Use VGG models as they achieved top performance in ILSVRC 2014 competition, modular stucture is easy to understand.

Consist of:

-Convolution layer with small 3x3 filters.

-Max pooling mapping layer

-form into block with layer in each box increase 32, 64, 128, 256....

-Padding to make sure output gesture matches the upcomming input layer.

-ReLU activation layer(avoid vanishing gradient descent) and He weight initialization.

MANUALLY(with img augmentation):

ARCHITECT NUMBER ONE:

# block 1

model.add(Conv2D(32, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same', input\_shape=(224, 224, 3)))

model.add(MaxPooling2D((2, 2)))

# block 2

model.add(Conv2D(64, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same'))

model.add(MaxPooling2D((2, 2)))

# block 3

model.add(Conv2D(128, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same'))

model.add(MaxPooling2D((2, 2)))

# block 4

model.add(Conv2D(256, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same'))

model.add(MaxPooling2D((2, 2)))

# block 5

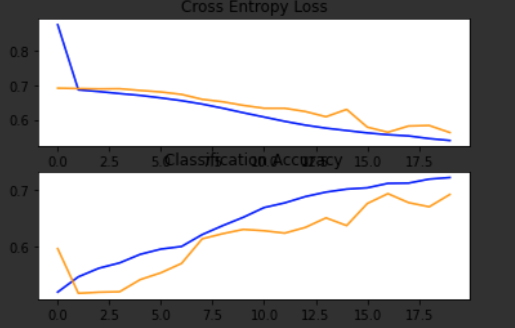
model.add(Conv2D(512, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same'))

model.add(MaxPooling2D((2, 2)))

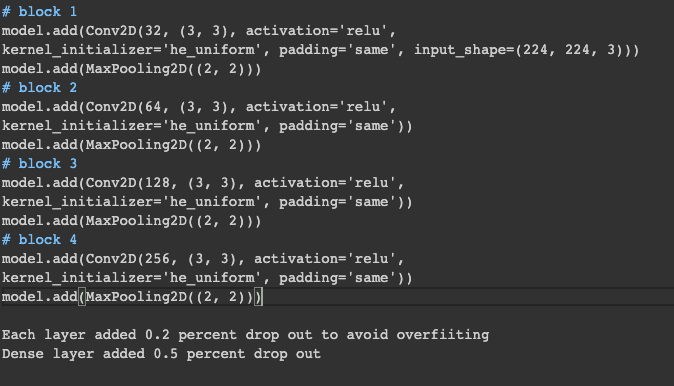
Each layer added 0.2 percent drop out to avoid overfiiting

Dense layer added 0.5 percent drop out

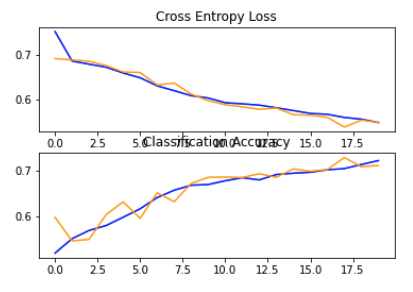
Accuracy 69.142



ARCHITECT SECOND:

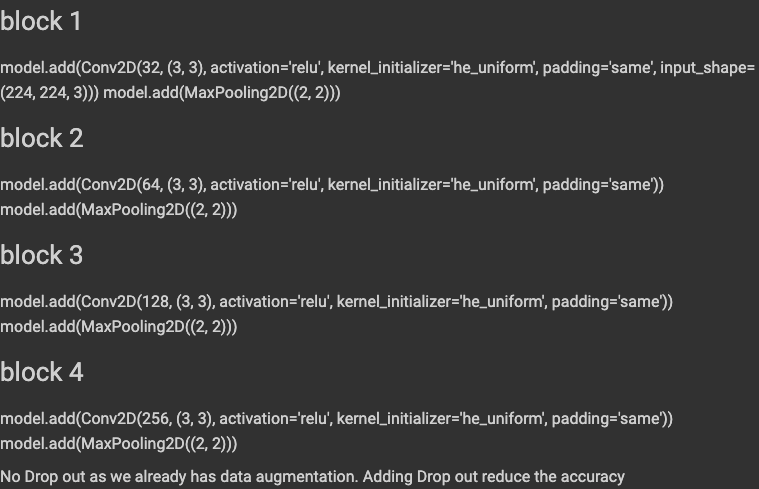


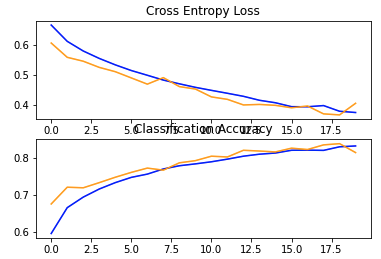




3rd ARCHITECT:

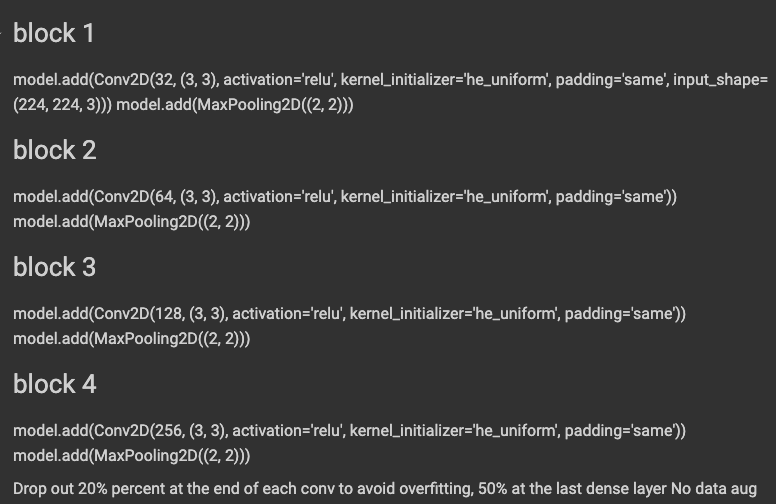






4th architeciture:





Architect 5th:



block 1

model.add(Conv2D(32, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same', input\_shape=(224, 224, 3))) model.add(MaxPooling2D((2, 2)))

block 2

model.add(Conv2D(64, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))

block 3

model.add(Conv2D(128, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))

block 4

model.add(Conv2D(256, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))

block 5

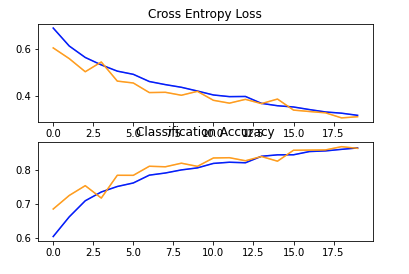
model.add(Conv2D(512, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))

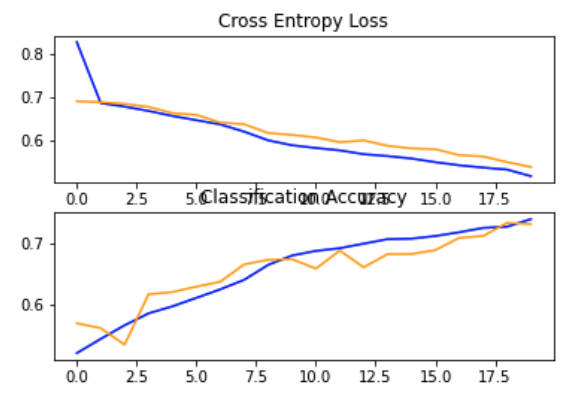
block 6

model.add(Conv2D(1024, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))

block 7

model.add(Conv2D(2048, (3, 3), activation='relu', kernel\_initializer='he\_uniform', padding='same')) model.add(MaxPooling2D((2, 2)))





With use of transfer learning from VGG-16



