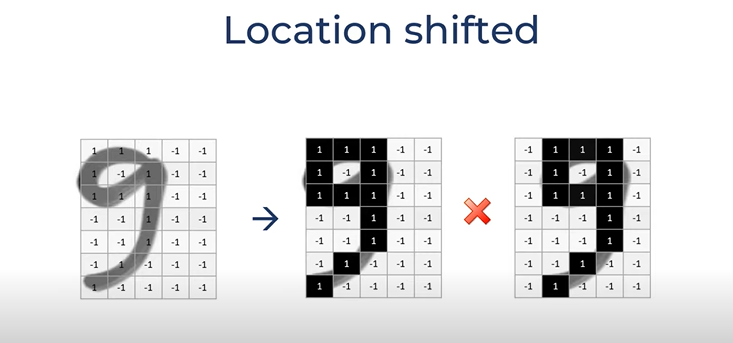
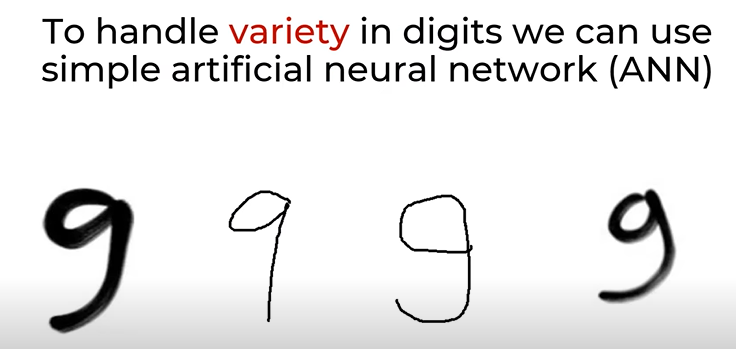
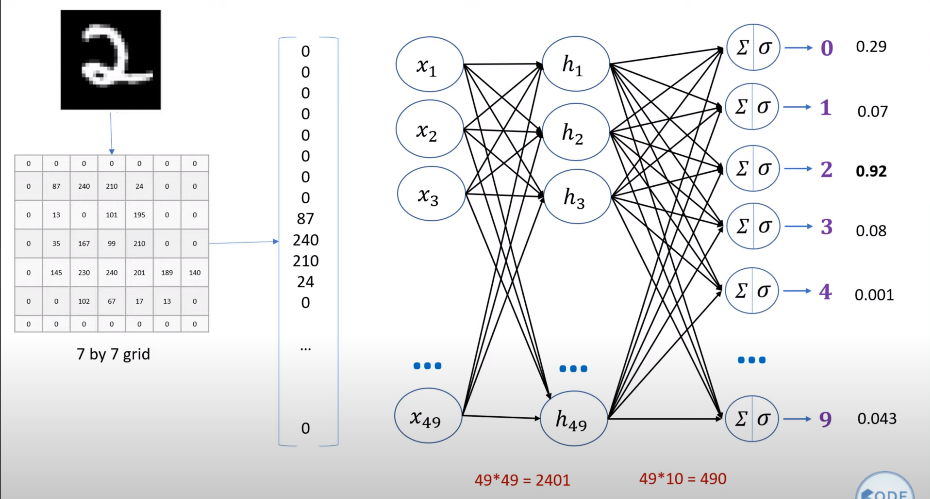
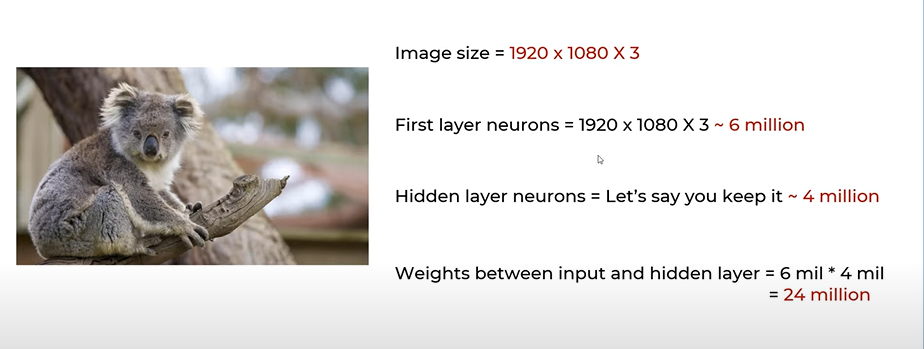
