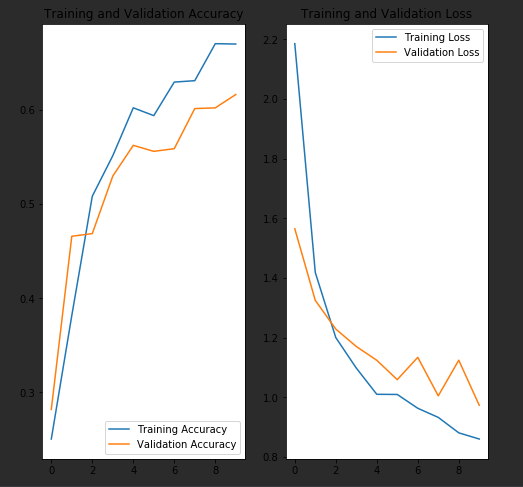


model = Sequential()  
model.add(Conv2D(32, 3, strides = 1, padding = "same", activation = "relu", input\_shape = (IMG\_HEIGHT, IMG\_WIDTH, 3)))  
model.add(Conv2D(32, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Conv2D(64, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(Conv2D(64, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Conv2D(128, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(Conv2D(128, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Flatten())  
model.add(Dense(4096, activation = "relu"))  
model.add(Dense(4096, activation = "relu"))  
model.add(Dense(1000, activation = "relu"))  
model.add(Dense(5, activation = "softmax"))

model.compile(optimizer = 'adam',  
 loss = 'categorical\_crossentropy',  
 metrics = ['accuracy'])

4gb



model = Sequential()  
model.add(Conv2D(32, 3, strides = 1, padding = "same", activation = "relu", input\_shape = (IMG\_HEIGHT, IMG\_WIDTH, 3)))  
model.add(Conv2D(32, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Conv2D(64, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(Conv2D(64, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Conv2D(128, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(Conv2D(128, 3, strides = 1, padding = "same", activation = "relu"))  
model.add(MaxPooling2D((2, 2)))  
  
model.add(Flatten())  
model.add(Dense(4096, activation = "relu"))  
model.add(Dense(1000, activation = "relu"))  
model.add(Dense(5, activation = "softmax"))

model.compile(optimizer = 'adam',  
 loss = 'categorical\_crossentropy',  
 metrics = ['accuracy'])

3.9gb