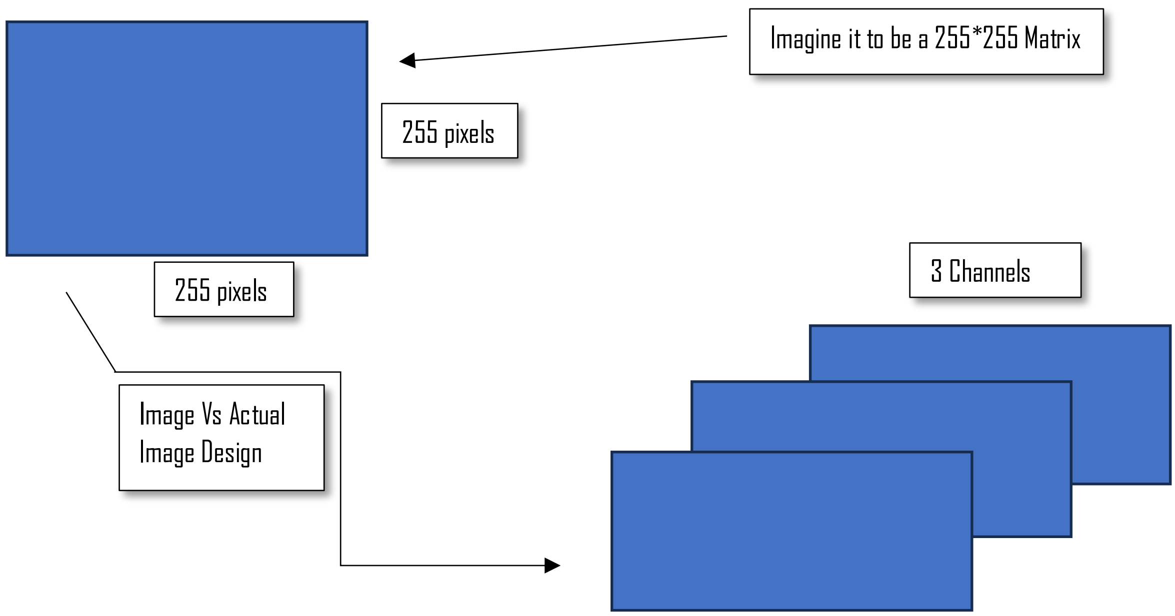
CNN For Beginners : The Magic Behind Image Recognition

Introduction:

Welcome to the wonderful world of Convolutional Neural Networks (CNNs)! This article is designed to be your friendly guide, explaining what CNNs are and how they work in a way that's easy to understand, even if you're new to the world of machine learning.

How does a Machine View an Image????

Now let’s Start! Imagine a beautiful photograph. To us, it's a captivating scene, but to a computer, it's actually a giant grid of numbers! Each tiny square, called a pixel, has a numerical value representing its colour intensity. A typical colour image might have a size of 255 x 255 x 3. Here's what that means:



- **255 x 255:** This refers to the width and height of the image, with each dimension having 255 pixels
- **3:** This represents the three colour channels – Red (R), Green (G), and Blue (B) – that combine to create the full spectrum of colours we see.

