

## INSTRUCTIONS FOR PREPROCESSING:

- We follow a simple preprocessing pipeline. The model is designed to accept the input of size 384,384
- Please ensure that the input images follow only RGB channel format with the pixel values ranging from (0,255).
- So no rescaling are required. Just make sure above two requirement are met successfully.

## DETAILS OF OUTPUT LAYER:

conv5_block32_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block32_1_relu[0][0]
conv5_block32_concat (Concatenation)	(None, 7, 7, 1664)	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]
bn (BatchNormalization)	(None, 7, 7, 1664)	6656	conv5_block32_concat[0][0]
relu (Activation)	(None, 7, 7, 1664)	0	bn[0][0]
avg_pool (GlobalAveragePooling2)	(None, 1664)	0	relu[0][0]
batch_normalization (BatchNormalization)	(None, 1664)	6656	avg_pool[0][0]
final_softmax (Dense)	(None, 5)	8325	batch_normalization[0][0]
=====			
Total params: 12,657,861			
Trainable params: 12,496,133			
Non-trainable params: 161,728			

- The model outputs the probabilities from final layer (**final\_softmax**) in the order [**No\_device, device\_A, device\_B, device\_C, device\_D**] (which is corrected to order [**device\_A, device\_B, device\_C, device\_D, No\_Device**] in csv.)
- The output is of shape (**n,5**) array for above mentioned classes where n refers to batch size(number of input images) of given input.
- The output from final layer bears brackets as they are raw tensors which is flattened to values when predicted to csv.