Base Model

Optimizers

1. Adam
2. Adamax
3. Adagrad
4. RMSProp
5. SGD
6. Adadelta
7. tf.keras.losses.SparseCategoricalCrossentropy

tf.keras.losses. CategoricalHinge

1. tf.keras.losses.CosineSimilarity

Model 1:

### input Shape

input\_shape = (img\_height,img\_width,3)

# Model

model = Sequential([

  layers.experimental.preprocessing.Rescaling(1./255, input\_shape=input\_shape),

  layers.Conv2D(32, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(32, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(64, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(64, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Flatten(),

  layers.Dense(512, activation='relu'),

  layers.Dense(num\_classes)

])

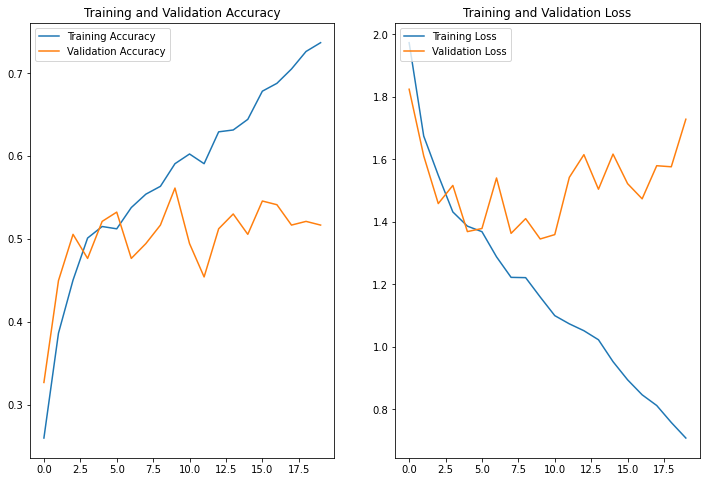
### Used adam optimizer and loss as SparceCategoricalCrossentropy

model.compile(optimizer='adam',

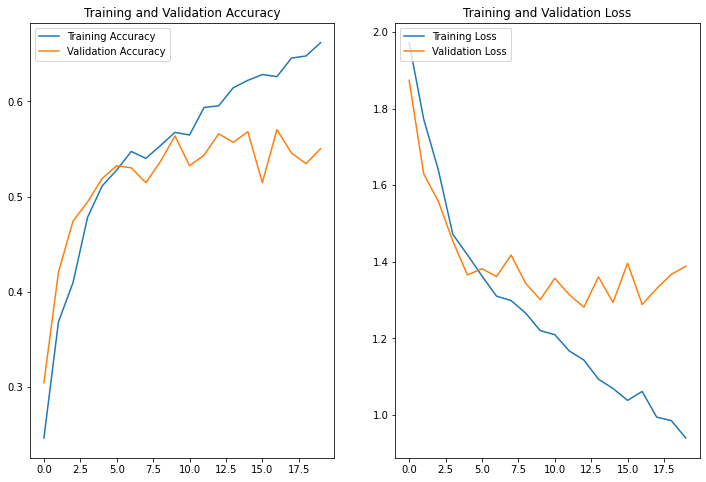
              loss=tf.losses.SparseCategoricalCrossentropy(from\_logits=True),

              metrics=['accuracy'])

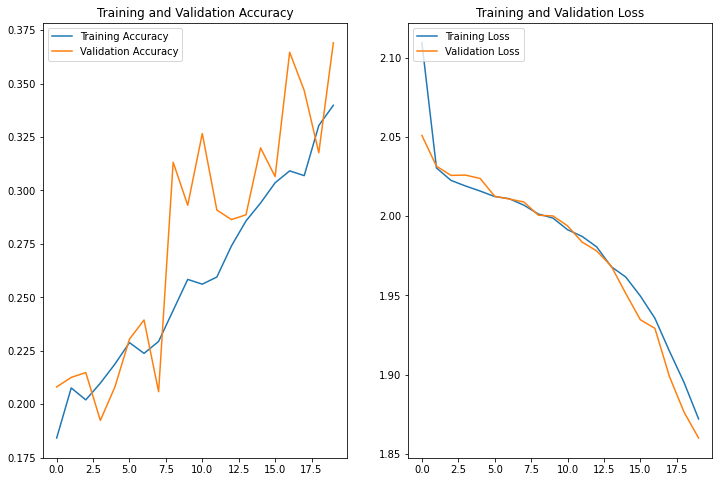
plotthis(model)



