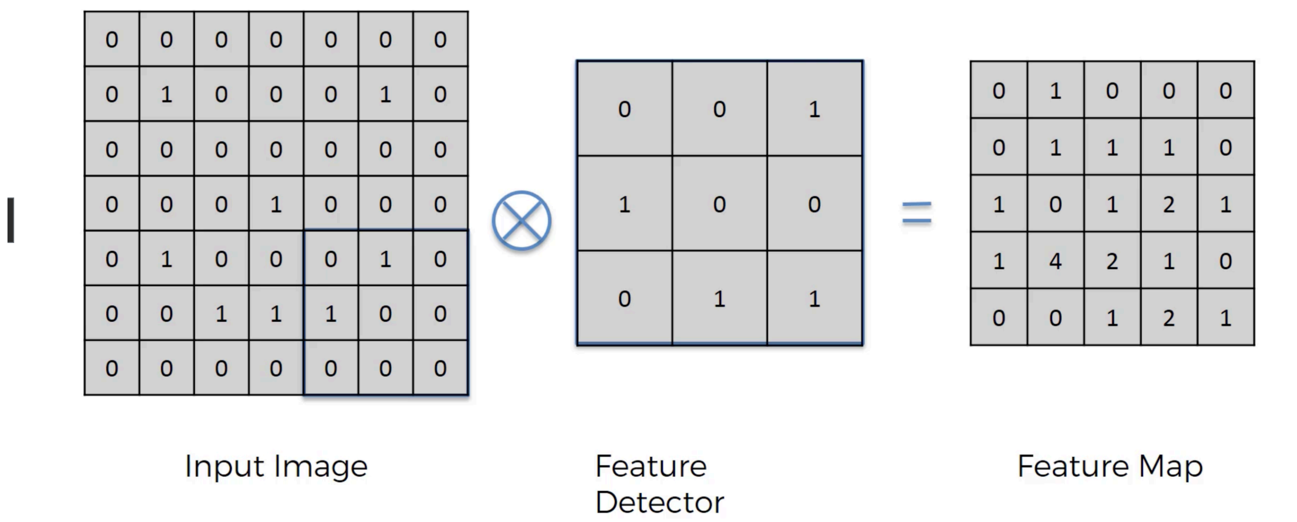
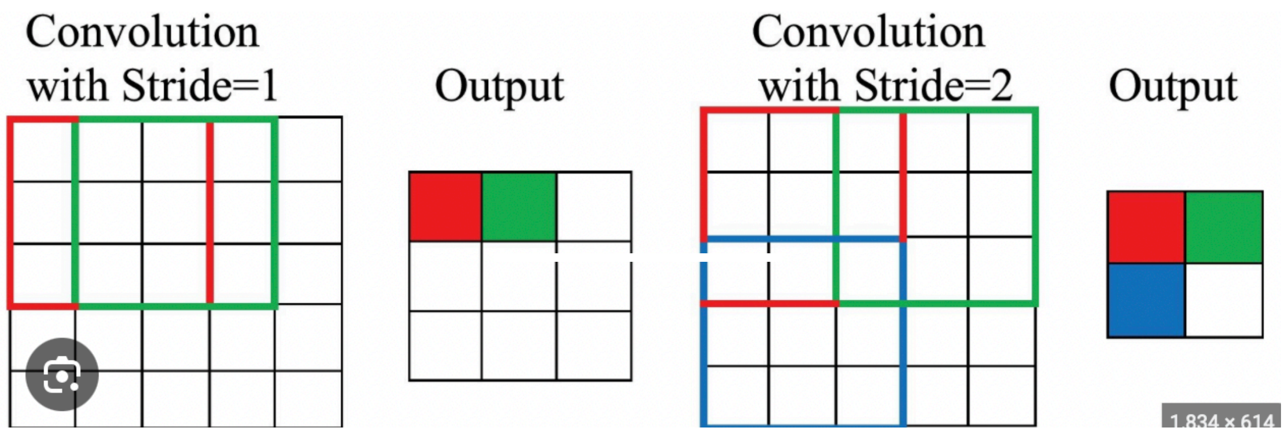
CNN NOTES

Convolutional Neural networks

Step 1 - Convolution



Strides are the boxes that we take in reference while multiplying here we are taking 3x3.

