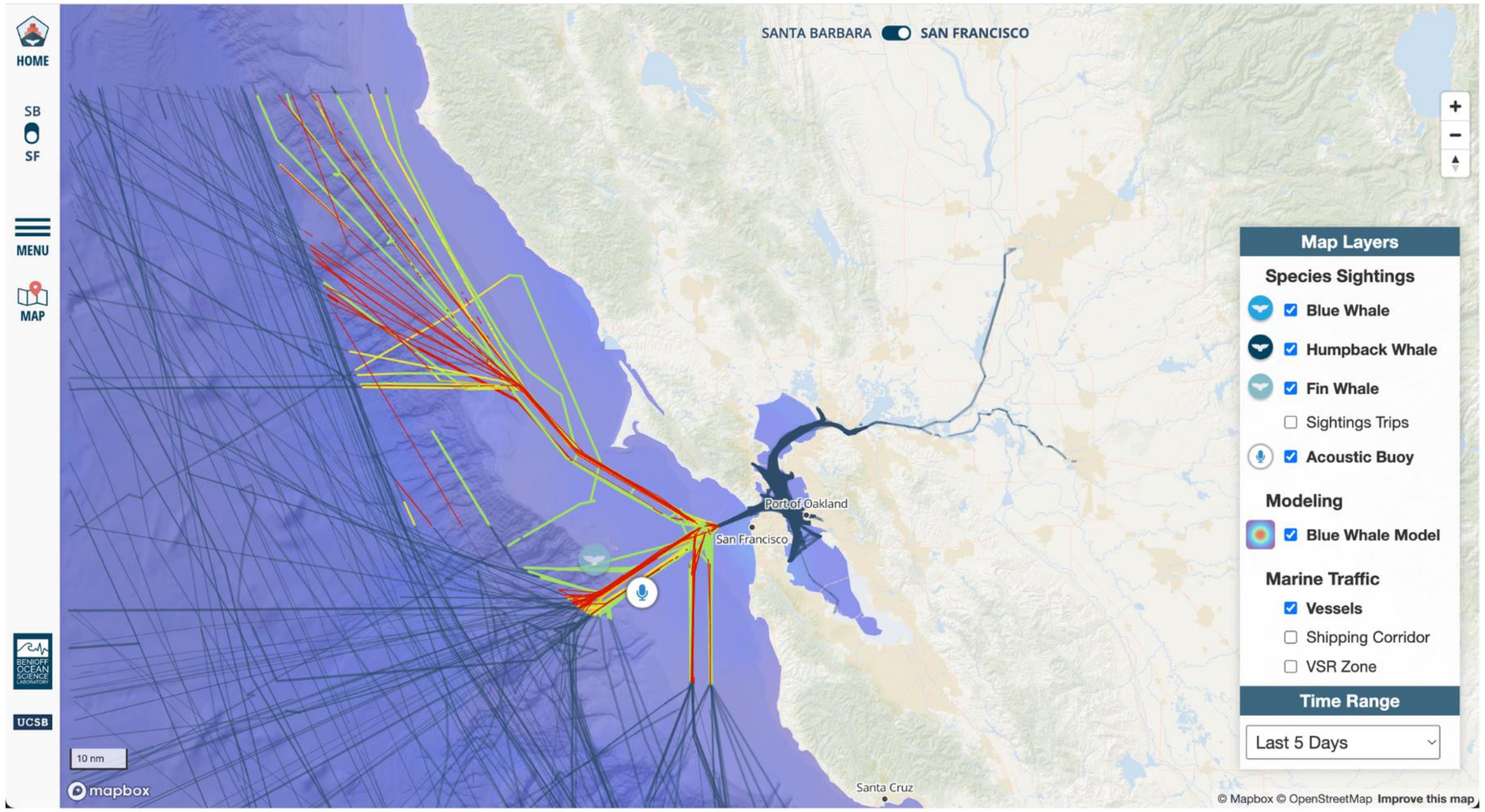
For example,
If you were training a custom classifier on fruits, you'd want to have positive training images of apples, bananas, and pears. For negative examples, you could have pictures of vegetables, cheese, or meat.



Use Cases: Satellite Imaging

Whales do not travel the same migratory path from nutrient-rich Antarctica to fertile grounds near the equator anymore.

Changing water temperatures have damaged the coral reef and altered the mating path of other fish that they feed on.



https://wwfwhales.org/news-stories/whale-trails

A screen view of the Whale Safe tool by Benioff Ocean Science Laboratory and UCSB. © Mapbox / Open Street Map / Whale Safe

Watch the whales, they know how Earth is changing!

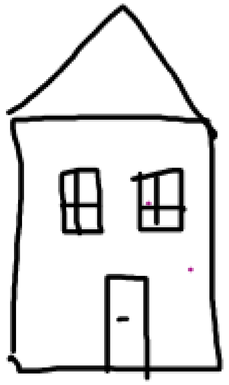
Use Cases: Video Search in Surveillance and Entertainment

The proliferation of cameras in recent years has led to an explosion in video data

In home surveillance videos, the only viable solution is still human monitoring 24/7, watching screens and raising an alert when something happens.

BlueChasm's product, VideoRecon, can watch and listen to videos, identifying key objects, themes, or events within the footage. It will then tag and timestamp those events, and then return the metadata to the end user