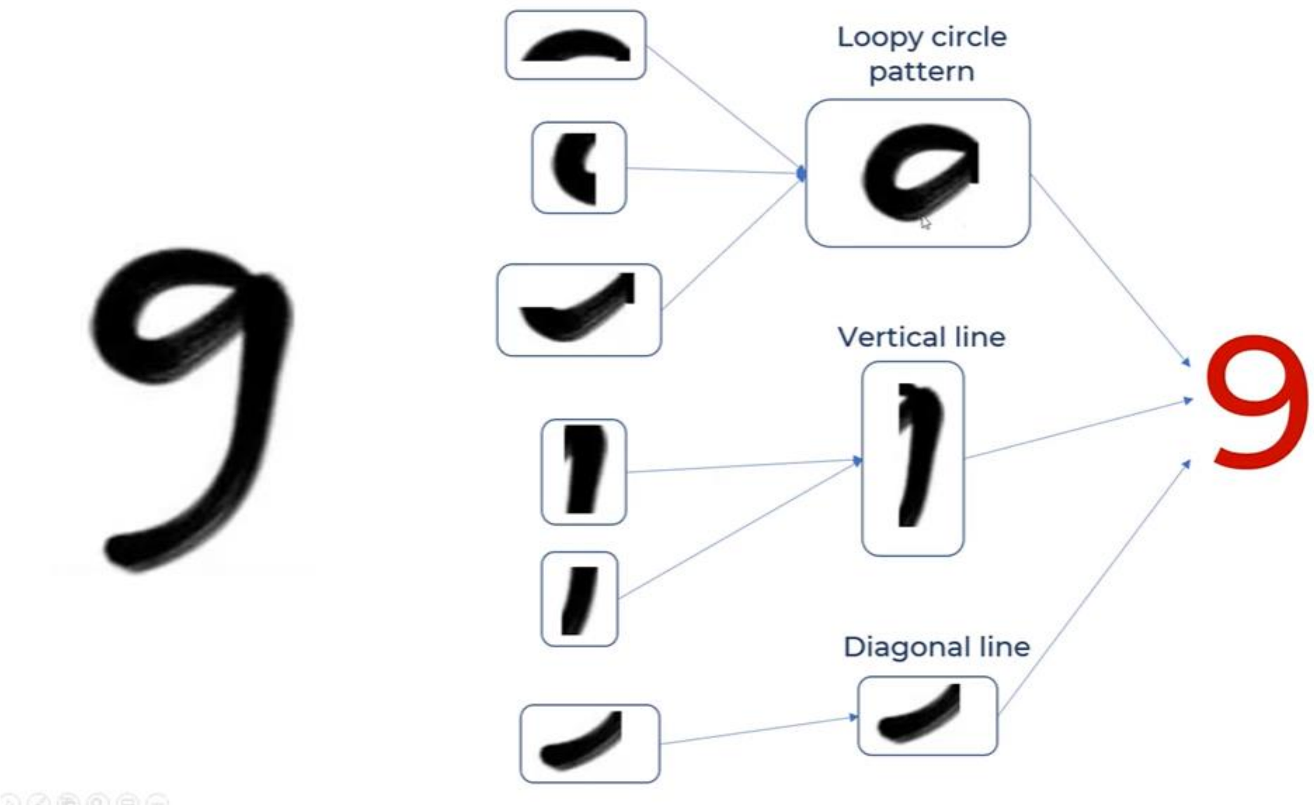
  So, In Our Case the image is essentially three 255 x 255 grids stacked on top of each other, one for each colour channel.

How Does An Machine differentiate Images??

Now, how does a CNN learn to recognize objects in these grids of numbers? It all Comes down to patterns! Let's take a classic example: handwritten digits.

We humans can instantly recognize a number 9, no matter its size or style. But how does a machine achieve this? It dissects the image into patterns, like:

- **Circle pattern**: The top part of the digit 9 often resembles a circle.
- **Vertical line**: The middle stroke of the 9 can be seen as a vertical line.
- **Diagonal line**: The tail of the 9 forms a diagonal line.