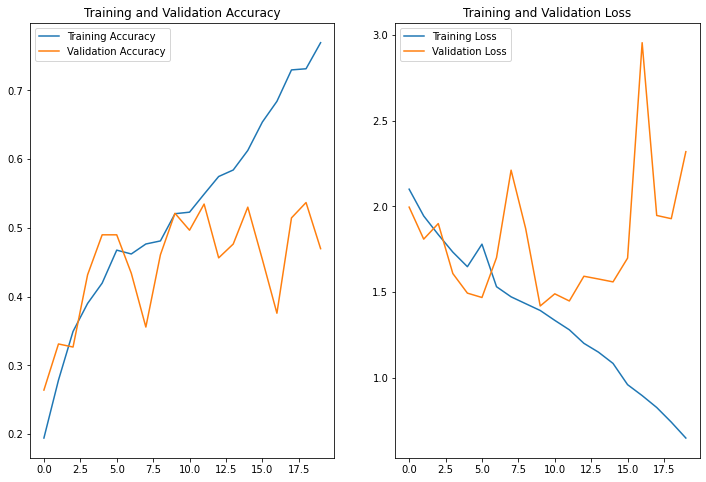
Model 2 = Model1 > Adamax



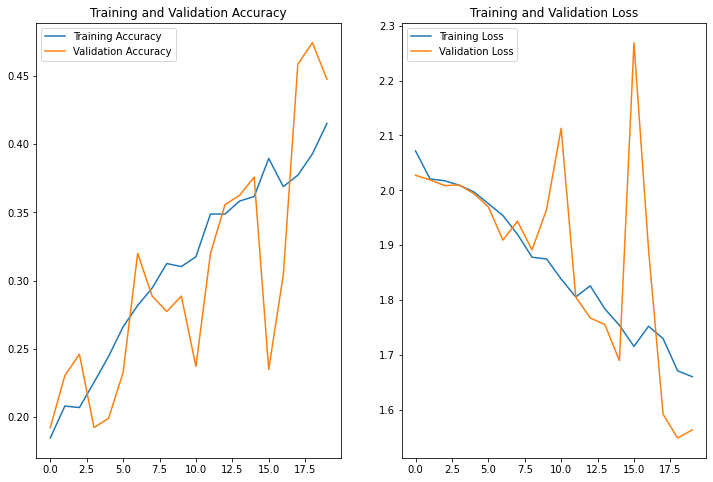
Model 3 = Model1 > Adagrad



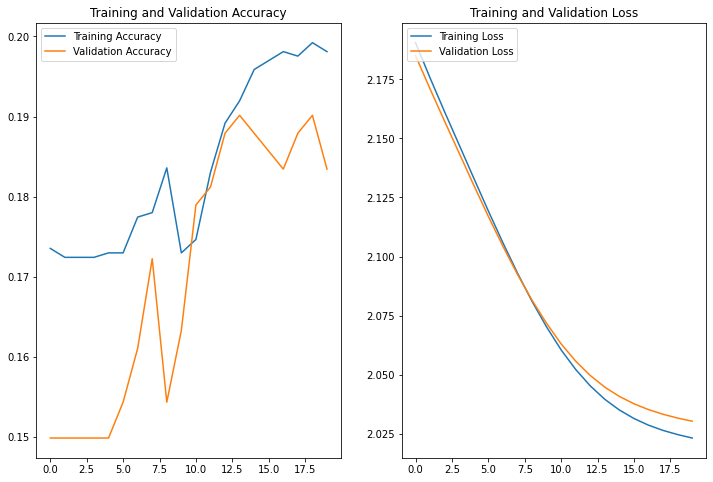
Model 4 > Rmsprop



Model 5 SGD



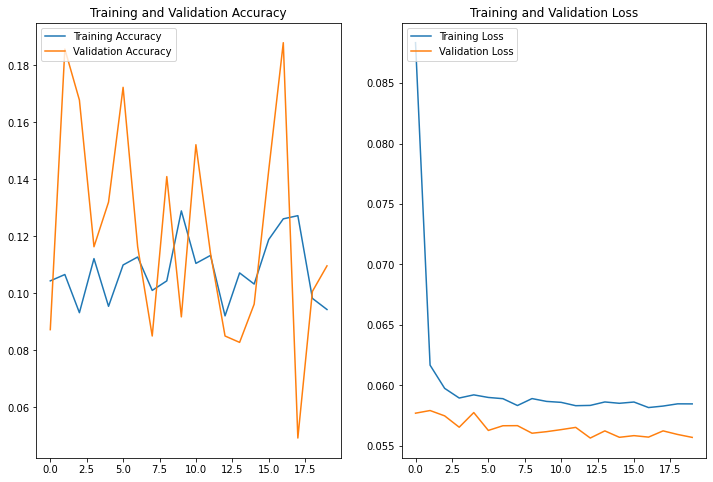
Model 6 > adadelta



Model 7 adamax and

tf.losses.SparseCategoricalCrossentropy(from\_logits=True)(same as model2)

