
Algorithm 1: Detailed Training Procedure for ResNet-n Models

input : Number of layers n ; input channels $in_channels$; output channels $out_channels$; activation function $activation$; training dataset $train_data$; validation dataset val_data

output: Trained ResNet models with different layer sizes (ResNet-8, 14, 20, 26, 32) incorporating specified activation functions and architectural configurations

```
1 Normalize  $train\_data$  and  $val\_data$  to have values between 0 and 1;
2 Convert labels of  $train\_data$  and  $val\_data$  to one-hot encoding;
3 for each  $n$  in  $[8, 14, 20, 26, 32]$  do
4   Initialize ResNet- $n$  model with  $in\_channels$ ,  $out\_channels$ , and  $activation$ ;
5   Compile the model with a cross-entropy loss function and SGD optimizer;
6   for  $epoch$  in 1 to 90 do
7     foreach  $batch$  in  $train\_data$  do
8       Perform forward pass with current batch;
9       Compute loss and perform backward pass;
10      Update model weights;
11    end
12    Evaluate model on  $val\_data$  and compute validation loss and accuracy;
13    Save model checkpoint if validation accuracy improves;
14  end
15  Output the final trained model;
16 end
```