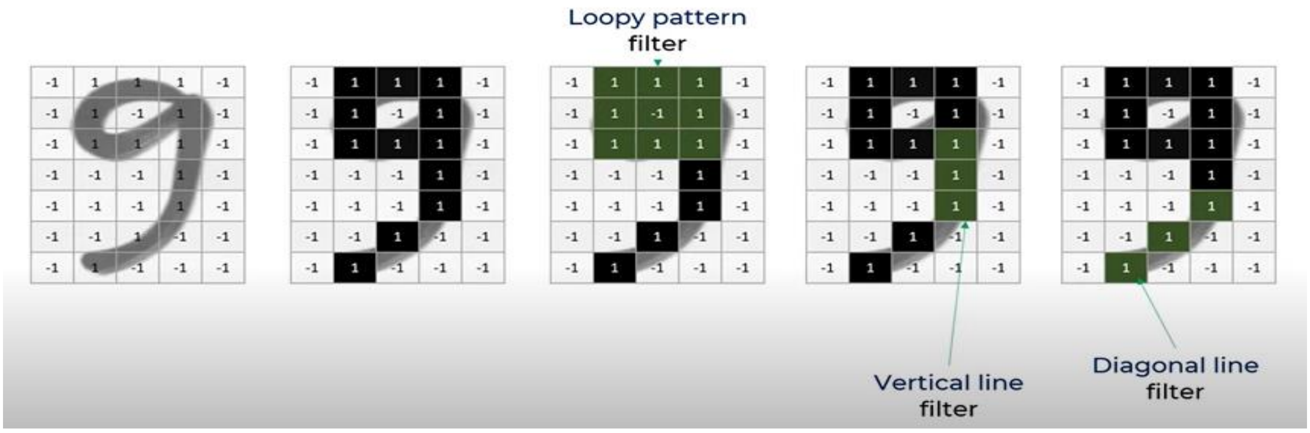


So, How it is actually doing this??? The Answer is "Filters"

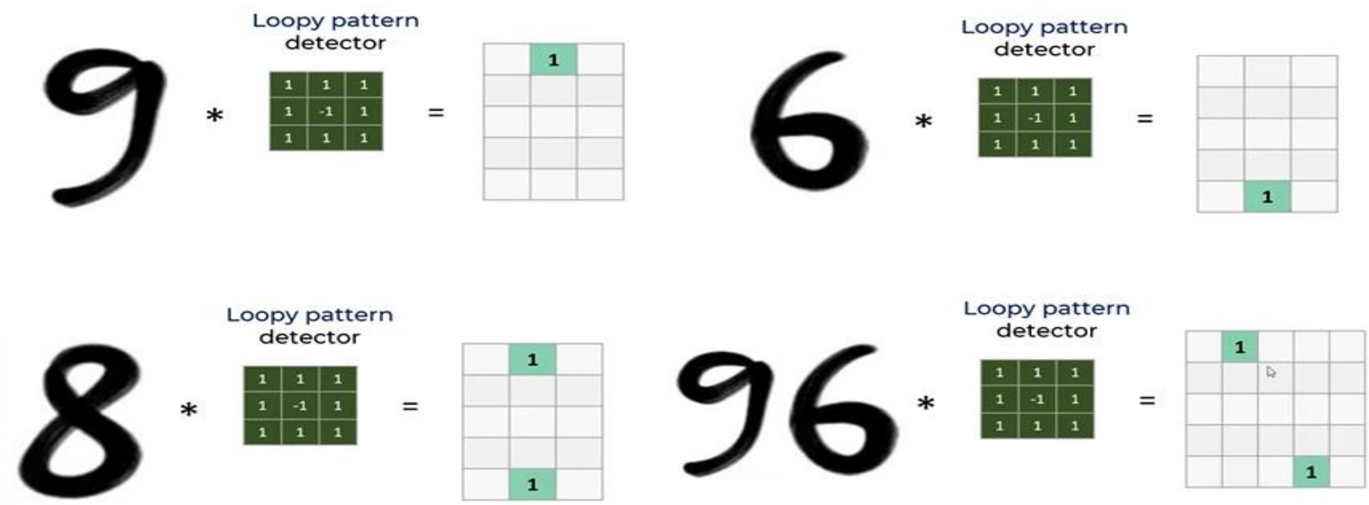
Filters: The Secret Weapon for Pattern Detection

A CNN uses filters, which are essentially small matrices (mini grids) that slide across the image. These filters are like detectives searching for specific patterns. Here's how it works:

- **Matching the Filter:** The filter is placed on a section of the image. The computer multiplies the corresponding pixel values in the filter with the pixel values in the image section.
- **Building an Activation Map:** This process is repeated for all sections of the image, creating a new grid called an activation map. This map shows how well the filter "activated" or matched the patterns in different parts of the image.
- **Multiple Filters for Different Features:** A CNN uses many different filters, each looking for a specific pattern. For example, one filter might look for circles, another for vertical lines, and so on. By combining the activation maps from all the filters, the CNN builds a richer understanding of the image's features.



