Task 3: Based on Research Study

1. Comparison of CNN and YOLO

Both the YOLO (You Only Look Once) family and the CNN (Convolutional Neural Network) family are popular and widely used in the field of computer vision and object detection. However, they differ in their approach to object detection and their specific architectures. Here's a comparison between the two:

YOLO Family:

Object Detection Approach: YOLO follows a single-shot detection approach, where the network directly predicts bounding boxes and class probabilities for multiple objects in a single pass through the network.

Architecture: YOLO consists of a base convolutional neural network followed by several convolutional layers and a final fully connected layer. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell.

Speed: YOLO is known for its real-time object detection capabilities. Since it processes the entire image at once, it is faster compared to many other object detection algorithms.

Accuracy: YOLO sacrifices some accuracy for speed. While it performs well in detecting large objects, it can struggle with small or densely packed objects due to the coarse grid used for predictions.

CNN Family:

Object Detection Approach: CNN-based object detection approaches, such as Faster R-CNN, use a two-stage approach. The first stage generates region proposals, and the second stage classifies the proposed regions and refines the bounding box predictions.

Architecture: CNN architectures used in object detection often consist of a feature extraction backbone (e.g., VGG, ResNet) followed by region proposal networks (RPN) and a classification head. The RPN generates potential object regions, and the classification head predicts class labels and further refines the bounding box predictions.

Accuracy: CNN-based approaches generally achieve higher accuracy compared to YOLO, especially in detecting small or densely packed objects. The two-stage architecture allows for more precise localization and handling of various object sizes.

Speed: CNN-based approaches are relatively slower compared to YOLO due to the two-stage process. However, with optimizations and advancements, their inference speed has improved significantly.

In summary, YOLO trades off some accuracy for speed and is suitable for real-time applications, while CNN-based approaches prioritize accuracy and are often used in scenarios where precise object localization is crucial. The choice between the two depends on the specific requirements of the application and the balance between speed and accuracy needed.

2. Different layers in CNN

The CNN architecture consists of a number of layers. Each layer in the CNN architecture, including its function, is described in detail below.

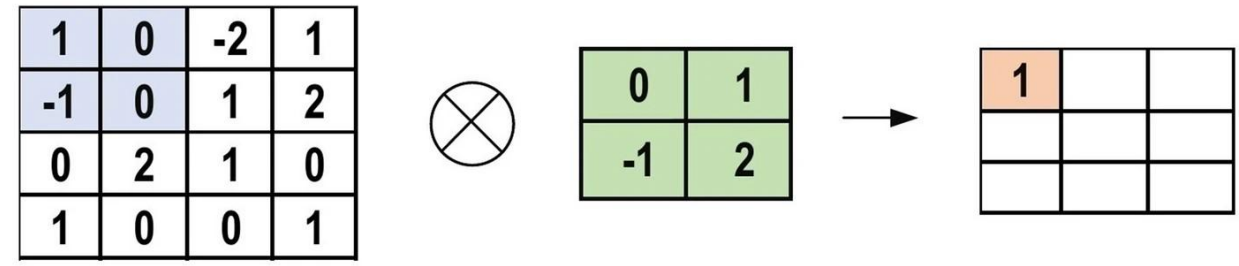
Convolutional Layer:

In CNN architecture, the most significant component is the convolutional layer. It consists of a collection of convolutional filters (so-called kernels). The input image, expressed as N-dimensional metrics, is convolved with these filters to generate the output feature map.

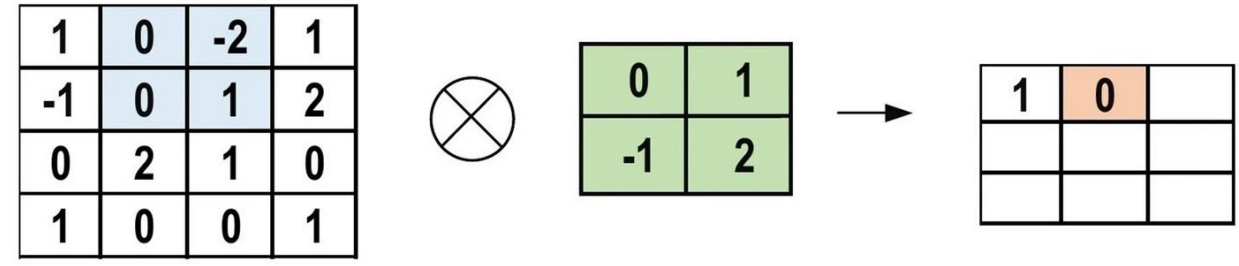
• **Kernel definition:** A grid of discrete numbers or values describes the kernel. Each value is called the kernel weight. Random numbers are assigned to act as the weights of the kernel at the beginning of the CNN training process.