Model 7 tf.losses. CategoricalHinge (from\_logits=True



Model 8 KLD

Adamax – sparse –

model = Sequential([

  layers.experimental.preprocessing.Rescaling(1./255, input\_shape=input\_shape),

  layers.Conv2D(32, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(32, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(64, 3, activation='relu'),

  layers.MaxPooling2D(),

  layers.Conv2D(64, 3, activation='relu'),

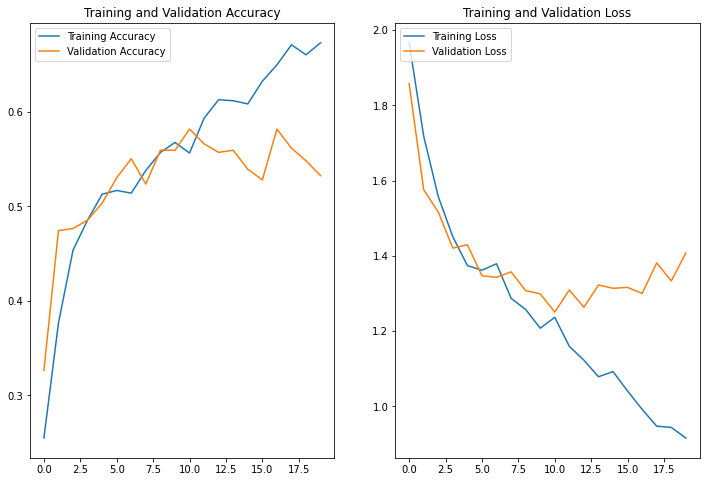
  layers.MaxPooling2D(),

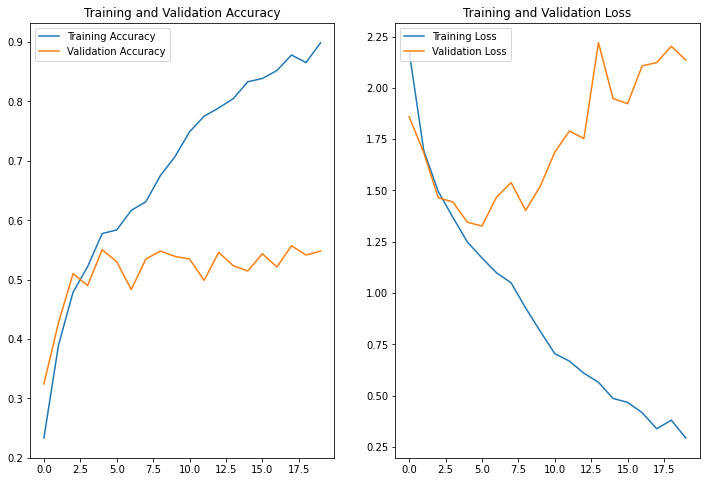
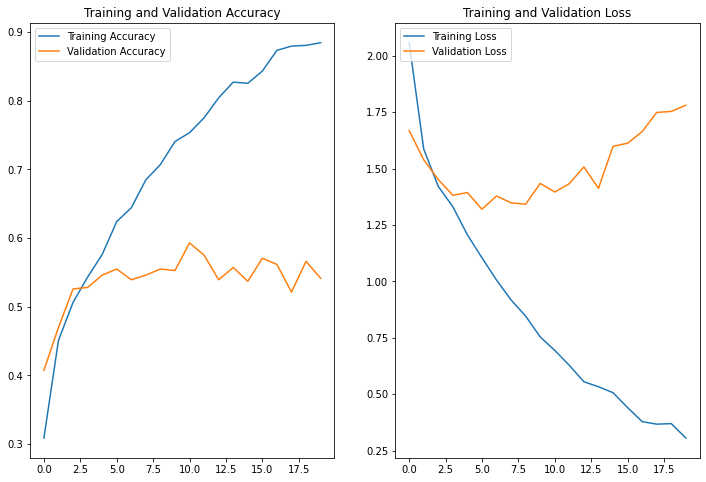
  layers.Flatten(),

  layers.Dense(512, activation='relu'),

  layers.Dense(num\_classes)

])



32p32p64p128d