How it is in the Main Picture:



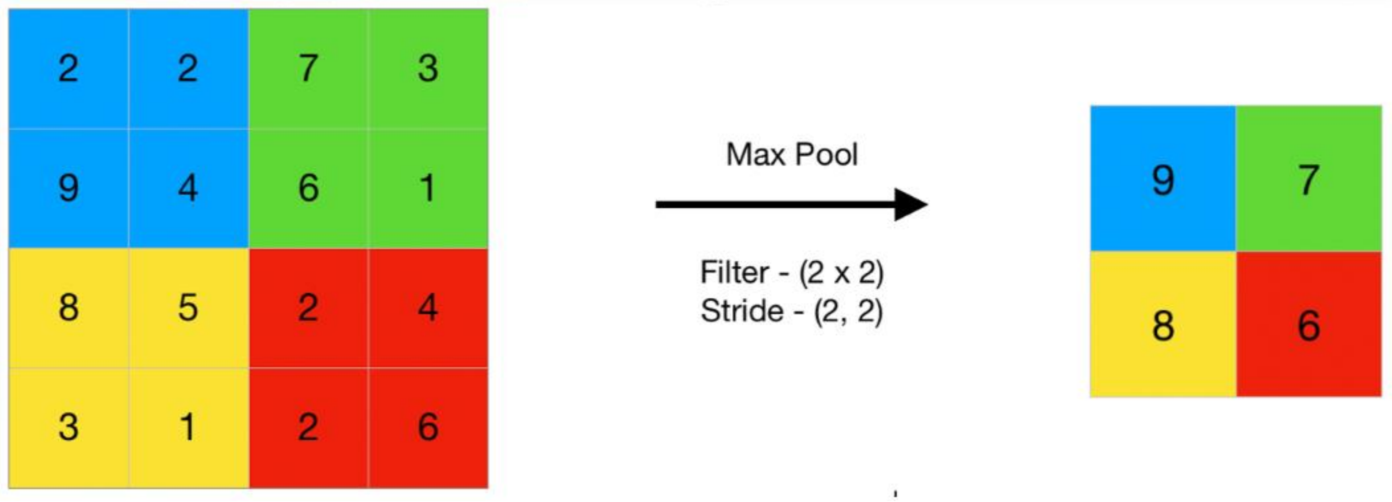
So,In a Nutshell:

“Filters Are Nothing but the Feature Detectors”

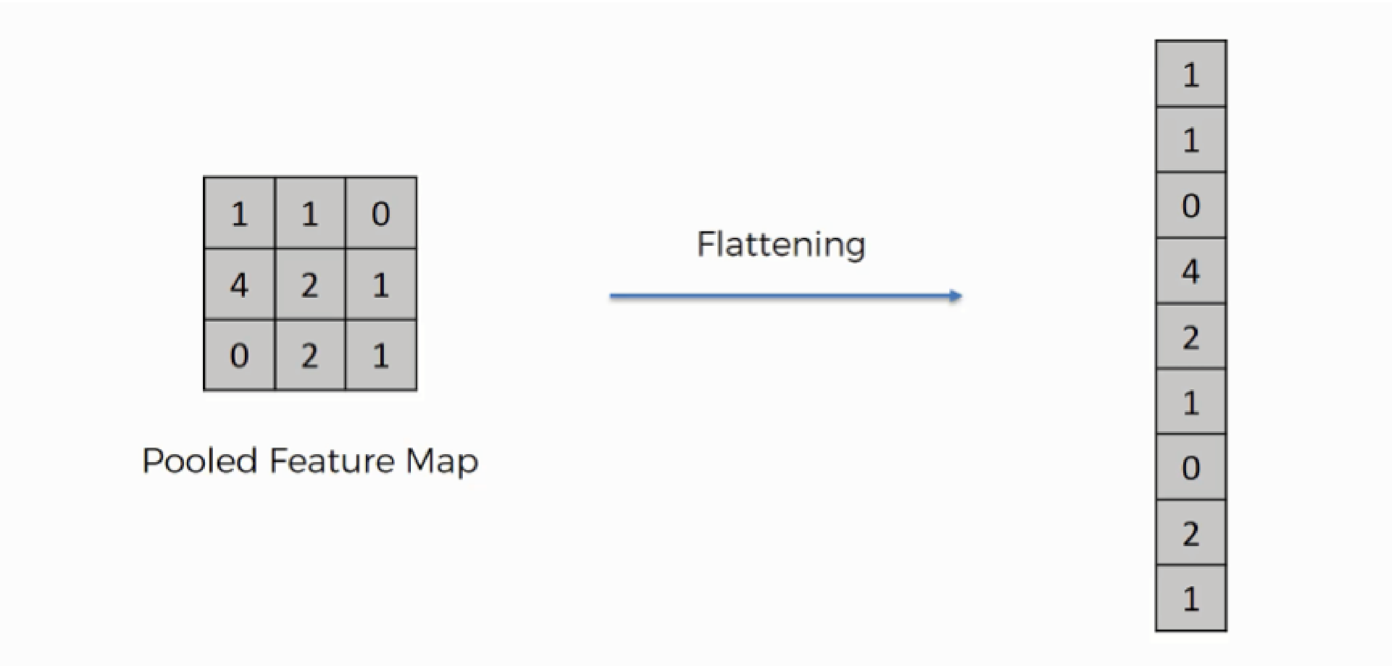
Reducing Complexity:

