CUSTOM CNN ARCHITECTURE:

ARCHITECTURAL CHOICES

The model uses ➖

1. 8 Convolutional modules(CN) (#specified)
2. 8 Batch normalization layers (added for better performance)
3. 4 Max Pooling module (#specified)
4. 3 Fully Connected layers(FC) (#specified)
5. 1 Global Average Pooling Module (#specified)
6. 1 Softmax module (#specified)
7. 2 Skip connections(#specified)
8. ReLU Activation(#used in CN,FC modules)

1. 8 Convolutional Modules:

- The model uses a series of convolutional layers with increasing filter sizes (32, 64, 128, 256) to capture hierarchical features of increasing complexity.
- Increasing the number of filters in deeper layers allows the model to learn more expressive representations and capture higher-level abstractions.

2. 8 Batch Normalization layers:

- Batch normalization is applied after each convolutional layer.
- Batch normalization normalizes the activations of the previous layer, reducing the internal covariate shift and making the training process more stable.
- By normalizing the activations, batch normalization helps to mitigate the vanishing/exploding gradient problem, improving gradient flow and accelerating convergence.

3. 4 Max Pooling Modules:

- Max pooling layers are used to downsample the spatial dimensions of the feature maps.
- Max pooling helps to reduce the spatial resolution, making the model more invariant to small local variations and reducing the computational cost.
- By extracting the maximum value within each pooling region, max pooling helps to retain the most salient features while discarding less relevant information.

4. 3 Fully Connected Layers:

- The flattened output from the Global Average Pooling layer is passed through two fully connected layers with ReLU activation.
- Fully connected layers introduce non-linear transformations and capture high-level abstractions.
- The final fully connected layer uses a softmax activation to produce the output probabilities for each class, suitable for multi-class classification.

5. 1 Global Average Pooling Module:

- The architecture includes a Global Average Pooling layer after the last convolutional layer.
- Global Average Pooling reduces the spatial dimensions of the feature maps to a fixed size, regardless of their original size.
- Global Average Pooling summarizes the spatial information by taking the average value for each feature map, resulting in a fixed-length vector that represents the entire image.

6. 1 Softmax module:

- The Softmax function is used in the final fully connected layer which provides the activation for the classification of the data s.

7. 2 Skip-connections:

- Skip connections, also known as shortcut connections, are added to the network to facilitate the flow of gradients during training and help improve information flow.

- The code adds two skip connections to the network. The first skip connection connects the output of the first pooling layer (pool1) with the second batch normalization layer after the second set of convolutional layers (bn2b). This skip connection is achieved by using a 1x1 convolutional layer (Conv2D) to adjust the dimensions of pool1 and then adding it to bn2b.

- The second skip connection connects the output of the third pooling layer (pool3) with the second batch normalization layer after the fourth set of convolutional layers (bn4b). Similar to the first skip connection, it uses a 1x1 convolutional layer to adjust the dimensions of pool3 and then adds it to bn4b.

Reasons for using skip connections:

Skip connections provide an additional path for information to flow through the network.

During the training of deep neural networks, the gradients calculated during backpropagation are used to update the model's weights. However, as the gradients propagate through many layers, they can become very small, leading to the problem of vanishing gradients. When the gradients vanish, the network struggles to update the early layers, inhibiting the learning process.

Skip connections address the vanishing gradients problem by providing shortcuts for gradient flow. They allow the gradients to bypass several layers and flow directly to the earlier layers, enabling more effective learning. By incorporating skip connections, the network can propagate gradients more easily and ensure that the information reaches the earlier layers during backpropagation.

8. ReLu activation:

- Rectified Linear Unit (ReLU) activation is applied after each convolutional and fully connected layer, except for the output layer.
- It introduces non-linearity into the network, allowing it to learn complex patterns and representations.
- ReLU is computationally efficient and avoids the vanishing gradient problem associated with activation functions like sigmoid or tanh.
- ReLU has become a popular choice in deep learning architectures due to its simplicity and effectiveness.