TASK 03

Comparison of CNN and Yolo

Aspect	CNN	YOLO
Object Detection Approach	Two-step approach, region proposal and classification	One-step approach, direct bounding box and class prediction
Speed and Efficiency	Computationally expensive, multiple passes through the network	Designed for real-time processing, single pass through the network
Localization Accuracy	Excellent accuracy in localizing objects	Slightly lower accuracy for the sake of speed
Handling Small Objects	Can struggle with small object detection and precise localization	Performs relatively well in detecting small objects

Flexibility and Customization	Highly customizable architecture	Simpler architecture with limited customization options
Training Time and Dataset Size	Requires large labeled datasets and longer training time	Faster training, can work with smaller datasets

Different layers in CNN

Convolutional Neural Networks (CNNs) consist of various layers that perform different operations on the input data. Here are the commonly used layers in a typical CNN architecture:

1. Convolutional Layer: This layer performs the primary operation of convolution, applying a set of learnable filters to the input data. Each filter detects specific features in the input and produces feature maps as outputs.
2. Pooling Layer: Pooling layers reduce the spatial dimensions of the feature maps, making the network more computationally efficient. The most common pooling operation is max pooling, which selects the maximum value within a pooling window and down samples the feature maps.
3. Activation Layer: Activation layers introduce non-linearity into the network by applying an activation function element-wise to the output of a previous layer. Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.
4. Fully Connected Layer: Fully connected layers connect every neuron from the previous layer to the neurons in the current layer. These layers are typically found at the end of the network and transform the learned features into the final output predictions.
5. Dropout Layer: Dropout layers prevent overfitting by randomly setting a fraction of the input neurons to zero during training. This helps to reduce interdependencies between neurons and encourages the network to learn more robust features.

6. Batch Normalization Layer: Batch normalization layers normalize the activations of the previous layer, ensuring that the mean and variance of the inputs are consistent across the mini-batches during training. This aids in faster and more stable training.
7. Flatten Layer: A flatten layer reshapes the multi-dimensional output from the previous layer into a one-dimensional vector. This is often required before connecting to fully connected layers.