While filters are powerful, dealing with such large grids of numbers can be computationally expensive. That's where pooling and flattening come in:

Pooling: This technique reduces the size of the activation maps. It takes a small neighbourhood of numbers (e.g., 2x2) and keeps only the maximum value (max pooling) or the average value (average pooling). This helps to capture the most important information while reducing the overall data size.

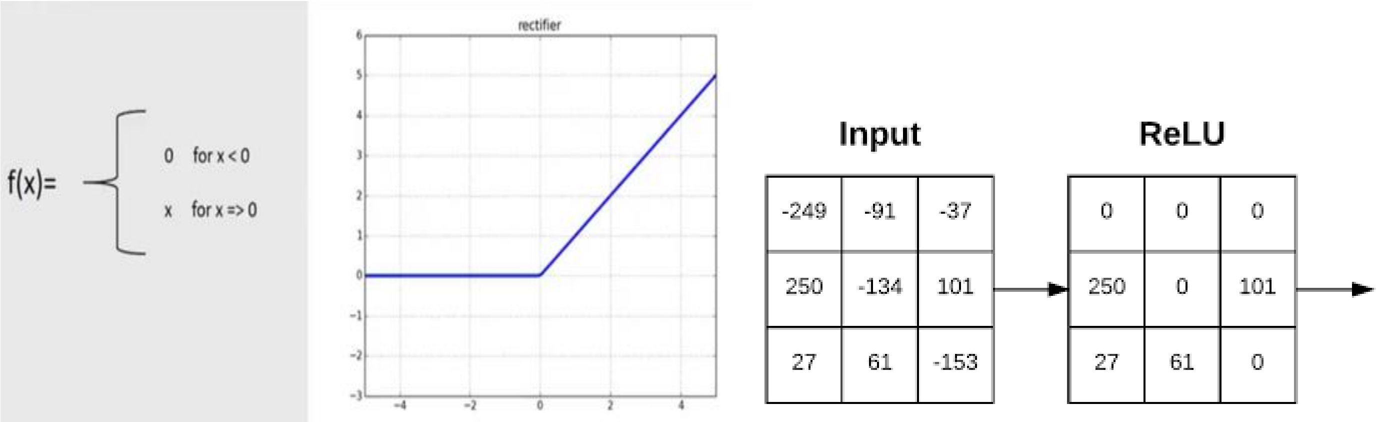


Flattening: After pooling, the remaining activation maps are transformed into a single, long one-dimensional array. This allows the information to be fed into the next stage of the CNN, which is a standard neural network.