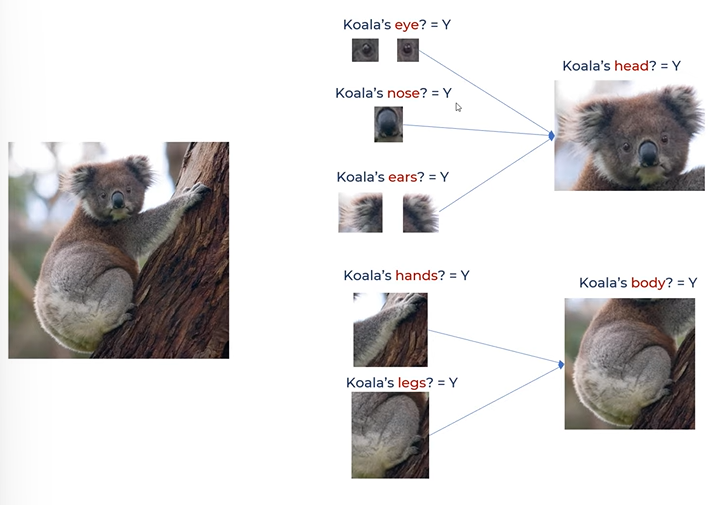
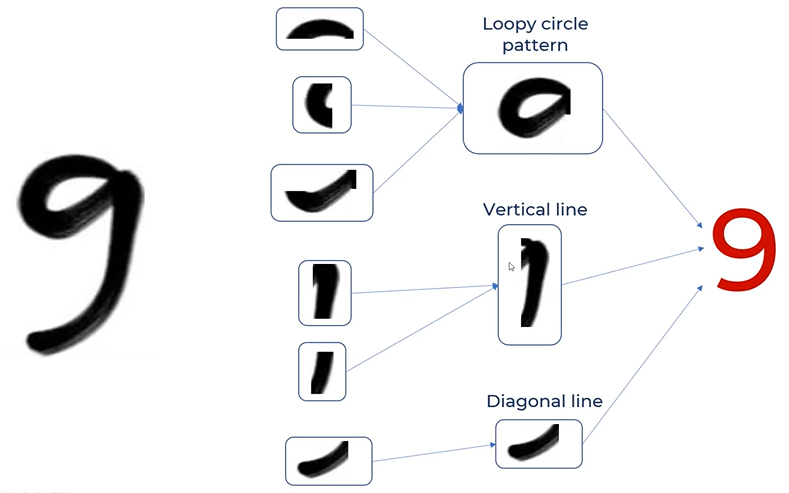
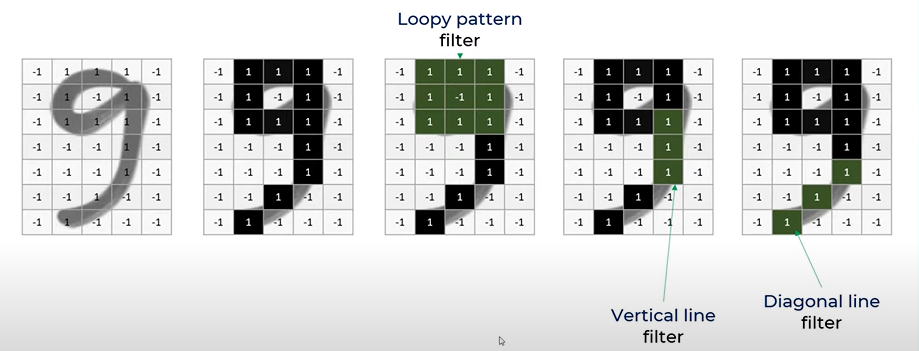
Issue without CNN:

* Machine cannot recognize the handwritten numbers if the image is shifted and the matrix representation of the number changes
* 
* 
* How ANN works?
  + 
  + 
* Disadvantages of using ANN for image classification:
  + Too much computation
  + Treats local pixels same as pixels far apart
  + Sensitive to the location of an object in an image

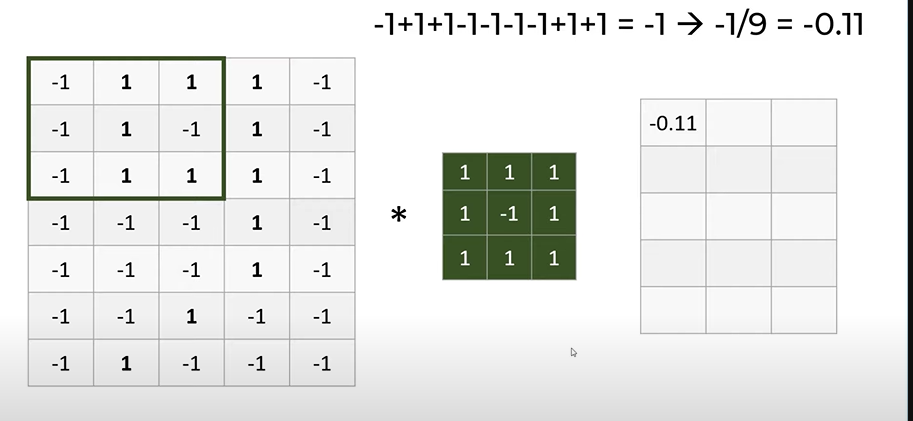
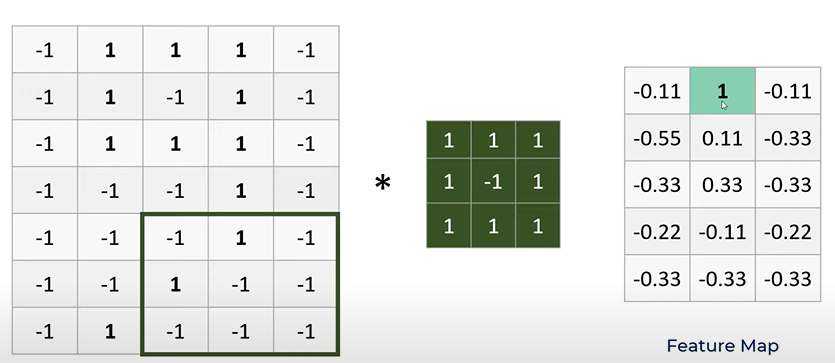
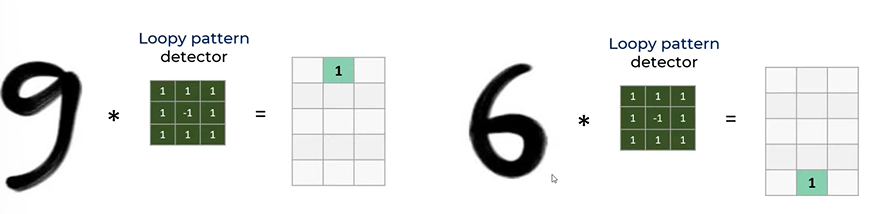
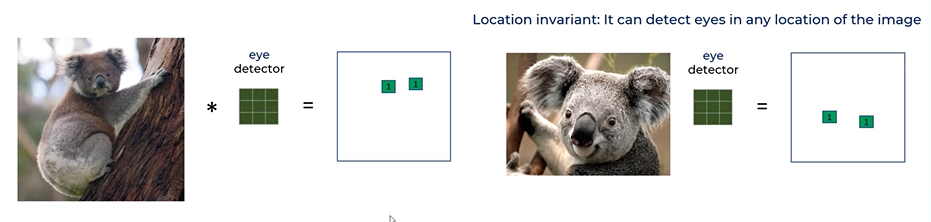
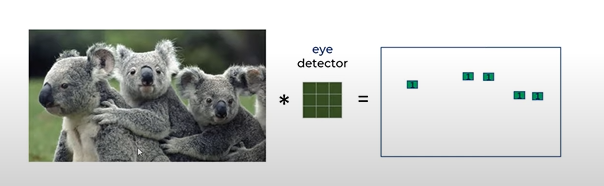
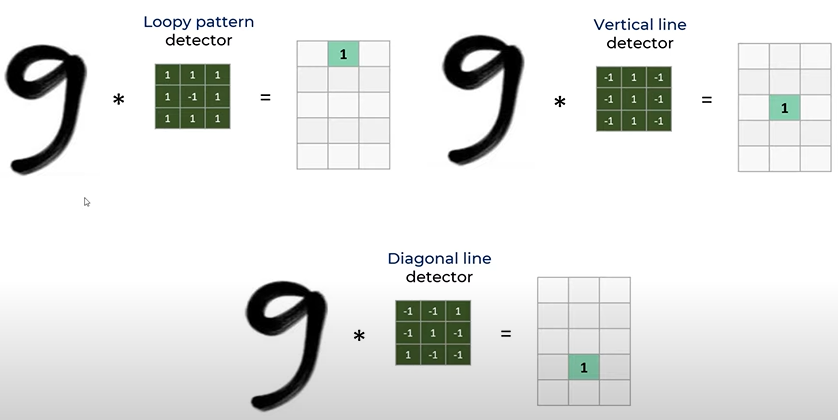
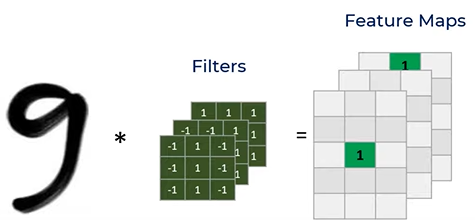
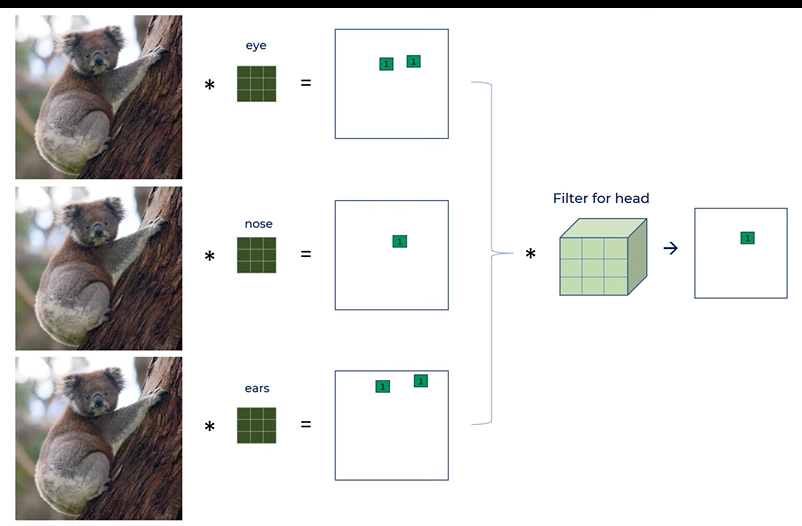