Architectural Building Blocks of Convolutional Neural Network: CNN

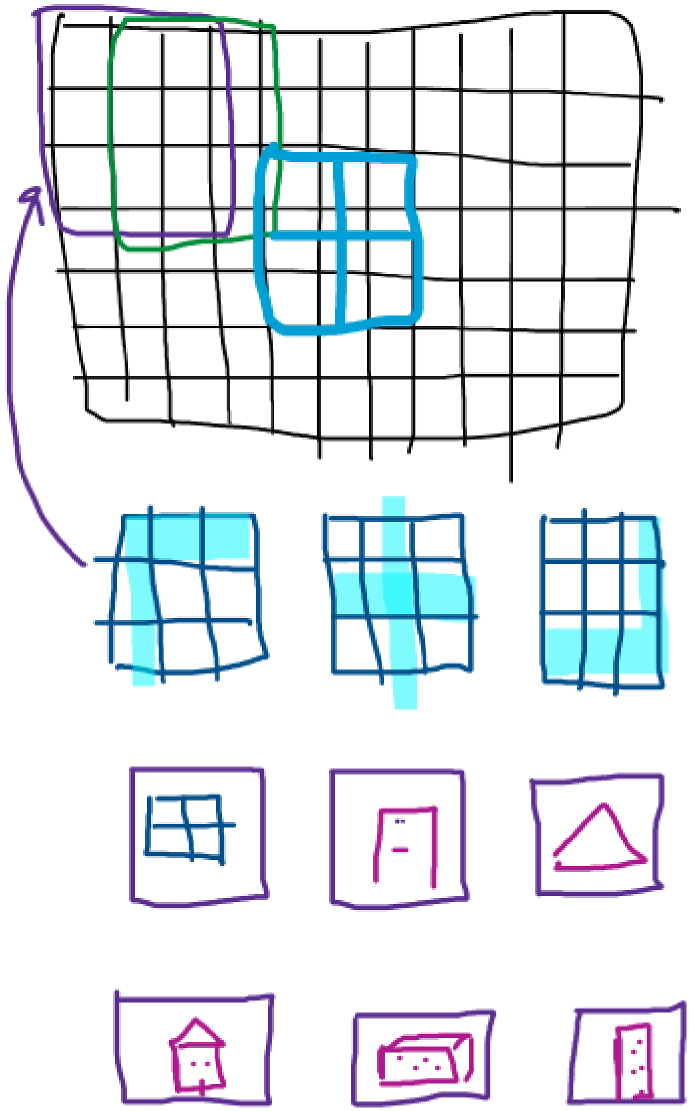


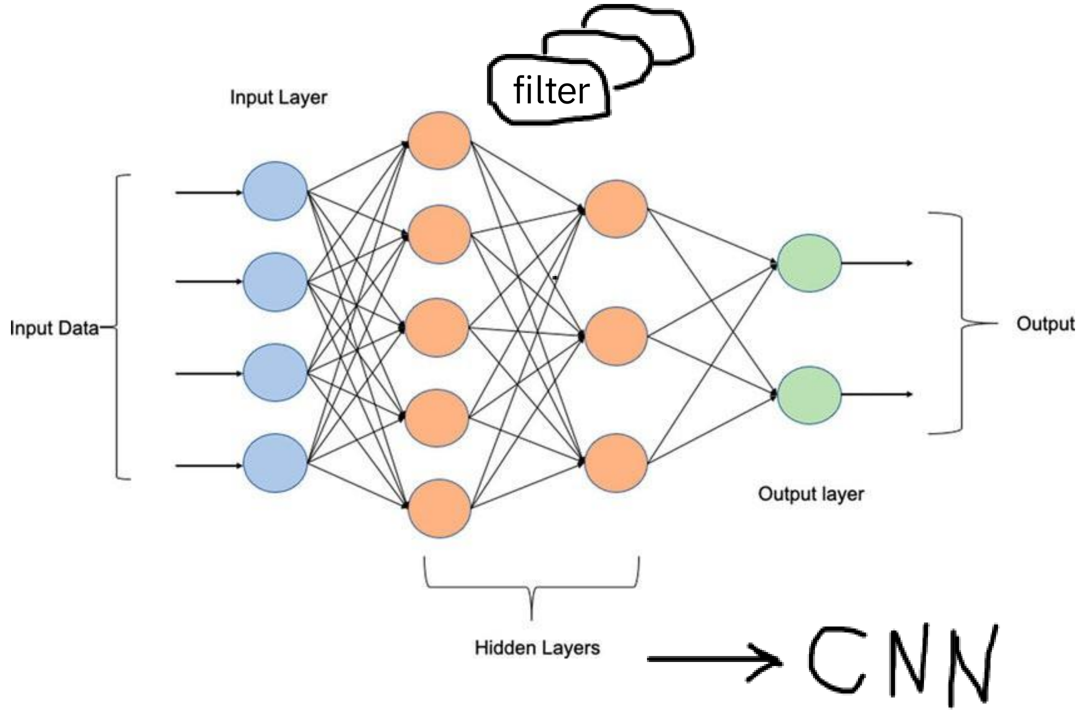
Three Predictions what this is:

This is a house

It came very easily to you and me, as to what it is

Not a very good artist

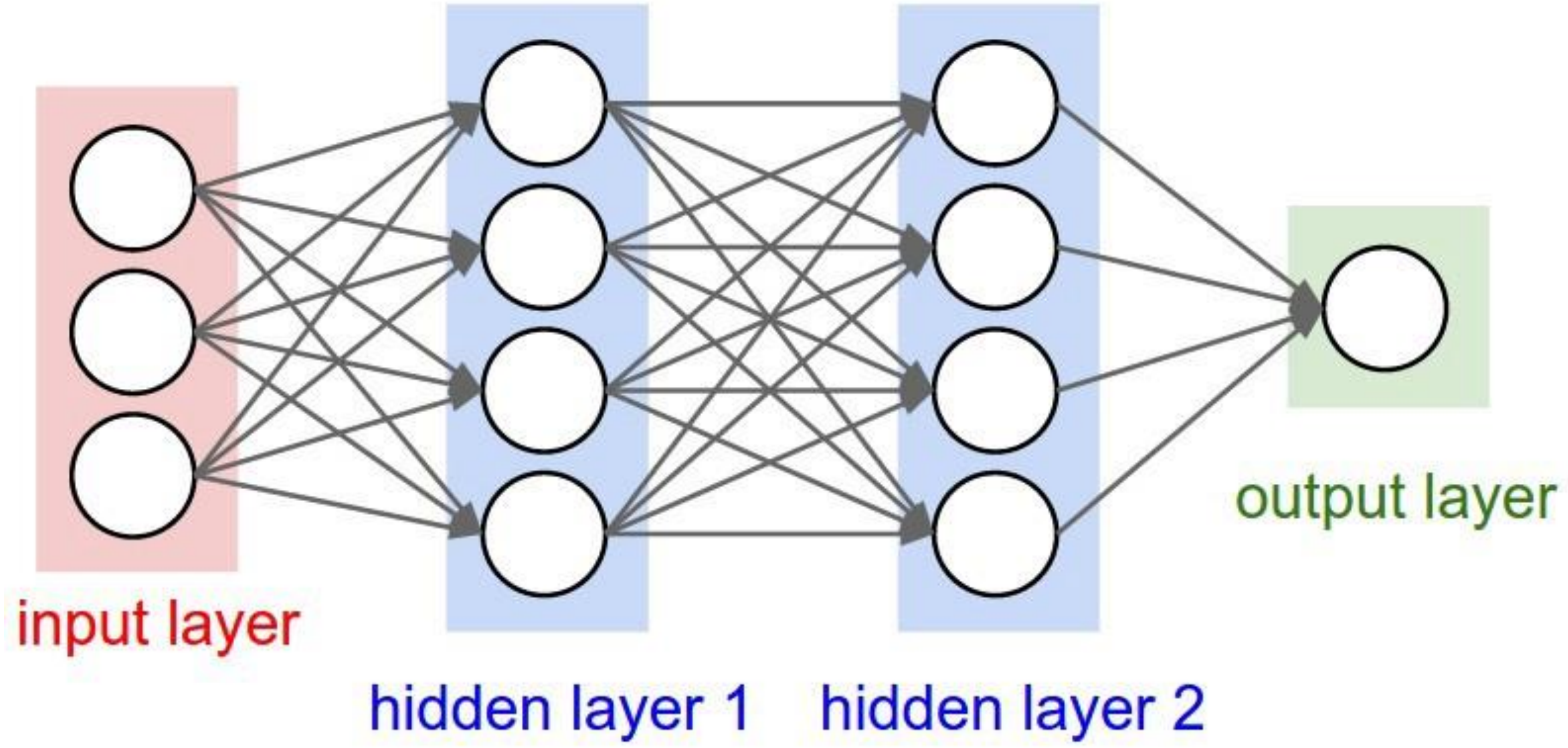




https://youtu.be/QzY57FaENXg

How do pixels and features turn into vectors? 1-hot encoding

Labeled input → supervised learning



What if your data comprise images, tweets, videos?