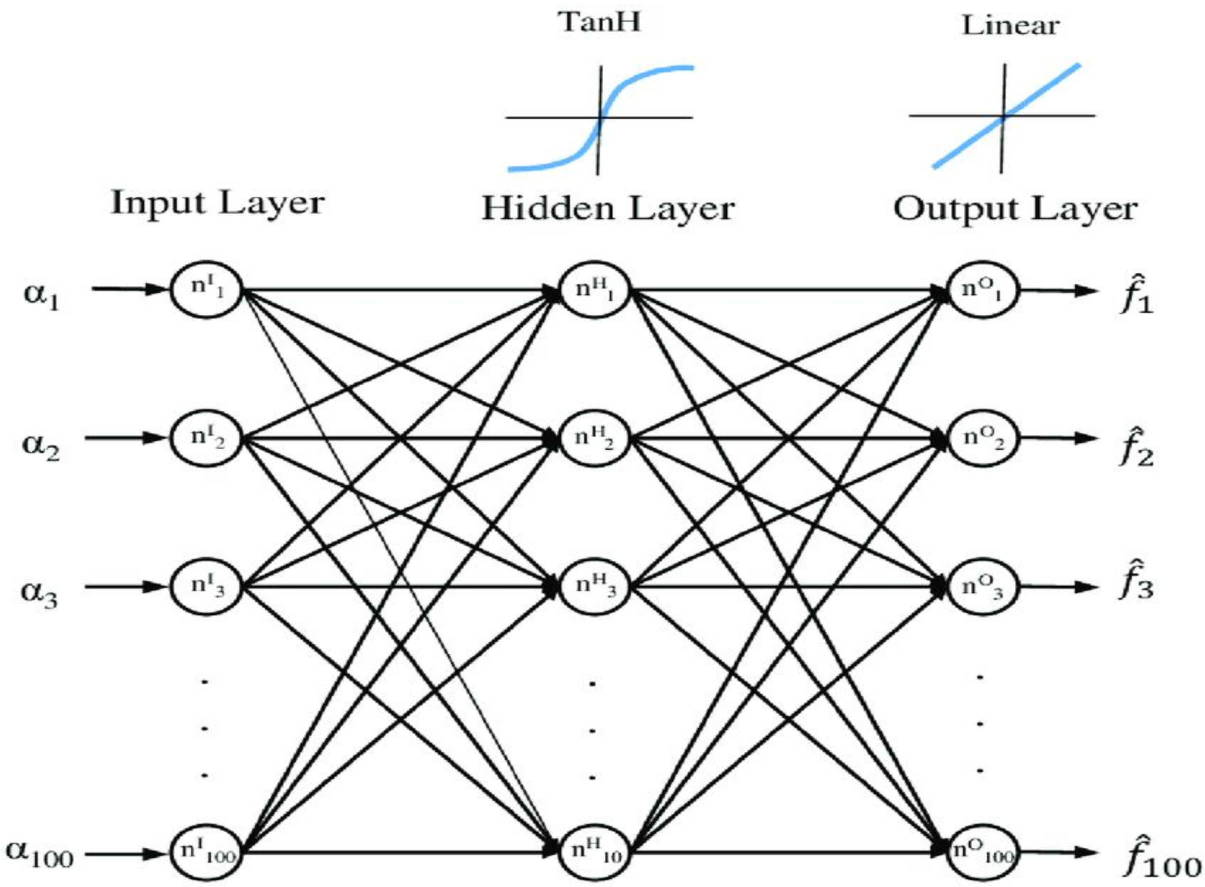
Activation Functions: Adding Non-Linearity

Before feeding the flattened data into the neural network, a layer called the activation function is applied. This layer introduces non-linearity, which is crucial for the network to learn complex relationships between the features. One common activation function is ReLU (Rectified Linear Unit), which simply removes negative values and keeps positive values unchanged.



Hidden Layers and Output: Making the Final Call

The flattened data, now preprocessed by filters, pooling, and activation, is then passed through multiple hidden layers of the neural network. These layers perform complex calculations, learning even more intricate relationships between the features. Finally, depending on the task (e.g., classifying digits 0-9), an output layer with an appropriate activation function (like softmax for multiple classes) determines the most likely outcome.



Putting It All Together:

Now let’s combine all the things we have learnt and put it in a Straight Line!

1.Input Layer: Receives image data, the starting point.

2.Convolutional Layer: Detects features like edges or shapes.

3.Activation Map: Highlights where features are found.