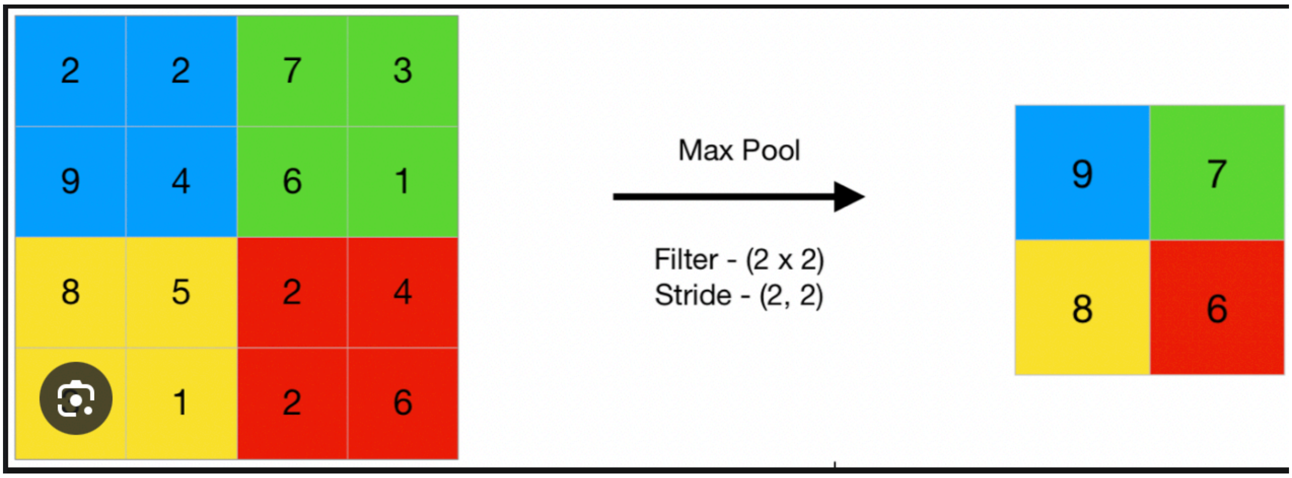


THE FEATURE DETECTORS ARE ALSO CALLED FILTERS CHANGING THE VALUE OF WHOM WE CAN PLAY AROUND WITH IMAGES LIKE BLURRING THEM SHARPENING THEM ETC.

STEP 1B RELU layer
Now multiple feature maps go through RELU (rectifier function) to achieve non linearity in output

STEP 2 Pooling

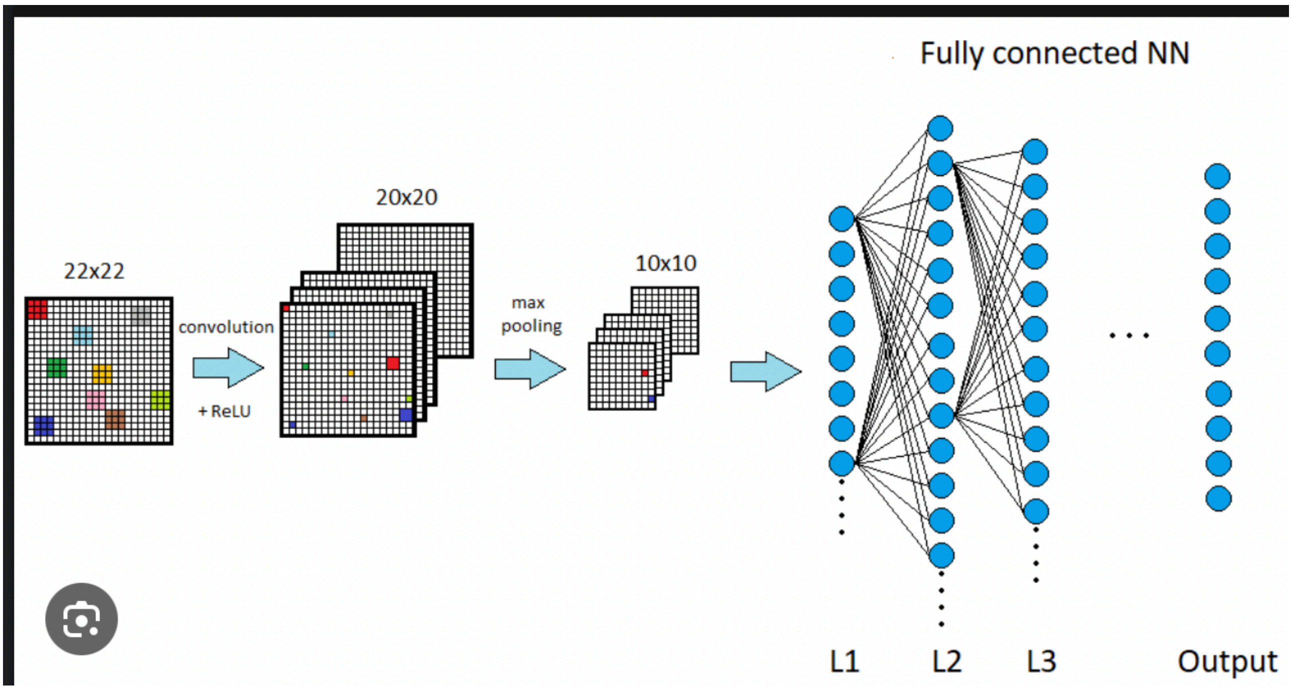
Pooling is generally done to remove unwanted part or just get rid of spatial misappropriations for ex to identify a cheetah we looked for crying spot near eyes so some images may have it a lil tilted so pooling helps to get rid of such problems