• **Convolutional Operation:** To understand the convolutional operation, let us take an example of a 4 × 4 gray-scale image with a 2 × 2 random weight-initialized kernel. First, the kernel slides over the whole image horizontally and vertically. In addition, the dot product between the input image and the kernel is determined, where their corresponding values are multiplied and then summed up to create a single scalar value, calculated concurrently. The whole process is then repeated until no further sliding is possible. By applying padding, the size of the input image will increase, and in turn, the size of the output feature map will also increase.

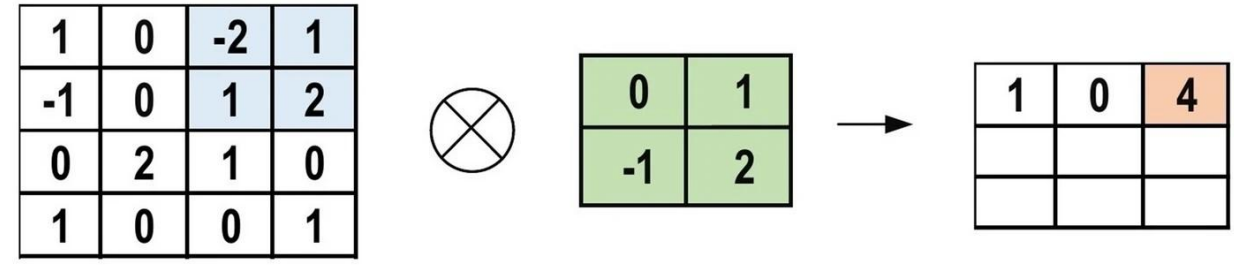
Step-1



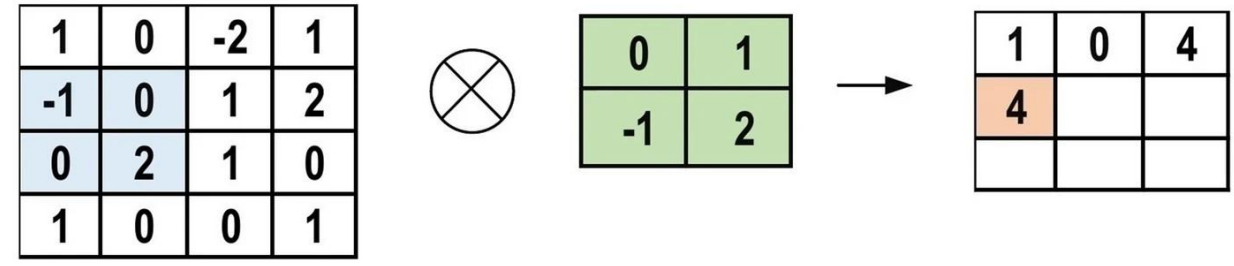
Step-2



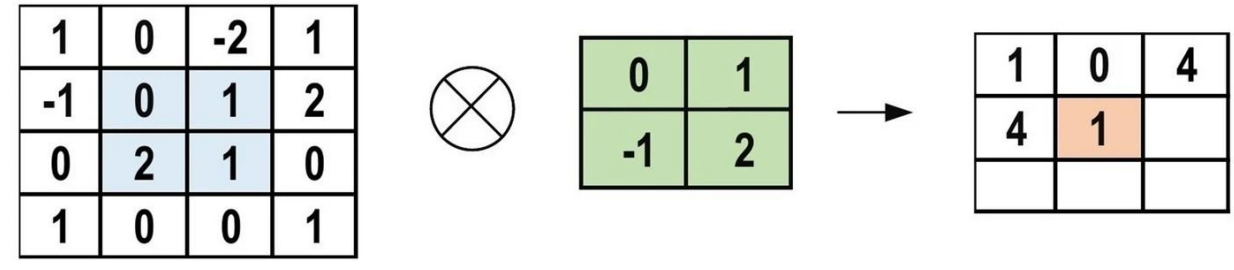
Step-3



Step-4



Step-5



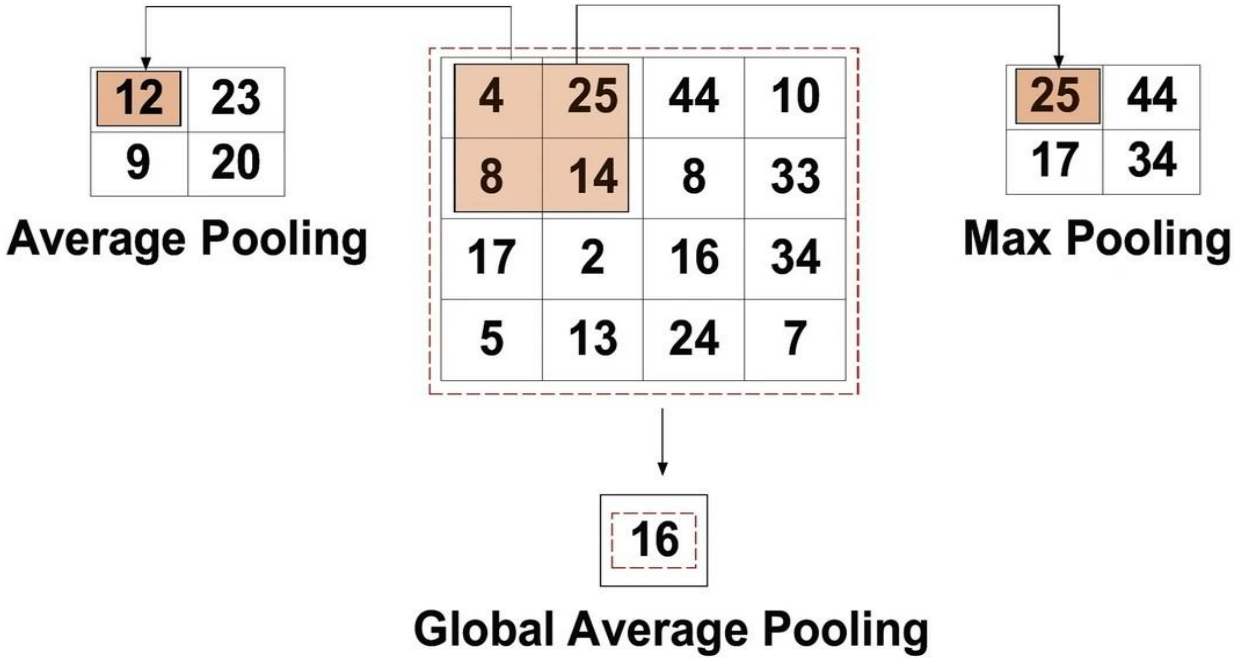
• **Sparse Connectivity:** Each neuron of a layer in FC neural networks links with all neurons in the following layer. By contrast, in CNNs, only a few weights are available between two adjacent layers. Thus, the number of required weights or connections is small, while the memory required to store these weights is also small; hence, this approach is memory-effective.

• **Weight Sharing:** There are no allocated weights between any two neurons of neighboring layers in CNN, as the whole weights operate with one and all pixels of the input matrix. Learning a single group of weights for the whole input will significantly decrease the required training time and various costs, as it is not necessary to learn additional weights for each neuron.

Pooling Layer:

In most cases, a Convolutional Layer is followed by a Pooling Layer.The main task of the pooling layer is the sub-sampling of the feature maps. These maps are generated by following the convolutional operations. In other words, this approach shrinks large-size feature maps to create smaller feature maps. Concurrently, it maintains the majority of the dominant information (or features) in every step of the pooling stage.

Several types of pooling methods are available for utilization in various pooling layers. These methods include tree pooling, gated pooling, average pooling, min pooling, max pooling, global average pooling (GAP), and global max pooling. The most familiar and frequently utilized pooling methods are the max, min, and GAP pooling.

