input_1	input:	[(None, 48, 48, 3)]	[(None, 48, 48, 3)]
InputLayer	output:		

augmentation	input:	(None, 48, 48, 3)	(None, 48, 48, 3)
Sequential	output:		

block1_conv1		input:	(None, 48, 48, 3)	(None, 48, 48, 64)
Conv2D	relu	output:		

block1_conv2		input:	(None, 48, 48, 64)	(None, 48, 48, 64)
Conv2D	relu	output:		

block1_normalizer	input:	(None, 48, 48, 64)	(None, 48, 48, 64)
BatchNormalization	output:		

block1_pool	input:	(None, 48, 48, 64)	(None, 24, 24, 64)
MaxPooling2D	output:		

block2_conv1		input:	(None, 24, 24, 64)	(None, 24, 24, 128)
Conv2D	relu	output:		

block2_conv2		input:	(None, 24, 24, 128)	(None, 24, 24, 128)
Conv2D	relu	output:		

block2_normalizer	input:	(None, 24, 24, 128)	(None, 24, 24, 128)
BatchNormalization	output:		

block2_pool	input:	(None, 24, 24, 128)	(None, 12, 12, 128)
MaxPooling2D	output:		

block3_conv1		input:	(None, 12, 12, 128)	(None, 12, 12, 256)
Conv2D	relu	output:		

block3_conv2		input:	(None, 12, 12, 256)	(None, 12, 12, 256)
Conv2D	relu	output:		

block3_conv3		input:	(None, 12, 12, 256)	(None, 12, 12, 256)
Conv2D	relu	output:		

block3_normalizer	input:	(None, 12, 12, 256)	(None, 12, 12, 256)
BatchNormalization	output:		

block3_pool	input:	(None, 12, 12, 256)	(None, 6, 6, 256)
MaxPooling2D	output:		