Let's compare convolution and pooling in neural networks in simple terms:

1. Convolution:
   - Convolution is like a detective looking closely at a small part of a picture.
   - It's about finding specific patterns or features, like edges, corners, or textures.
   - Imagine the detective has a tiny window that he slides across the picture, studying each area carefully.
   - The goal is to understand what's happening in different parts of the picture.

2. Pooling:
   - Pooling is like the detective taking a step back and looking at the big picture.
   - It's about simplifying and summarising what's been learned.
   - Think of the detective using a special tool to pick the most important information and ignore the less important.



   - The goal is to make the picture smaller and easier to work with while keeping the key details.

In a neural network, convolution helps identify important features in the data, while pooling helps reduce the data's size and complexity, keeping the essential information. Together, they work like a detective team to make sense of the data.*****

STEP 3) Flattening

STEP 4) Full connection

Attaching ANN to the end of flattened layer.

We apply softmax (type of an activation function) and cross entropy at the output layers to get clear judgement about the performance of nn.
Cross-entropy is a commonly used loss function in neural networks, particularly in classification tasks. It measures the dissimilarity between the predicted probability distribution and the actual (true) probability distribution of the target class labels. In simpler terms:

1. **Predicted Probability Distribution**: This is what the neural network thinks is the probability of each class. For example, if you're doing image classification, it might assign a 70% probability to the image being a cat and a 30% probability of it being a dog.

2. **True Probability Distribution**: This is the actual distribution of the class labels. In most cases, it's like a one-hot encoded vector where one class is "hot" (1) and the others are "cold" (0). For instance, if the true label is "cat," it's [1, 0]. If it's "dog," it's [0, 1].

The cross-entropy loss function quantifies how well the predicted probabilities match the true distribution. If the predictions are very close to the true labels, the cross-entropy will be low (indicating a good model). If the predictions are far from the true labels, the cross-entropy will be high (indicating a poor model).

The formula for binary cross-entropy (for two classes, e.g., cat or dog) is:

```
H(y, p) = - [y * log(p) + (1 - y) * log(1 - p)]
```

- `H(y, p)` is the cross-entropy loss.

- `y` is the true label (0 or 1).
- `p` is the predicted probability (between 0 and 1).

For multi-class problems, you extend this concept to more than two classes.

The goal during training is to minimize this cross-entropy loss. In other words, the neural network tries to adjust its parameters (weights and biases) to make the predicted probabilities as close as possible to the true labels.

****

The softmax function is a common activation function used in neural networks, particularly in the output layer of multi-class classification problems. It takes a vector of real numbers as input and transforms it into a probability distribution over multiple classes. In simpler terms, it's like converting raw scores into probabilities.