[x,x,x] [x,x,x] [x,x,x]

[1,0,0] [0,1,0] [0,0,1]

Index or element in a vector

1-hot encoding allows you to encode categorical data into numbers

1-Hot Encoding:
- Create a matrix of 0's and 1's
- Make each category a column in a table
- Warning: *you must encode (n-1) categories you have in the variable*
  - Otherwise, you will have *perfect multicollinearity* and encounter mathematical issues

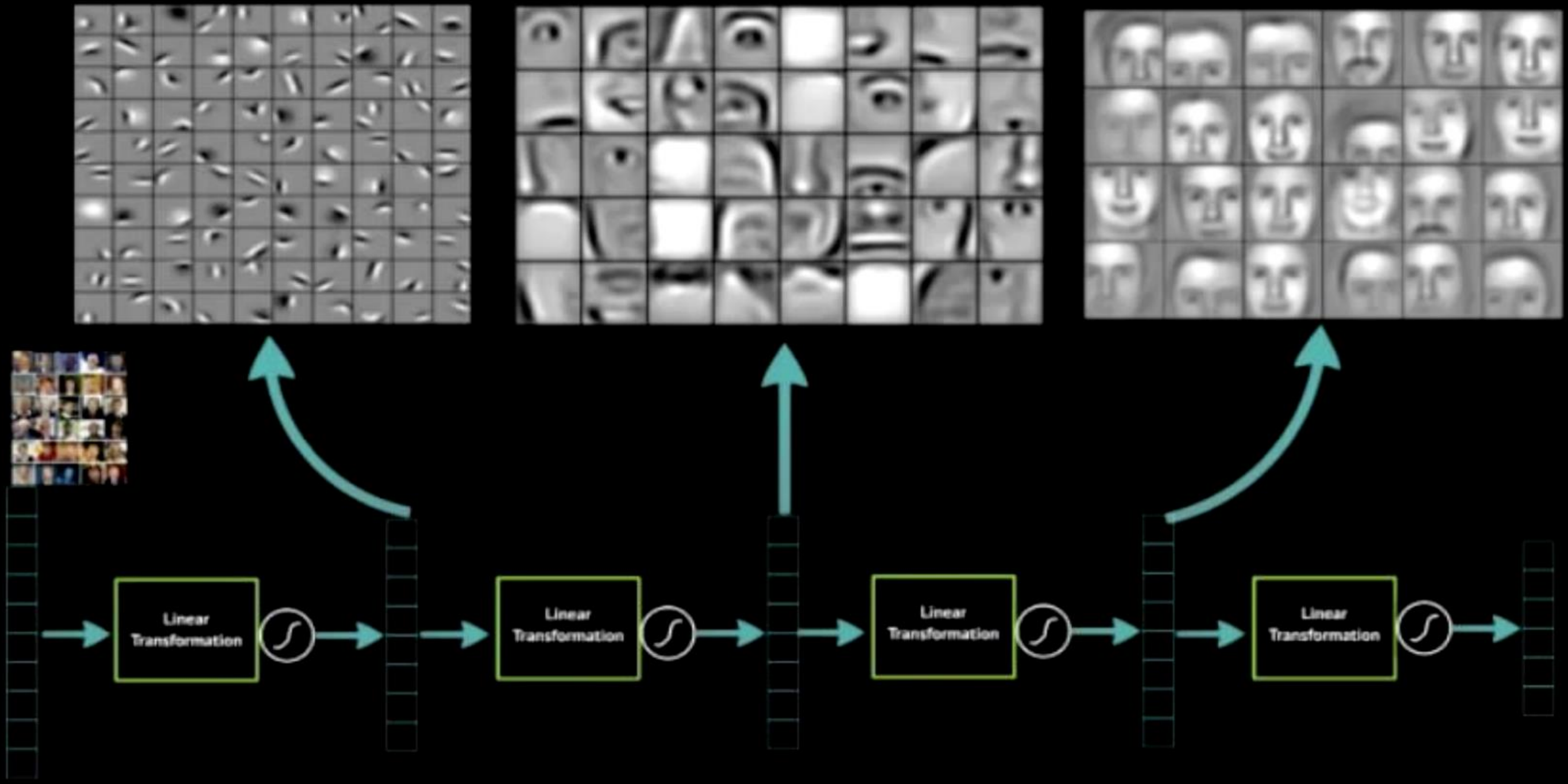
Cons:
- This makes the data very large and sparse
- Better methods for text data
- Does not scale well to big data

Solutions:
- Bag of words / TF-IDF for text data
- Advanced algorithms such as neural networks with embedding layers

