

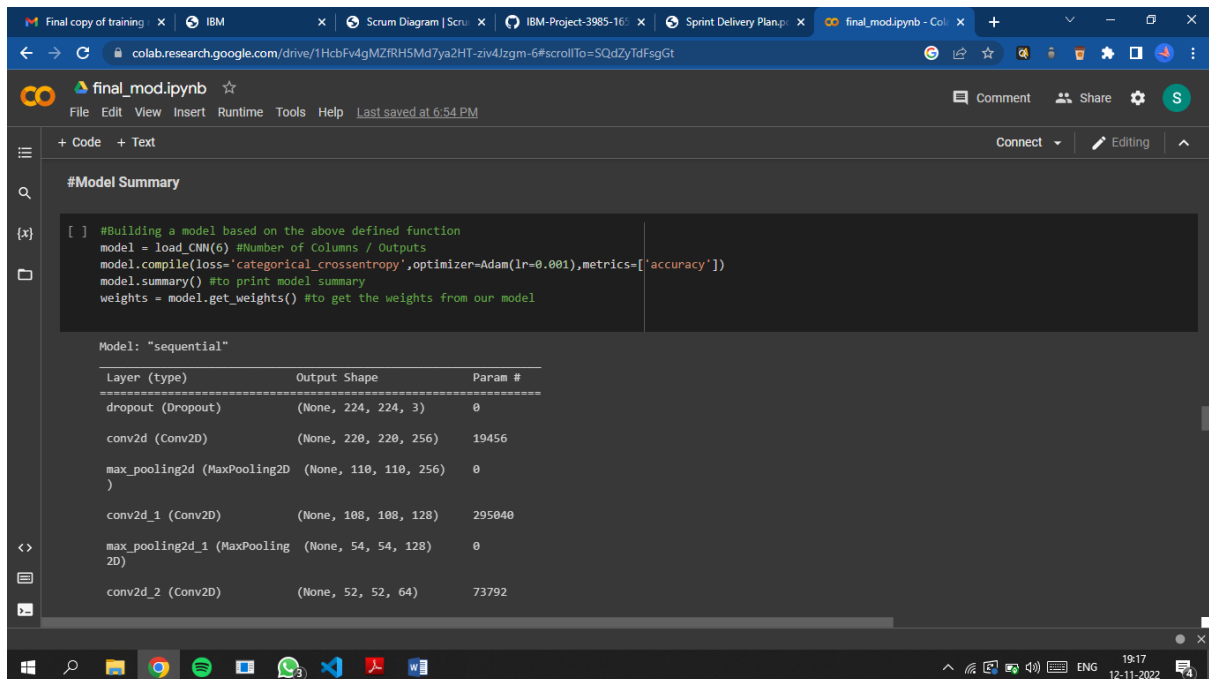
BUILDING A CNN MODEL

BUILDING A CNN MODEL USING THE COLLECTED DATA:

```
Final copy of training x IBM x Scrum Diagram | Scru x IBM-Project-3985-16 x Sprint Delivery Plan.p x final_mod.ipynb - Col x + - x
colab.research.google.com/drive/1HcbFv4gMZrH5Md7ya2HT-ziv4Jzgm-6#scrollTo=SQdZyTdfsgGt
final_mod.ipynb
File Edit View Insert Runtime Tools Help Last saved at 6:54 PM
+ Code + Text
Getting Started with Convolutional Neural Networks (CNN)
Start(sequential)
#MODEL BUILDING
[ ] early_stop_loss = EarlyStopping(monitor='loss', patience=3, verbose=1)
early_stop_val_acc = EarlyStopping(monitor='val_accuracy', patience=3, verbose=1)
model_callbacks=[early_stop_loss, early_stop_val_acc]
Add Layers(Conv, Maxpool, Flatten, Dense, Dropout)
[ ] #defining our model, All the layers and configurations
def load_CNN(output_size):
    K.clear_session()
    model = Sequential()
    model.add(Dropout(0.4,input_shape=(224, 224, 3)))
    model.add(Conv2D(256, (5, 5),input_shape=(224, 224, 3),activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))
    #model.add(BatchNormalization())
    model.add(Conv2D(128, (3, 3), activation='relu'))
    Building Model(Summary, Compile, Fit, Predict)
#Model Summary
```

```
model.add(MaxPool2D(pool_size=(2, 2)))
#model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
#model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
#model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(output_size, activation='softmax'))
return model
Building Model(Summary, Compile, Fit, Predict)
#Model Summary
```