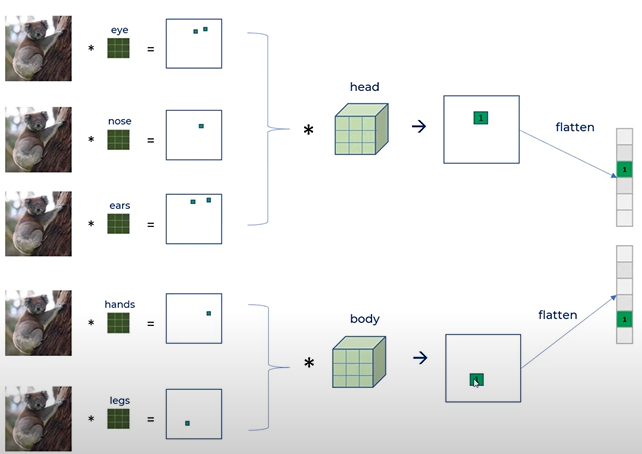
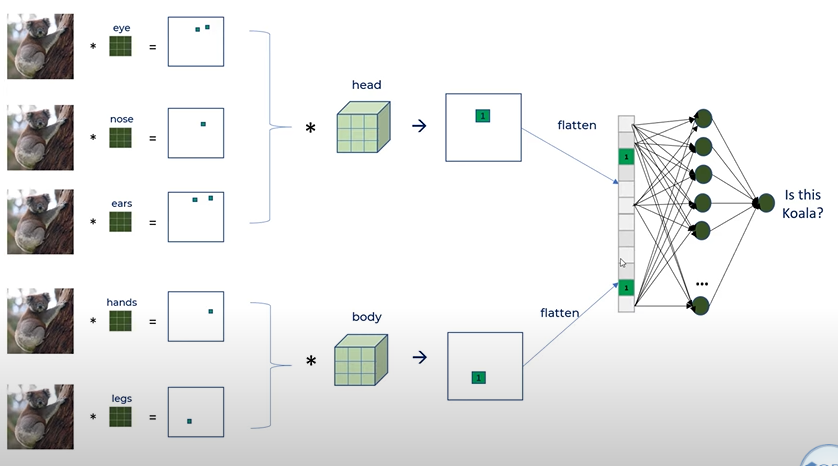
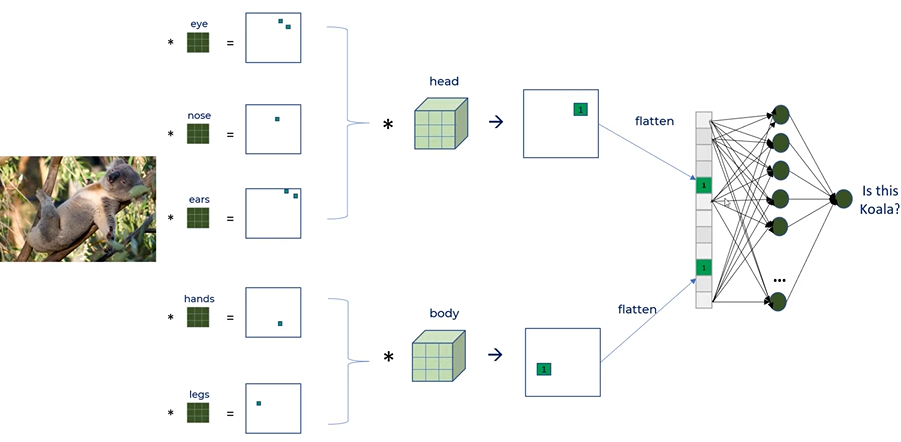
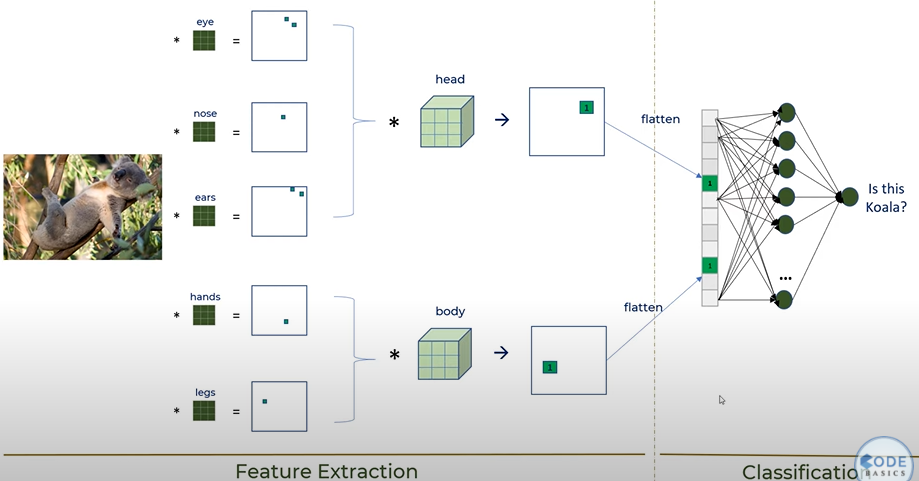
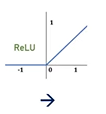
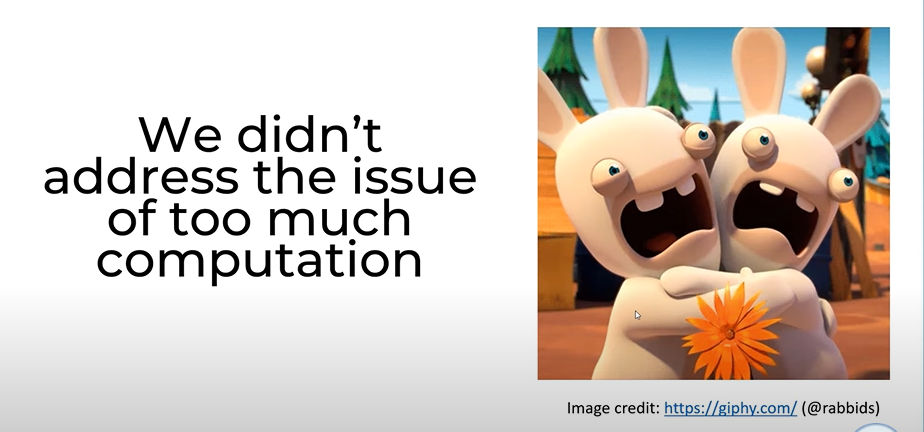
How do humans recognize images even if the image is not centralized?