Sample	Species
1	Human
2	Human
3	Penguin
4	Octopus

Sample	Human	Penguin	Octopus
1	1	0	0
2	1	0	0
3	0	1	0
4	0	0	1

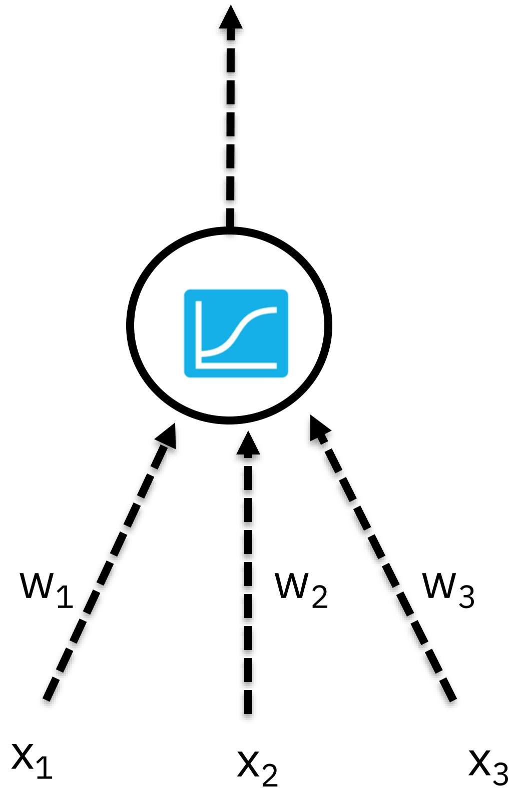
Deep Learning learns layers of features



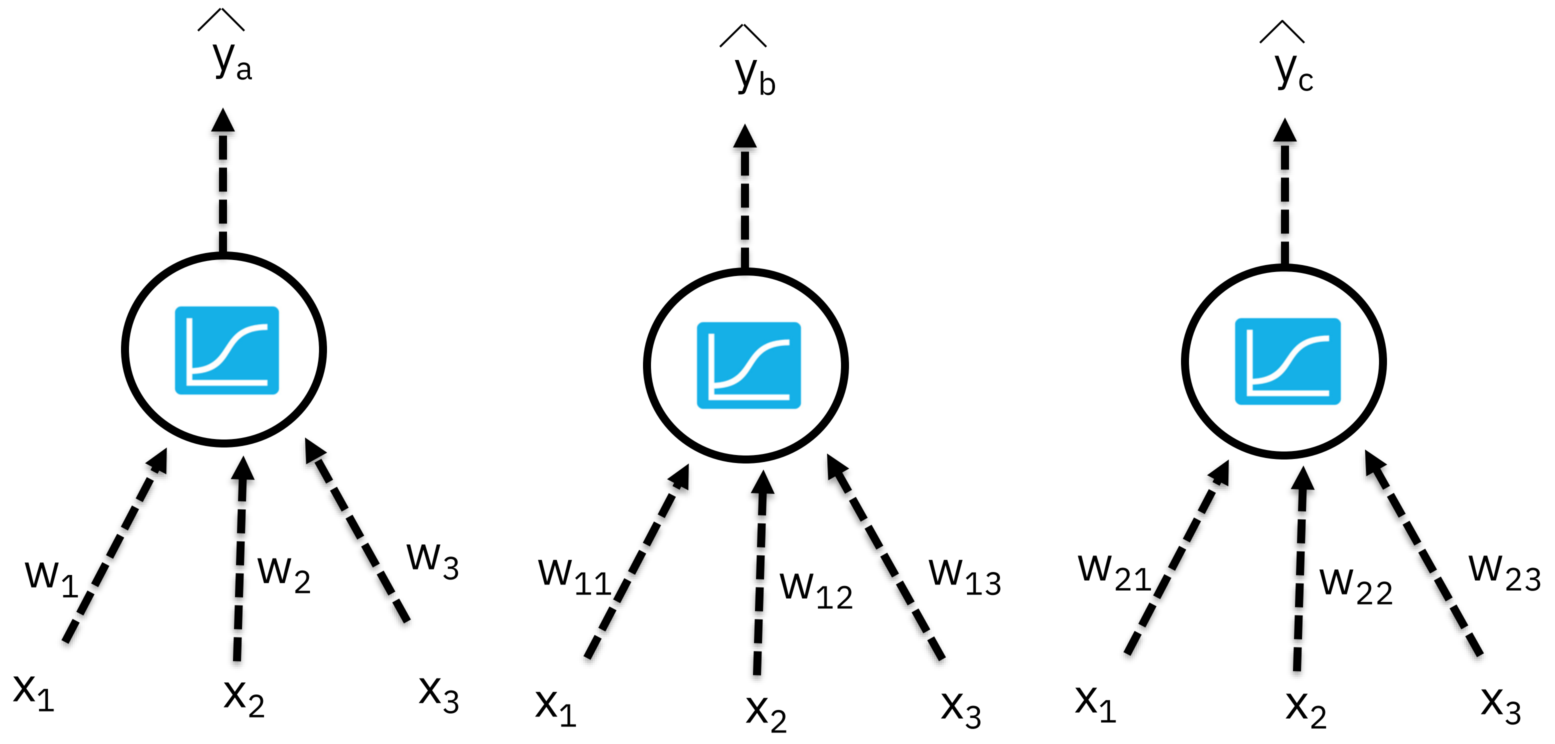
Let’s Build-up a Neural Network

○	Node (neuron)
- - - - - →	Connection (Synapse)
$x_1, x_2, x_3, \ldots. x_n$	Input
$w_1, w_2, w_3, \ldots. w_n$	Weights
$\hat{y}$	Predicted output

$\hat{Y}$ = function(x_1, x_2, x_3, w_1, w_2, w_3)