MODEL SUMMARY:

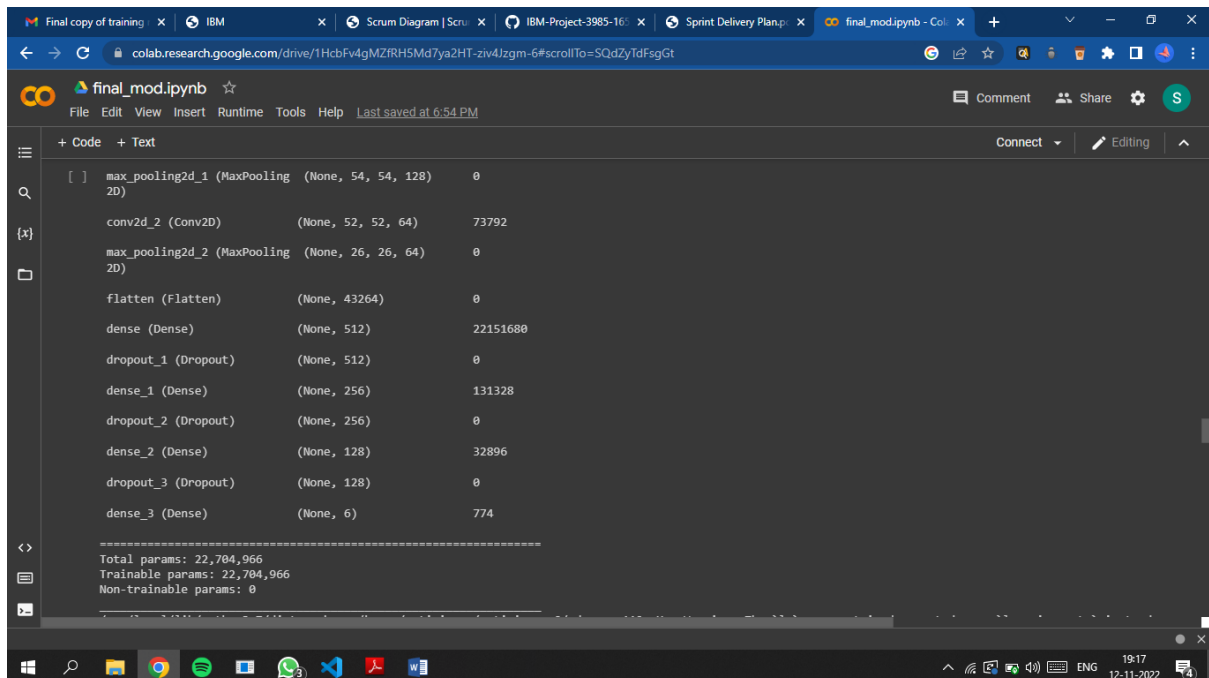


The screenshot shows a Google Colab notebook titled "final_mod.ipynb". The code cell contains the following Python code:

```
[ ] #Building a model based on the above defined function
model = load_CNN(6) #Number of Columns / Outputs
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.001),metrics=['accuracy'])
model.summary() #to print model summary
weights = model.get_weights() #to get the weights from our model
```

The output of the code is a summary of the model architecture, titled "Model: 'sequential'". It lists the layers, their output shapes, and the number of parameters for each layer.

