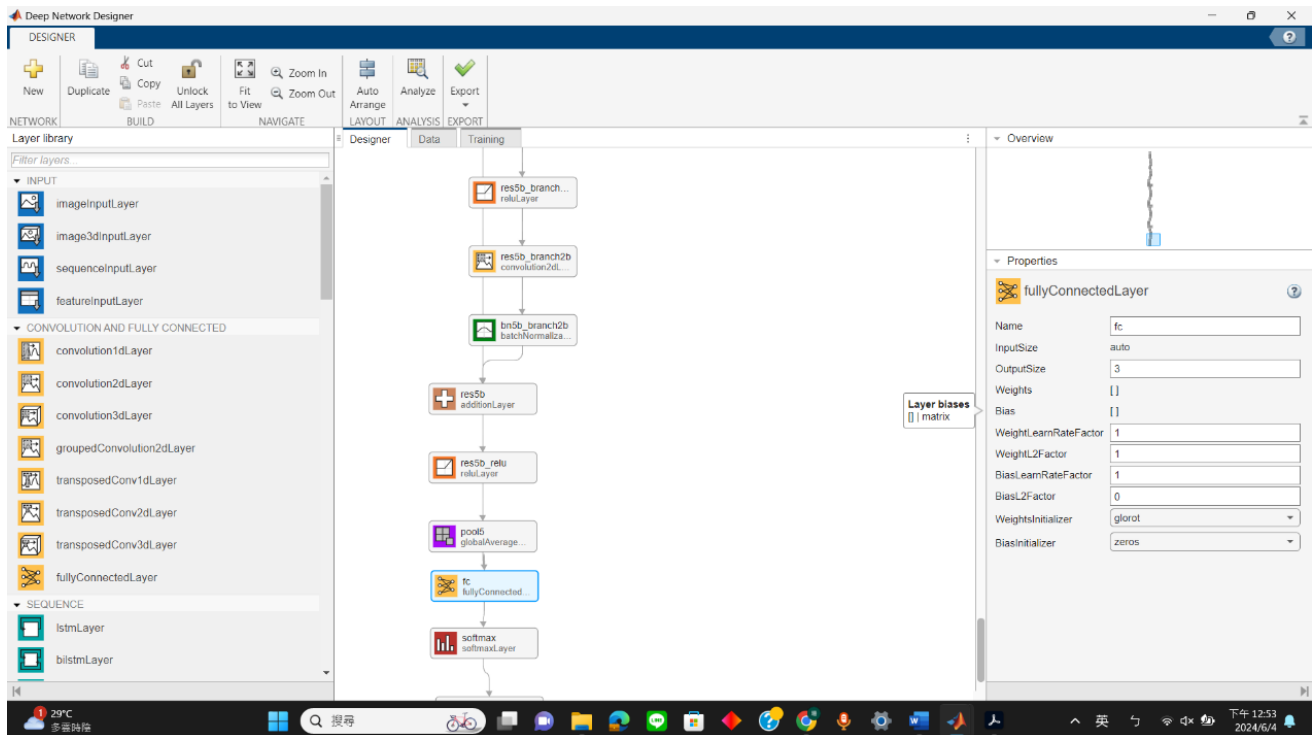


## Report\_D1265154



1. Fullyconnectedlayer: A Fullyconnectedlayer is a type of neural network layer in deep learning that connects all nodes (or features) from the previous layer to all nodes in the current layer. Each node's input is the output of all nodes from the preceding layer. It is typically used as the final layer of a neural network, responsible for classifying and predicting the features extracted by preceding convolutional and pooling operations. The output of a fully connected layer is usually a vector, with each element representing the probability of a specific class.

Input: A vector of features extracted from previous layers.

Output: A vector of size specified by the OutputSize parameter, which represents the number of classes or the number of desired outputs.

The OutputSize parameter in a fullyConnectedLayer specifies the number of neurons in this layer. Setting this parameter is crucial as it determines the dimensionality of the output vector. For instance, I modified the output size to 3, so in a classification problem with 3 classes

2. Softmaxlayer:

In MATLAB, the softmaxLayer is a neural network layer that applies the softmax function to its input. The softmax function converts a vector of raw scores (logits) into a vector of probabilities, where the probabilities sum to 1. This layer is typically

used in the final stages of a classification network to convert the outputs from the fully connected layer into a form suitable for classification.

Input: A vector of raw scores from the previous fully connected layer.

Output: A vector of probabilities, where each probability corresponds to a class and the sum of all probabilities is 1.

### 3. Classificationlayer:

In MATLAB, the `classificationLayer` is a type of neural network layer used for classification tasks. This layer computes the cross-entropy loss between the predicted probabilities from the softmax layer and the true class labels. The cross-entropy loss is used during training to measure how well the predicted probabilities match the true labels, guiding the backpropagation process to adjust the network's weights.

Input: A vector of predicted probabilities from the `softmaxLayer`.

Output: A scalar value representing the cross-entropy loss.