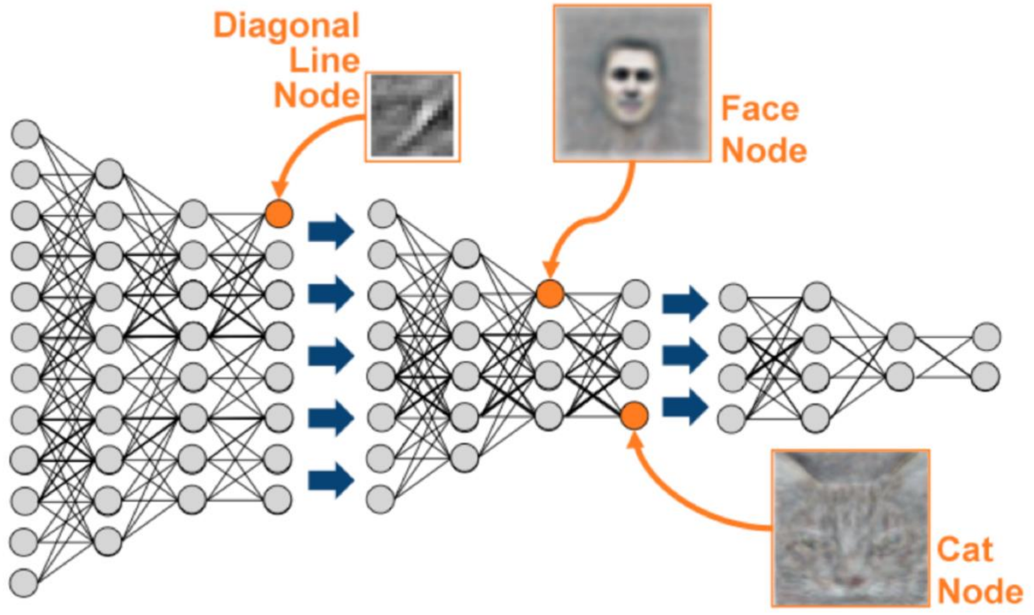


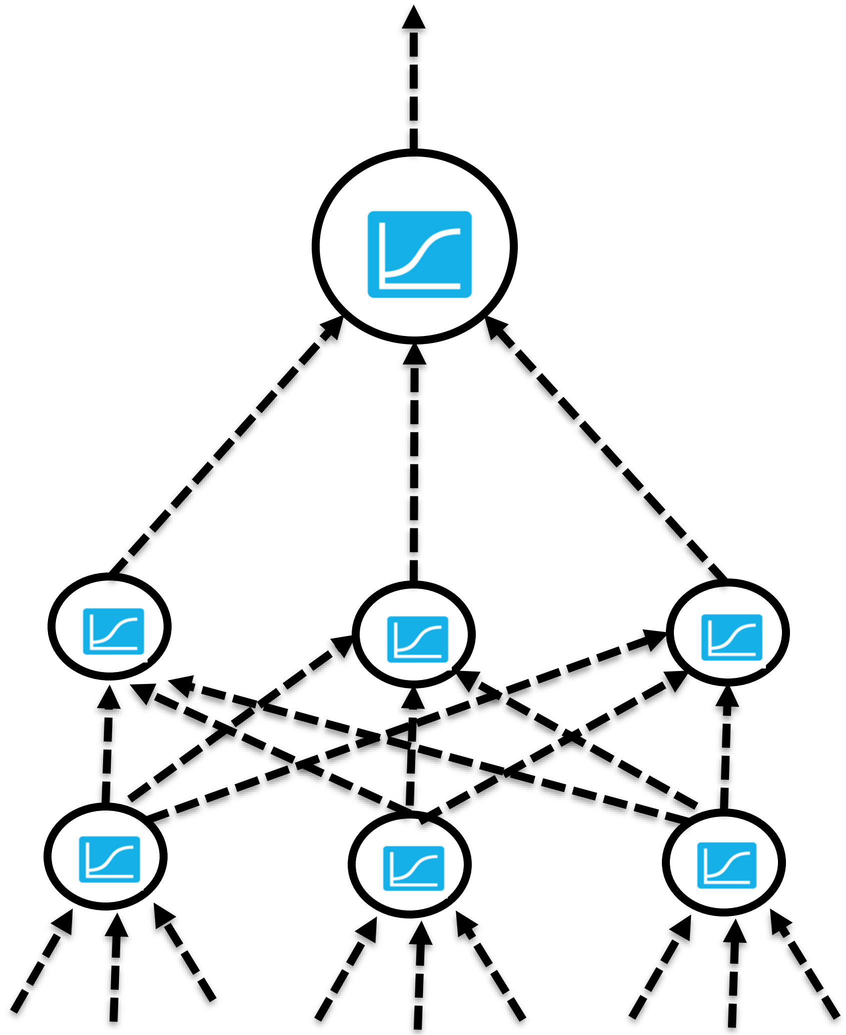
A Layer is a group of nodes organized at the same level



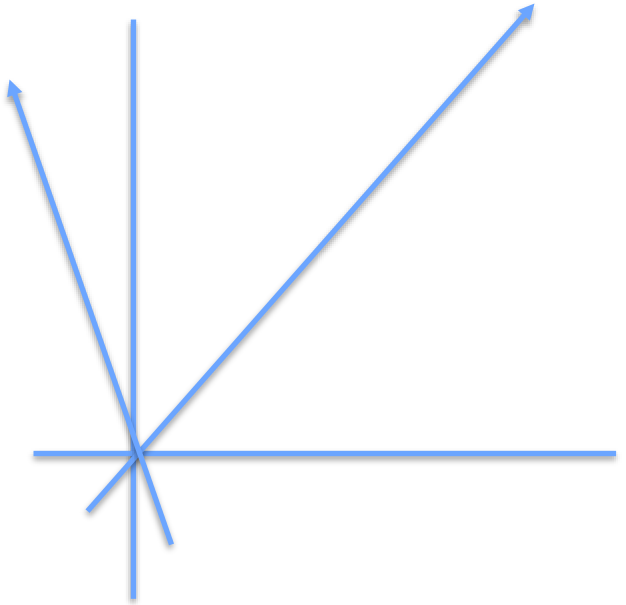
Multiple layers are known as a multiperceptron or deep learning systems

- A network that uses binary classification the output will be two nodes.
- The bias used in the nodes can be a sigmoid operation, parabolic tangent or a Rectify linear unit (ReLU)

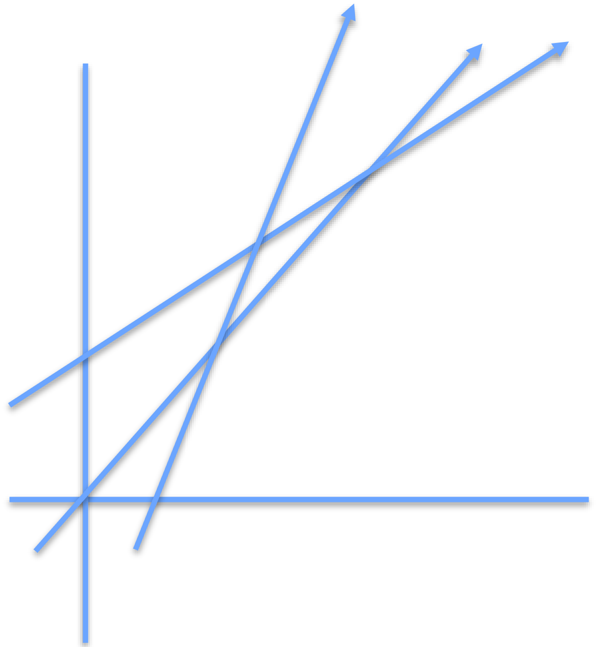




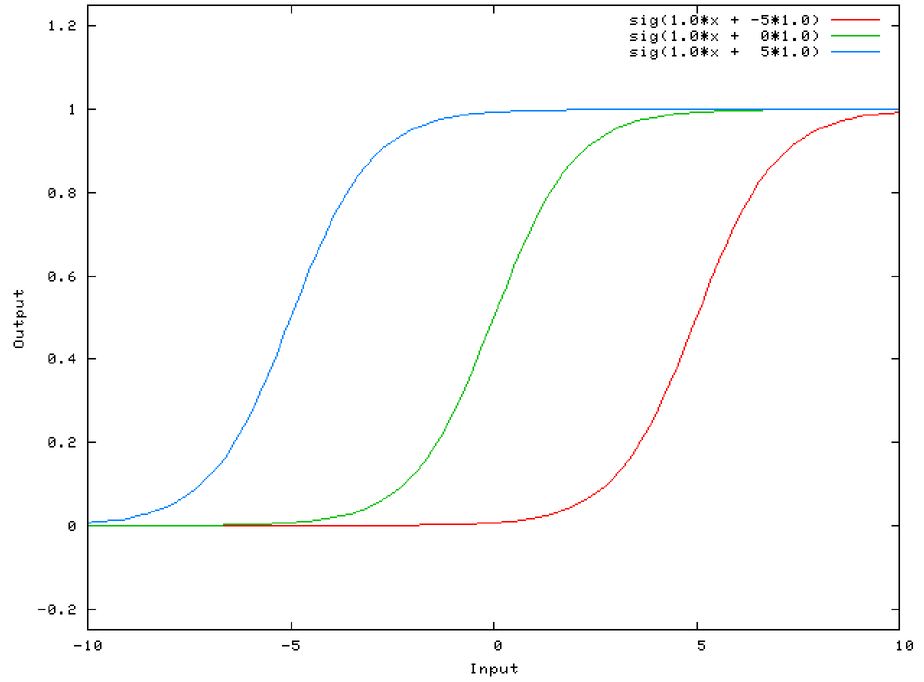
Why weights and why bias?



y= (f)x



y = (f)x + b



Three sigmoid curves—
the same input data, but with different biases

The bias value (b) allows the activations function to be shifted to the left or to the right, to better fit the data; hence, the changes to the weight alter the steepness of the vector or curve. The bias, on the other hand, shifts the entire vector or curve so it fits better.