* Let’s take the example of Koala’s image,
  + First we look at each part of the Koala’s body
  + For example, we look at Koala’s eyes, and then nose, and then ears and so on, and then final
  + Different neurons recognize different parts, and all the neurons that detects the parts of Koala’s face and combined and detects Koala’s face
  + 
  + 

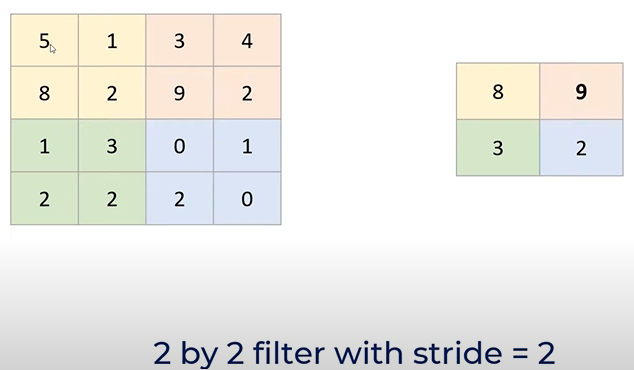
How can we make computers think the same as explained above?

* We use the concept of **filters**
* ****

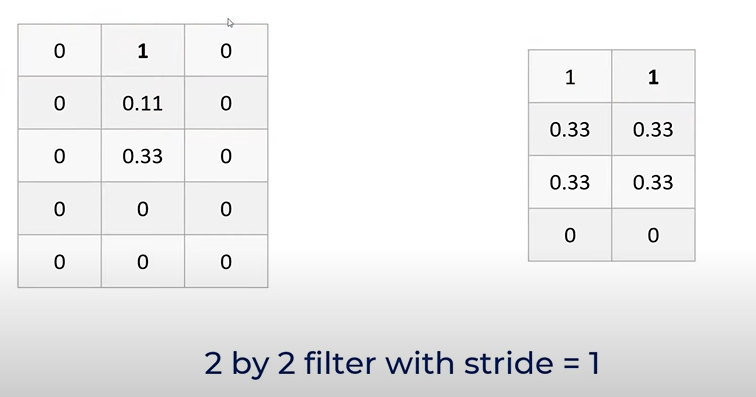
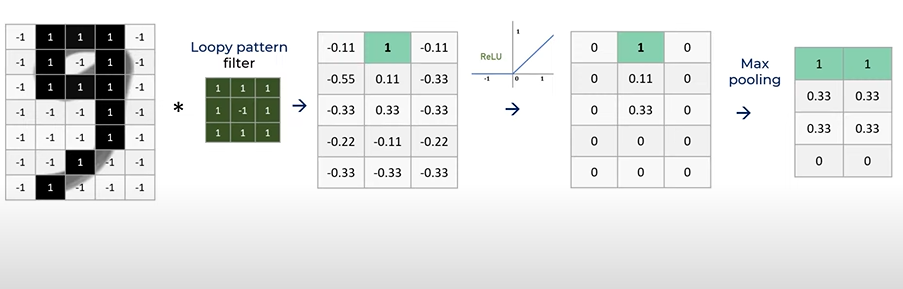
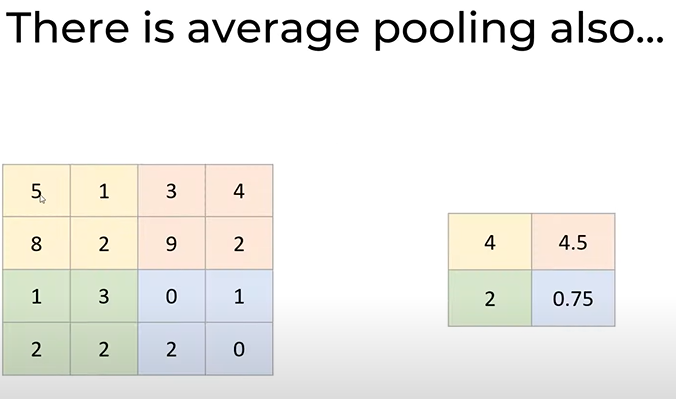
How do filters work?

* Filters are nothing but the feature detectors
* 
* 
* ***During feature mapping in the above image, the feature got activated in the 2 column of 1st row in the above image***
* 
* 
* 
* 
* 
* 3 filters of 9, i.e., loop, straight line and final diagonal line gives 3 feature maps
* 
* 
* The filter need not just be 2D, it can even be 3D in case of Koala’s head
* 
* 
* And then the filters are converted to 1D array using ***flattening*** process
* 
* 
* Here the first part, where we use convolution operation is called ***Feature Extraction*** and the second part where we use dense neural network is called ***Classification***
* NOTE: ReLU Function is an activation function that is used to add non linearity into the model, where all the -ve values and 0s are converted to 0s and the +values are retained without any change
* 
* 