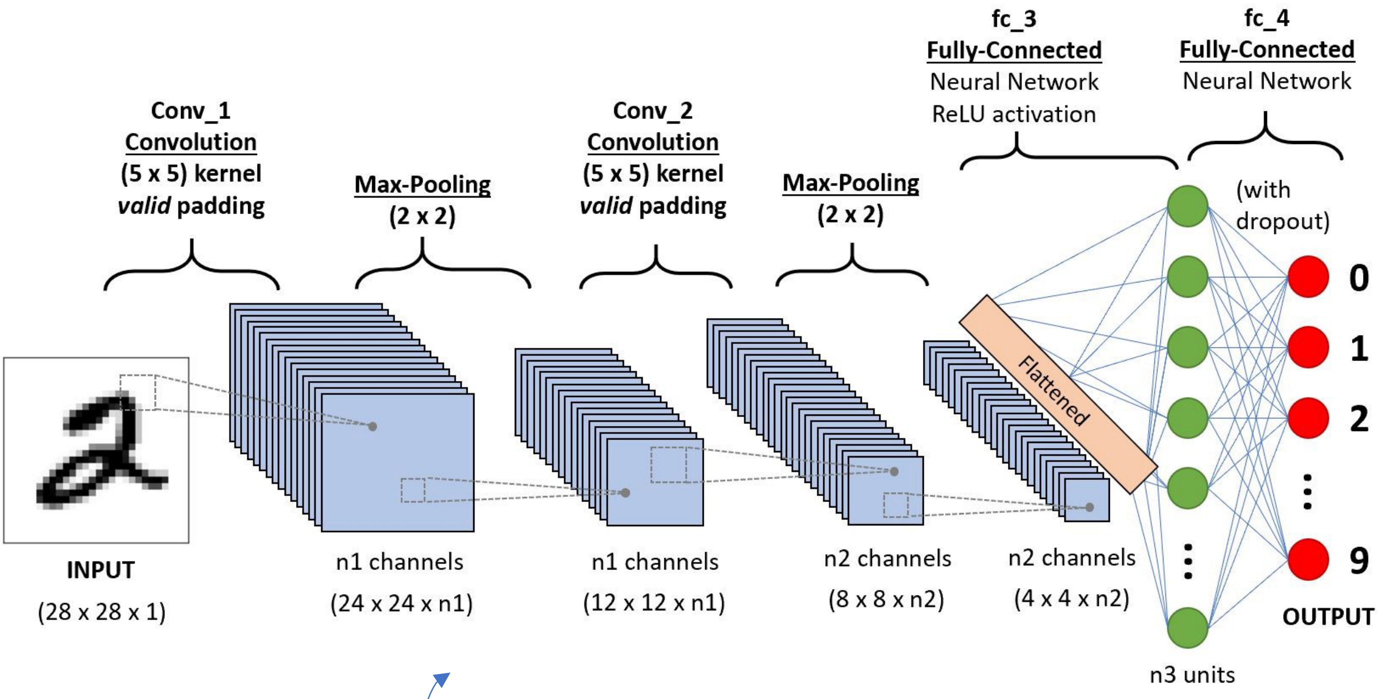
4.Pooling Layer: Shrinks data, keeps vital info.

5.Repeat Convolution and Pooling: Evolves feature detection.

6.Flattening: Makes data ready for deeper analysis.

7.Hidden Layers: Uncover complex image relationships.

8.Output Layer: Decides the image's identity.



This is How the Flow Looks Like

How It Looks in Coding:

```
model.add(Conv2D(32,(3,3),activation='relu', input_shape=(200,200,3)))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(512,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

This is an Basic CNN Code for Binary Classification.

Real-World Applications of CNNs:

- **Image Classification:** Recognizing objects, animals, or faces in images.
- **Object Detection and Localization:** Identifying and drawing bounding boxes around objects within an image.
- **Medical Image Analysis:** Diagnosing diseases, analysing medical scans, and assisting with surgery.
- **Self-Driving Cars:** Recognizing traffic signs, pedestrians, and other vehicles for autonomous navigation.
- **Video Analysis:** Understanding video content to track objects, analyse behaviour, and generate captions.

Conclusion:

This article provides beginners with a simple introduction to Convolutional Neural Networks (CNNs), offering insight into their workings and applications. It breaks down key components and processes, giving readers a foundational understanding of CNNs and their role in analysing visual data.

Article By:

Prithvi Ragavendiran R

(Batch-MIP-ML-07)