Let’s talk about:

Learnable Parameters: It learns on it’s own

…and **Hyperparameters**: humans help the machine learn

What are Learnable Parameters?

These are the **things the model learns on its own** during training.

In a CNN, learnable parameters include:

- Weights of filters/kernels in convolutional layers They learn *what patterns* (like edges or curves) to detect
- Bias values that help improve flexibility of the model

Analogy:

Learnable parameters are automatically updated using backpropagation and gradient descent as the model trains.

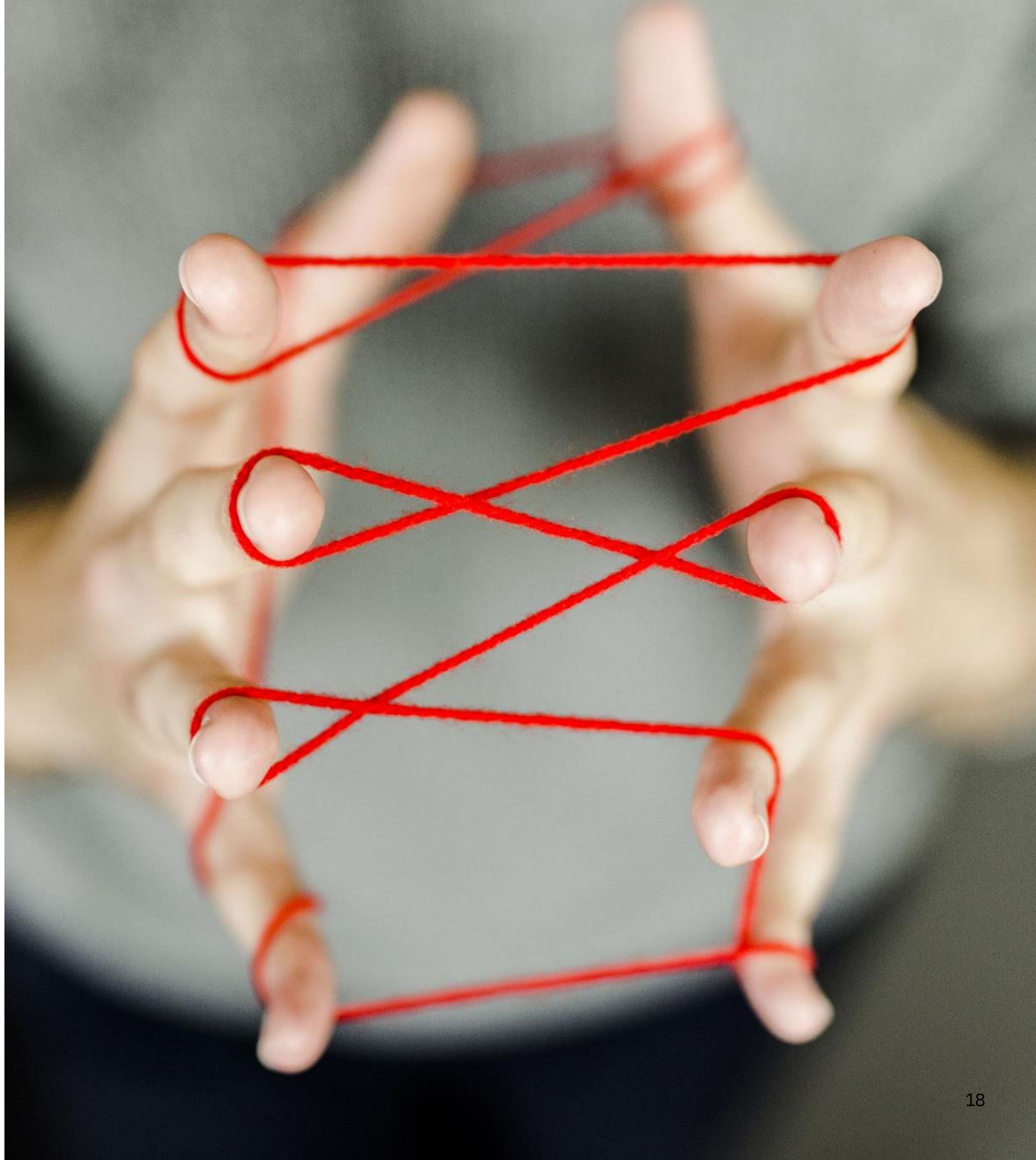
🛠

Analogy:

Imagine you're learning to throw a basketball into a hoop.

The way your arm moves (angle, strength) improves with practice.

These are *learned*—like the model learning filter weights.



What are Hyperparameters?

These are the **settings YOU choose before training** the model.
The model doesn't learn these by itself—you set them manually.

Common CNN hyperparameters:

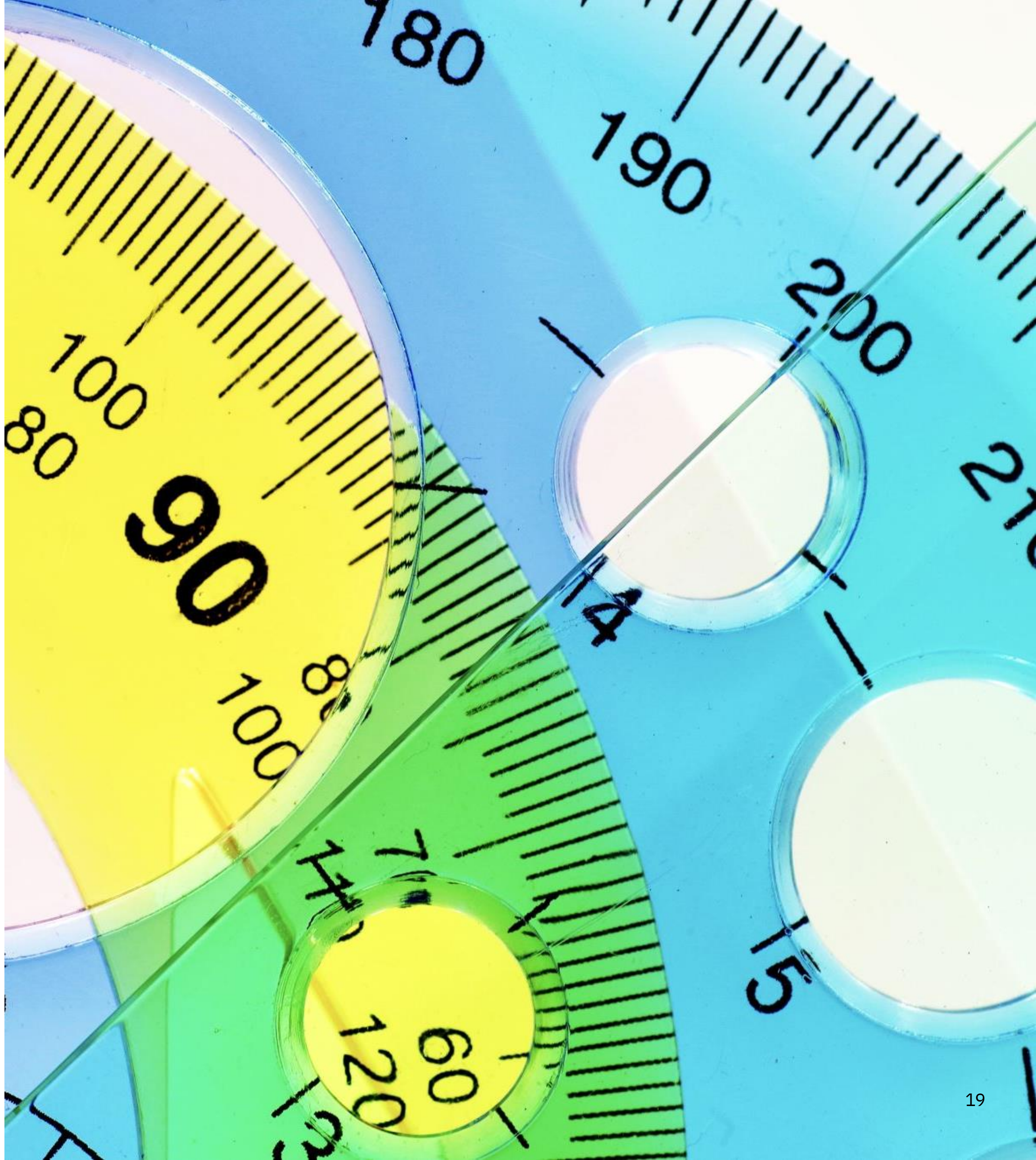
- **Number of filters** (how many features to detect)
- **Filter size** (like 3×3 or 5×5)
- **Learning rate** (how fast the model learns)
- **Batch size** (how many samples to train at once)
- **Number of epochs** (how many times to see the full dataset)
- **Dropout rate** (how much to randomly ignore neurons to avoid overfitting)

Analogy:

In cooking, the ingredients (like spices or oven temp) are **hyperparameters**.

You adjust them before cooking, but the dish "learns" to taste good as it cooks.

If your model is not performing well, tweaking hyperparameters may help!

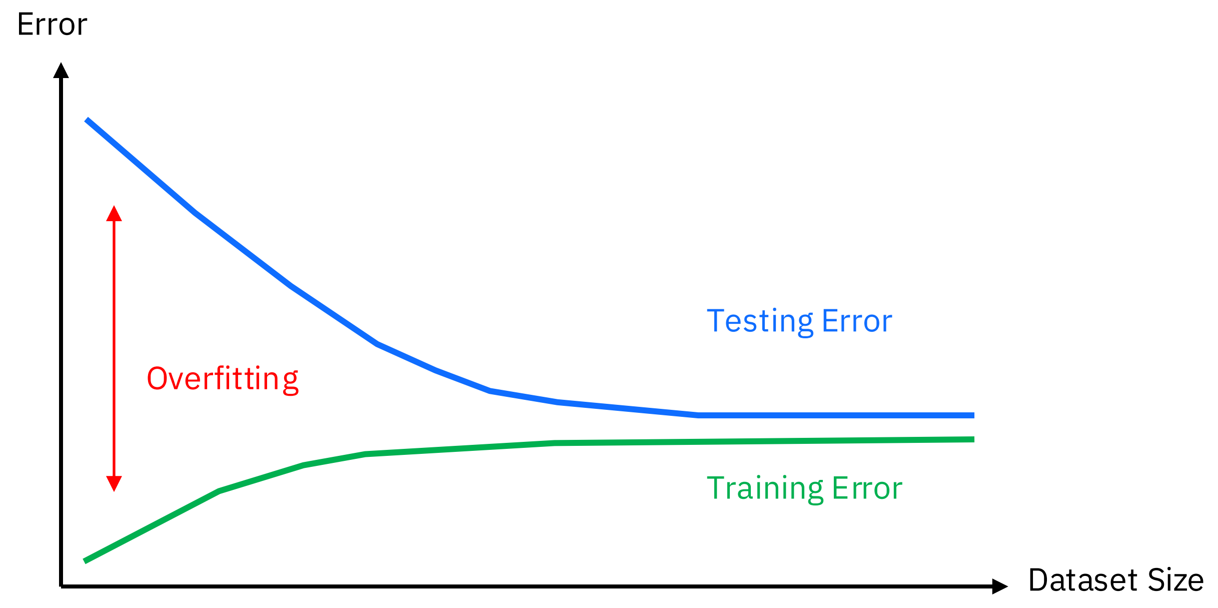


Dangers of Overfitting Data

The biggest risk is overfitting

- A model that works well on training data but performs badly on new data

Trick is to select a model without knowing the data it will be applied to



Example of overfitting

We want to predict if a student with a GPA of 3.3 will be admitted to Saint Peter's University

Assume we train a model from a dataset of 5,000 students and their outcomes.

Next, we try the model out on the original dataset, and it predicts outcomes with 99% accuracy… wow!

But now comes the bad news.

When we run the model on a new ("unseen") dataset of student admissions, we only get 50% accuracy… uh-oh!

Our model doesn't *generalize* well from our training data to unseen data.

This is known as overfitting, and it's a common problem in machine learning and data science.

Details of a CNN

Input Layer: This layer represents the input image itself. For color images, it typically has three channels (red, green, and blue) representing the pixel values. You can visualize this as a 3D matrix where the height and width correspond to the image dimensions, and the depth corresponds to the color channels.

