In this module, I learned how convolutions are used by convolutional neural networks (CNNs) and saw how CNNs can be applied in the lab assignment. I learned that convolutions are the core operation that enables the model to learn and extract features from input data, such as images. The convolution process involves sliding a small matrix, called a kernel or filter, over the input data, performing element-wise multiplication, and summing the results to produce a single value in the output feature map. By applying multiple kernels to the input, CNNs can learn to detect various features, such as edges, textures, and patterns, at different spatial locations. The output feature maps from one convolutional layer serve as the input for the next layer, allowing the network to learn increasingly complex and abstract representations of the data as it progresses through the layers.

In this module's lab, I learned about the role of the Conv2D layer in a CNN, which uses learnable filters to extract relevant features from the input data, enabling the network to learn hierarchical feature representations essential for tasks like image classification and object detection. I also gained an understanding of the purpose of the MaxPooling2D layer, which downsamples the feature maps obtained from the convolutional layers, reducing computational complexity and controlling overfitting. Additionally, I learned about one-hot encoding, a technique used to represent categorical variables as binary vectors, and the Flatten layer, which transforms the 2D feature maps into a format that can be fed into fully connected layers. The optimizer used in this lab was Stochastic Gradient Descent (SGD), and the loss function was Cross-Entropy Loss, which is commonly used for multi-class classification problems. Although I encountered an issue with an incompatible matrix multiplication error while building and training the CNN, I was able to resolve it by adjusting the number of inputs to a linear layer affected by the pooling layer. After successfully testing the model, it achieved an F1 accuracy score of 0.98, demonstrating the effectiveness of the CNN architecture for the given task.