**Convolutional Neural Network**

* mostly used for image/speech recognition, outperform ANN in these categories
* ANN is unable to recognize random pixels when unordered/covered :(
* CNN looks at **groups** of pixels instead of individual pixels, more likely to predict the object

Convolutional Layers

* First input layer: our image!
* **Convolutional layer:** composed of neurons which hold information from a section of pixels of the previous layer
* Several of these would allow the algorithm to concentrate on lower-level details in the first layer, then assemble them into higher-level features in the next hidden layer, and so on

Stride

* **Stride:** How many pixels is moved in each step of convolution
* Small strides: high amount of overlap within neurons
* Large strides: less overlap, fewer neurons required in upper layer. But too large -> cannot cover entire image

Padding

* Sometimes we can pick a stride which runs out of room in an image row/column
* Bad as we want each neuron to receive similar receptive feed
* Solutions:
  + **Valid Padding (default)**: Only use valid window locations, ignore extra pixels at border
  + **Same Padding:** Add extra rows/columns at the border of blank pixels, use when borders contain important information

Filter

* How do neurons intake multiple pixels when they only hold one value?
* **Filter:** a matrix with same dimensions of the receptive feed
* We multiply each pixel with its corresponding index in the filter and add it up. This value is passed to the neuron
* Filter values are learnt by the model

Feature Map

* **Feature map:** output of one filter applied to the previous layer
* Many filters may be used, resulting in many feature maps with different features highlighted
* Each convolutional layer will be a bundle of feature maps
* Each convolutional layer will receive information of ALL feature maps of the layer before it

Channels

* In a greyscale information, each pixel has a value which means how white the pixel is
* In a coloured image, each pixel has three values: red, green, and blue
* We can view each coloured image as having 3 **channels**: each layer holding the values of one colour

Pooling layer

* **Pooling layer:** reduces computational load, memory usage, and number of parameters to be estimated
* Difference: pooling layers have no weights. Use max or mean aggregate functions to determine the single value to be passed to the neuron
* Greatly reduces the size of the layer, reducing computation demand, but loses details/extra information