CONFIGURE THE LEARNING PROCESS

- The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires loss function during the model compilation process.

- Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer

- Metrics are used to evaluate the performance of your model. It is similar to the loss function, but not used in the training process

```python
# Since we use categorical_crossentropy loss, we need categorical labels

class_mode=('categorical')


from tensorflow.keras.optimizers import RMSprop
early = tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=5)
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['accuracy'])

total_sample=train_generator.n

n_epochs = 20

history = model.fit_generator(
        train_generator,
        steps_per_epoch=int(total_sample/batch_size),
        epochs=n_epochs,
        verbose=1)


model.save('model.h5')


acc = history.history['accuracy']

loss = history.history['loss']

epochs = range(1, len(acc) + 1)
```