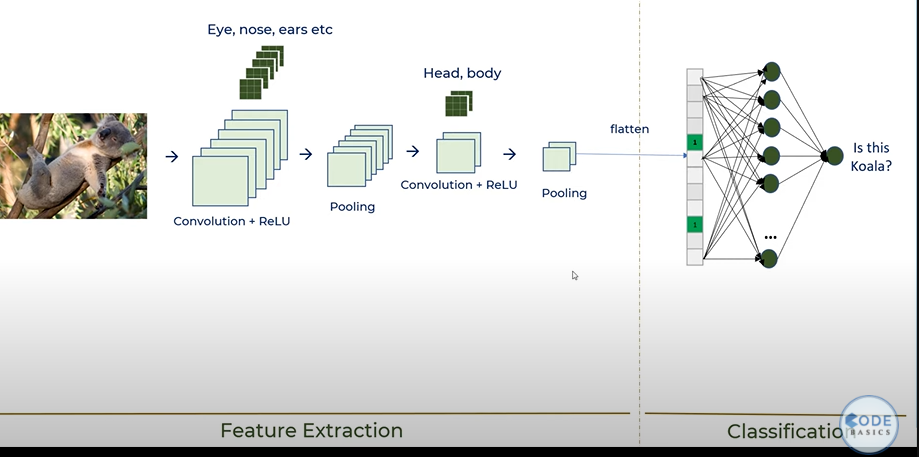
Pooling

* Pooling is a layer that is used in CNN to reduce the size of the image file, in order to overcome the disadvantage of too much computation in ANN
* ***It is also called creating new Feature Map from existing feature map***
* Types of pooling:
  + Max-pooling: pull Max of all numbers in a widow
  + ****
  + **Stride of 2 means, once we are done with first 2 columns, then we go with 2 windows further**
  + **Example of 1 stride**

|  |  |
| --- | --- |

* + ****
  + **Output of 1 stride example is in the above image**
  + ****
  + There is average pooling also
  + ****
  + ***NOTE: But max pooling is generally used***
  + Benefits of pooling:
    - Reduces dimensions & computation
    - Reduce overfitting as there are less parameters, because there are less params
    - Model is tolerrent towards variations, distortions and **noise is less because of ReLU, by capturing only main features**

**Complete CNN process:**

****

**Benefits of Convolution:**

* Connections sparsity reduces overfitting
  + Connections sparsity means not every node is connected to every other node like Artificial Neural network, which is called dense neural network. Instead we have filters which we move around the image. And at a time we are talking about only one feature and we are not affecting the whole image
* Conv + pooling gives location invariant feature detection
* Parameter sharing
  + Which is when learn the parameter/feature filter we can use it across entire image

**Benefits of ReLU**

* Introduces non linearity
* Speeds Up training, faster to compute

**Benefits of Pooling**

* Reduces Dimensions and computation
* Reduces overfitting
* Makes the model tolerant towards small distortion and variations

**How about rotation and thickness/scale?**

* CNN by itself doesn’t take care of rotation and scale
* You need to have rotated, scaled samples in training dataset
* If you don’t have such samples then use data augmentation methods to generate new rotated/scaled samples from existing training samples
  + Data Augmentation
    - Make thicker and rotated samples out of existing data

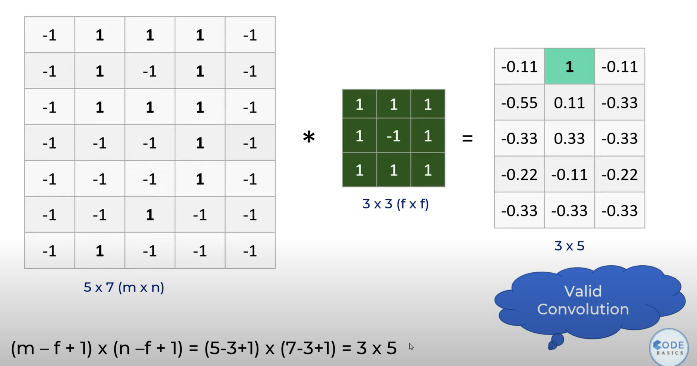
**Magic in CNN**

* It will detect the filters on it’s own, which is part of the training
* Filters will be generated by CNN using back propagation
* And it will also detect the right amount and filters, and the right values in each filter

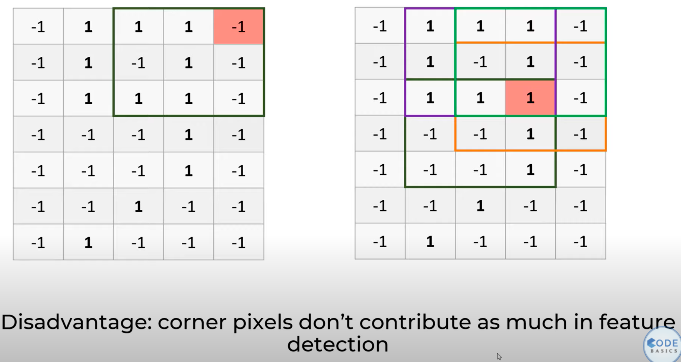
**Hyperparameters in CNN**

* Specify the amount of filters required
* What is the size of each filter?
* But you do not specify the values in filters, CNN will learn on it’s own the values to be there in each filter(blueprint)