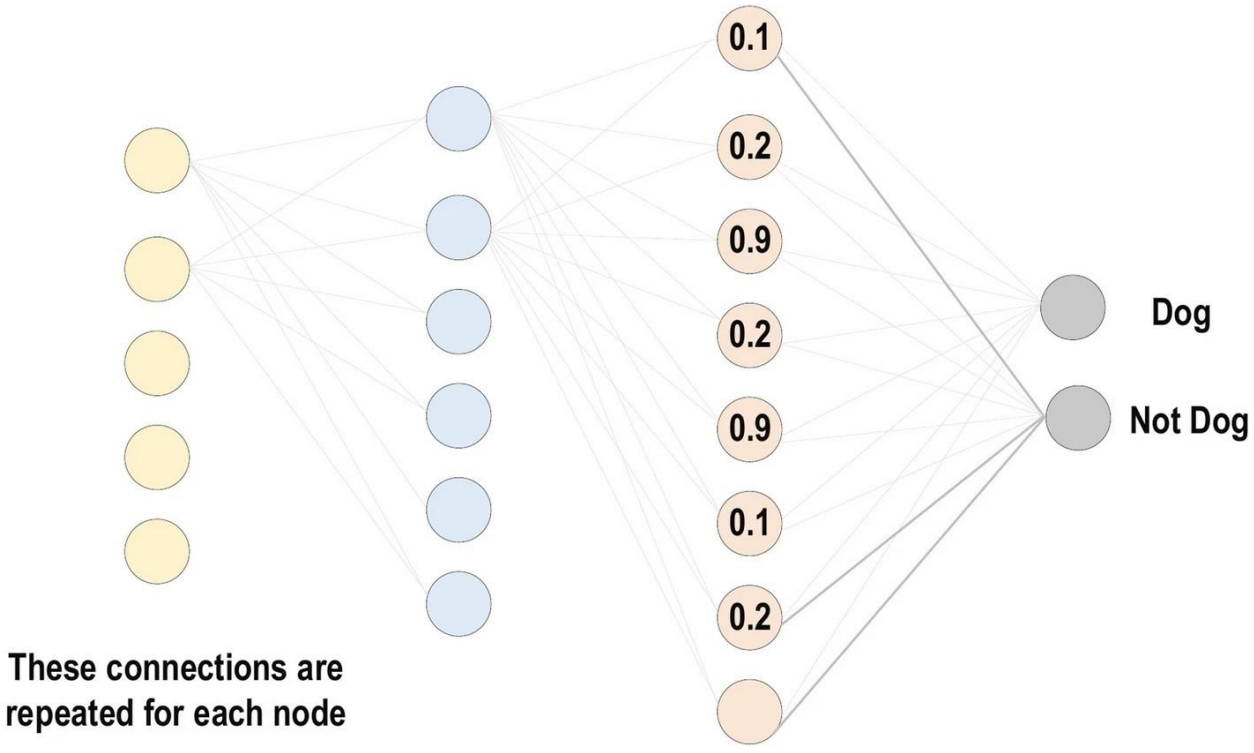


The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.This CNN model generalizes the features extracted by the convolution layer, and helps the networks to recognise the features independently. With the help of this, the computations are also reduced in a network.

Fully Connected Layer:

Commonly, this layer is located at the end of each CNN architecture. Inside this layer, each neuron is connected to all neurons of the previous layer, the so-called Fully Connected (FC) approach. It is utilized as the **CNN classifier**.
It follows the basic method of the conventional multiple-layer perceptron neural network, as it is a type of feed-forward ANN.

The input of the FC layer comes from the last pooling or convolutional layer. This input is in the form of a vector, which is created from the feature maps after fattening. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place. The reason two layers are connected is that two fully connected layers will perform better than a single connected layer. These layers in CNN reduce human supervision.



Activation Function (non-linearity):

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds **non-linearity** to the network. This non-linear performance of the activation layers means that the mapping of input to output will be non-linear; moreover, these layers give the CNN the ability to learn extra-complicated things.

The activation function must also have the ability to differentiate, which is an extremely significant feature, as it allows error back-propagation to be used to train the network.
There are several commonly used activation functions such as the **ReLU**, **Softmax**, **tanH** and the **Sigmoid** functions. Each of these functions have a specific usage.

For a binary classification CNN model, sigmoid and softmax functions are preferred and for a multi-class classification, generally softmax is used. In simple terms, activation functions in a CNN model determine whether a neuron should be activated or not. It decides whether the input to the work is important or not to predict using mathematical operations.

Loss Functions:

The previous section has presented various layer-types of CNN architecture. In addition, the final classification is achieved from the output layer, which represents the last layer of the CNN architecture.

Some loss functions are utilized in the output layer to calculate the predicted error created across the training samples in the CNN model. This error reveals the difference between the actual output and the predicted one. Next, it will be optimized through the CNN learning process. However, two parameters are used by the loss function to calculate the error. The CNN estimated output (referred to as the prediction) is the first parameter. The actual output (referred to as the label) is the second parameter.

Several types of loss function are employed in various problem types which are **Cross-Entropy or Softmax Loss Function**, **Euclidean Loss Function** and **Hinge Loss Function.**