**Introduction**

The problem of object detection and classification is an inherent issue in Computer Vision. IBM defines Computer Vision as “a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action.” [IBM]

To address this problem, earlier approaches included feature extraction to identify objects, which are time consuming and highly manual. A later approach that emerged is CNNs (Convolutional Neural Networks) which is a more scalable approach that leverages methods from Linear Algebra, such as Matrix Multiplication to identify patterns in an image.

Within the standard 2D object detection, the information and the features used include edges and RGB colors [IBM] which abstracts the features (corners and edges as well as RGB channels). To position these objects, which is a task under the umbrella of object detection, a bounding box is regressed, as follows up, an image would be an input and the output if a box or a box coordinates that are around or correspond to this object (a car in an image for example) [MaxPlank].

**3D Object Detection:**

Recently, higher level applications like scene understanding and object positioning require a richer form of information such as Point Clouds, in which the regression 3D box provides not only viewpoint information, but also information about the position of the object in the 3D space [MaxPlank].

3D information is available as a set of vectors containing the x, y and z pairs representing the vertices that make up the point cloud. As the data in the pointcloud is random, and is hard to learn from, many proposals include working on the 2D images using CNNs to extract features, as the work with 2D images has been far more extensive than working with point clouds. After that, a 2D box is regressed to give the object’s position, which is further relayed to obtain the box’s 3D coordinates or the object’s z position, an example would be the works of [MobileNetSSD with Realsense Repo][MobileNetSSD repo]

Many works incorporate using the point clouds data, after voxelizing, such as VoxelNet. However, this approach suffers from giving up information during the voxelization process [VoxelNet].

Other approaches propose working directly on the pointcloud, feeding it to the network and performing the classification and/or the detection task [PointNet][PointNet++][VoteNet][BoxNet]. Though the nature of the pointcloud is random and lacks order, these approaches, almost all of them, make use of learning the pointcloud features with a modification in the network, specifically, incorporating symmetrical functions to preprocess the data for learning.

REFS:

[IBM] <https://www.ibm.com/cloud/learn/convolutional-neural-networks#toc-types-of-c-yL2bT7qZ>

[MaxPlank] <https://www.mpi-inf.mpg.de/news/spotlights/understanding-images-videos/3d-object-detection/>