

First we need to define our Metric.

```
METRICS = [  
    tf.keras.metrics.BinaryAccuracy(),  
    tf.keras.metrics.Precision(name="precision"),  
    tf.keras.metrics.Recall(name="recall"),  
]
```

Then we'll create CNN architecture.

```
model = Sequential()  
model.add(Conv2D(16, (3,3),padding='Same',activation='relu',input_shape  
= (128,128,3)))  
model.add(Conv2D(16, (3,3),activation='relu',padding='Same'))  
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))  
model.add(BatchNormalization())  
model.add(Dropout(0.3))  
  
model.add(Conv2D(32, (3,3),activation='relu',padding='Same'))  
model.add(Conv2D(32, (3,3),activation='relu',padding='Same'))  
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))  
model.add(BatchNormalization())  
model.add(Dropout(0.3))  
  
model.add(Conv2D(64, (3,3),activation='relu',padding='Same'))  
model.add(Conv2D(64, (3,3),activation='relu',padding='Same'))  
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))  
model.add(BatchNormalization())  
model.add(Dropout(0.5))  
  
model.add(Conv2D(64, (3,3),activation='relu',padding='Same'))  
model.add(Conv2D(64, (3,3),activation='relu',padding='Same'))  
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))  
model.add(BatchNormalization())  
model.add(Dropout(0.5))  
  
model.add(Flatten())  
model.add(Dense(512,activation='relu'))  
model.add(BatchNormalization())  
model.add(Dropout(0.7))
```

```
model.add(Dense(128,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(64,activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(1 , activation = 'sigmoid'))
adam =
Adam(learning_rate=lr_schedule,beta_1=0.9,beta_2=0.999,epsilon=1e-07)
model.compile(optimizer = adam,loss = 'binary_crossentropy' , metrics =
METRICS)
```