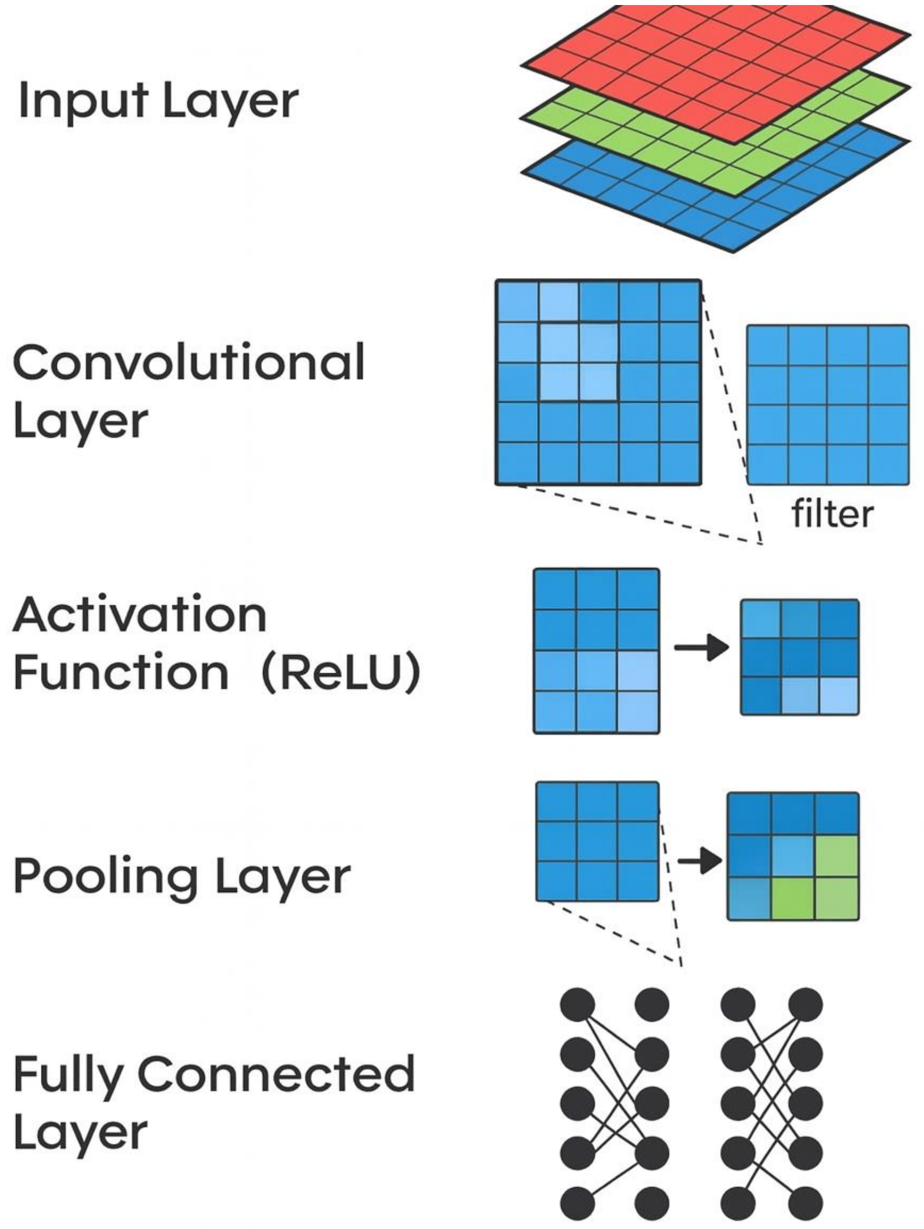
Convolutional Layer: This is the core of a CNN, responsible for feature extraction. It involves applying a set of learnable filters (also known as kernels) to the input image. Each filter is a small matrix of numbers that slides across the image, performing an element-wise multiplication with the part of the image it covers. This process, called convolution, extracts features like edges, textures, or shapes

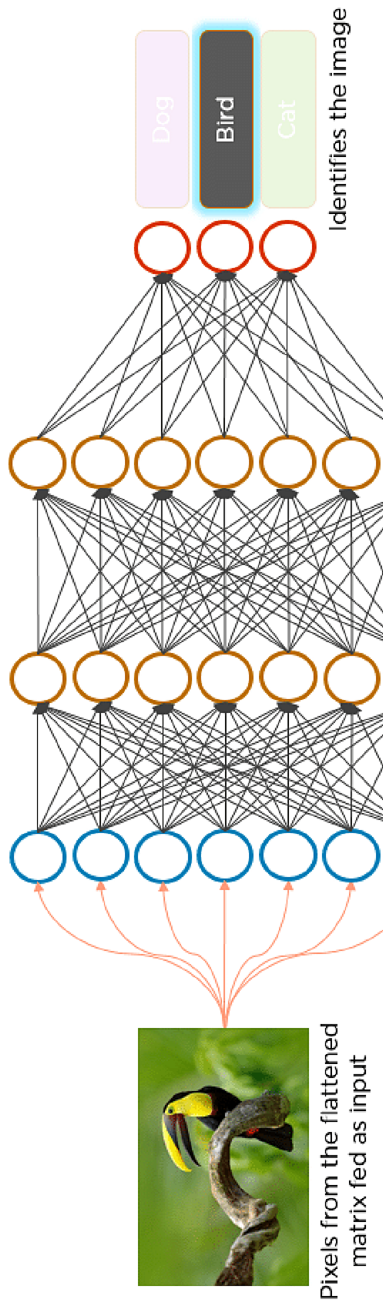
Activation Function (e.g., ReLU): After a convolutional operation, an activation function, commonly ReLU (Rectified Linear Unit), is applied to introduce non-linearity into the network. ReLU sets negative values to zero while keeping positive values unchanged, effectively removing irrelevant features from the activation maps and improving training speed. You can visualize this as a filter "lighting up" only when certain features are strongly present.

Pooling Layer: This layer helps reduce the spatial dimensions of the feature maps, reducing the number of parameters and computations, and making the network more robust to variations in object position. Max pooling, for example, takes the maximum value from a small region of the feature map, effectively summarizing the most prominent features in that area. Imagine taking the brightest spots in different sections of a feature map to create a smaller, more focused representation.

Fully Connected Layer: After several convolutional and pooling layers, the flattened feature maps are fed into fully connected layers. These layers are similar to traditional neural networks, where every node in the layer connects to every node in the previous layer. They learn to combine the extracted features to perform the final classification or prediction task.

Output Layer (e.g., Softmax): The final layer of a CNN for classification tasks typically uses a Softmax activation function. Softmax converts the outputs of the fully connected layer into a probability distribution over the possible classes, where the sum of probabilities for all classes equals 1. The class with the highest probability is then chosen as the predicted class.





Advantages of CNNs	Drawbacks
Learns features automatically	Needs a lot of labeled data
Works great with image data	Training takes time and computing power
Reduces number of parameters	Hard to understand what it's learning
Handles large images efficiently	May get confused by rotated or unusual images