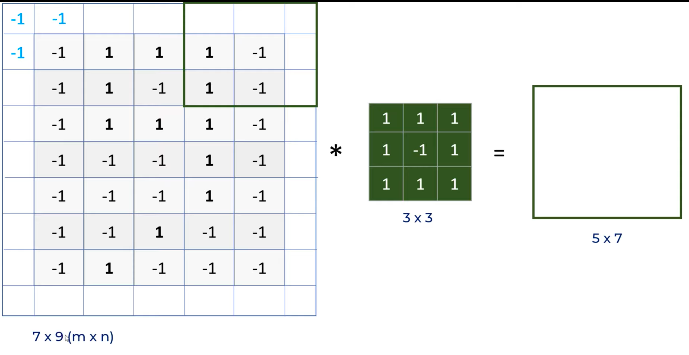
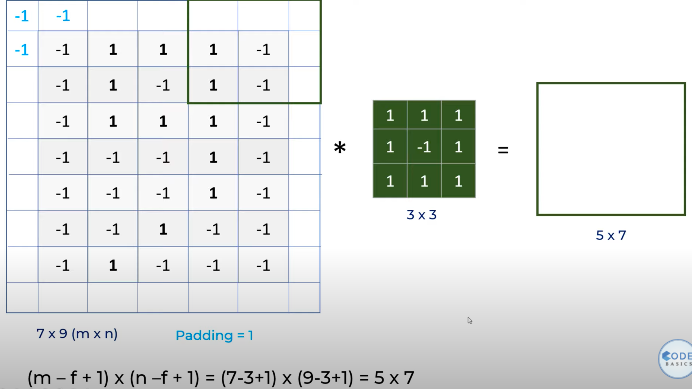
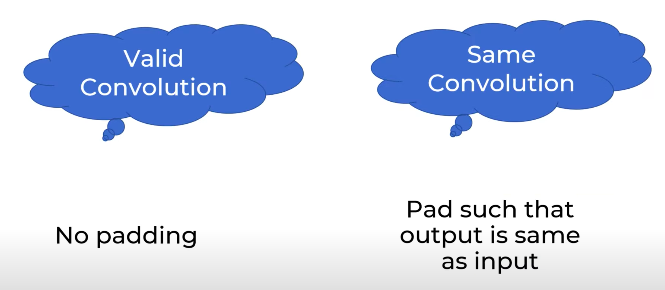
**Valid Convolution**

* 

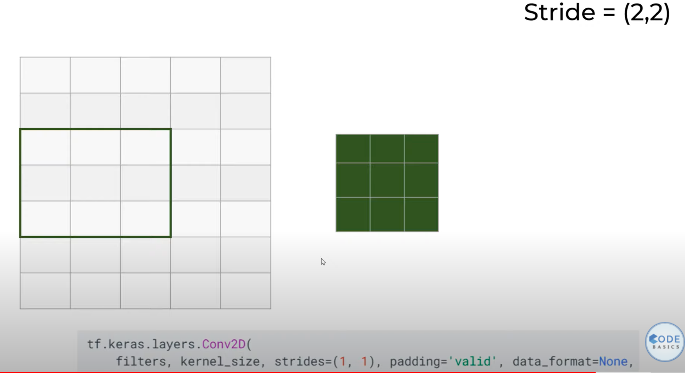
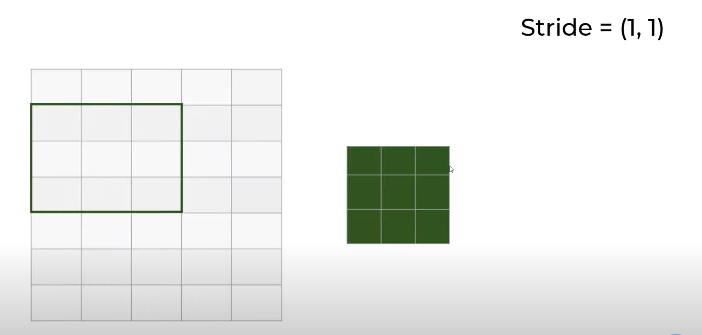
Problem with valid convolution

* 

Padding

* In order to overcome the disadvantage of valid convolution we use padding
* 
* This increases the dimension of the image, but the dimension after feature mapping is 5\*7 which brings back the final image pack. This method preserves the image as it is and also the corner pixels get to play a fair role. And this process is called **same convolution**
* 

**Stride**

* 
* 