Layer (type)	Output Shape	Param #
dropout (Dropout)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 220, 220, 256)	19456
max_pooling2d (MaxPooling2D)	(None, 110, 110, 256)	0
conv2d_1 (Conv2D)	(None, 108, 108, 128)	295040
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 128)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	73792



The screenshot shows the same Google Colab notebook, but the code cell now contains the full model summary, including the final dense and dropout layers. The code is:

```
[ ] max_pooling2d_1 (MaxPooling2D) (None, 54, 54, 128) 0
conv2d_2 (Conv2D) (None, 52, 52, 64) 73792
max_pooling2d_2 (MaxPooling2D) (None, 26, 26, 64) 0
flatten (Flatten) (None, 43264) 0
dense (Dense) (None, 512) 22151680
dropout_1 (Dropout) (None, 512) 0
dense_1 (Dense) (None, 256) 131328
dropout_2 (Dropout) (None, 256) 0
dense_2 (Dense) (None, 128) 32896
dropout_3 (Dropout) (None, 128) 0
dense_3 (Dense) (None, 6) 774
```

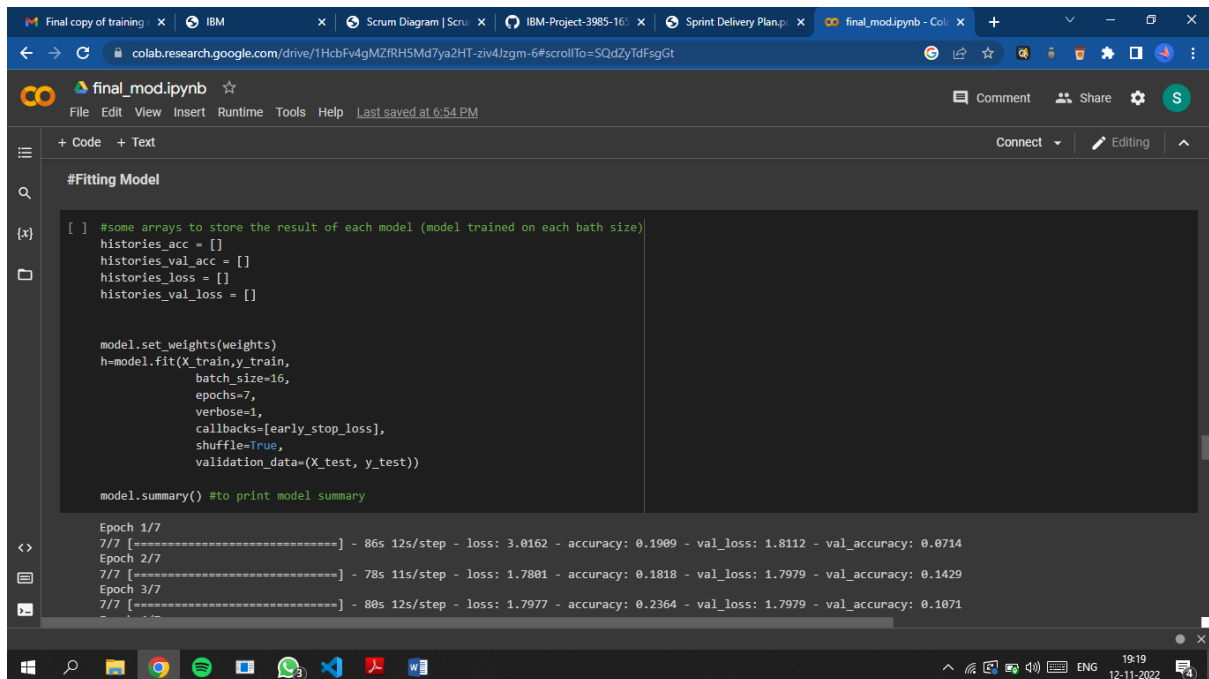
The output shows the complete model summary, including the final dense and dropout layers. It also provides the total number of parameters, trainable parameters, and non-trainable parameters.

Layer (type)	Output Shape	Param #
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 128)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	73792
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
flatten (Flatten)	(None, 43264)	0
dense (Dense)	(None, 512)	22151680
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 6)	774

Summary statistics:

- Total params: 22,704,966
- Trainable params: 22,704,966
- Non-trainable params: 0

FITTING THE MODEL:



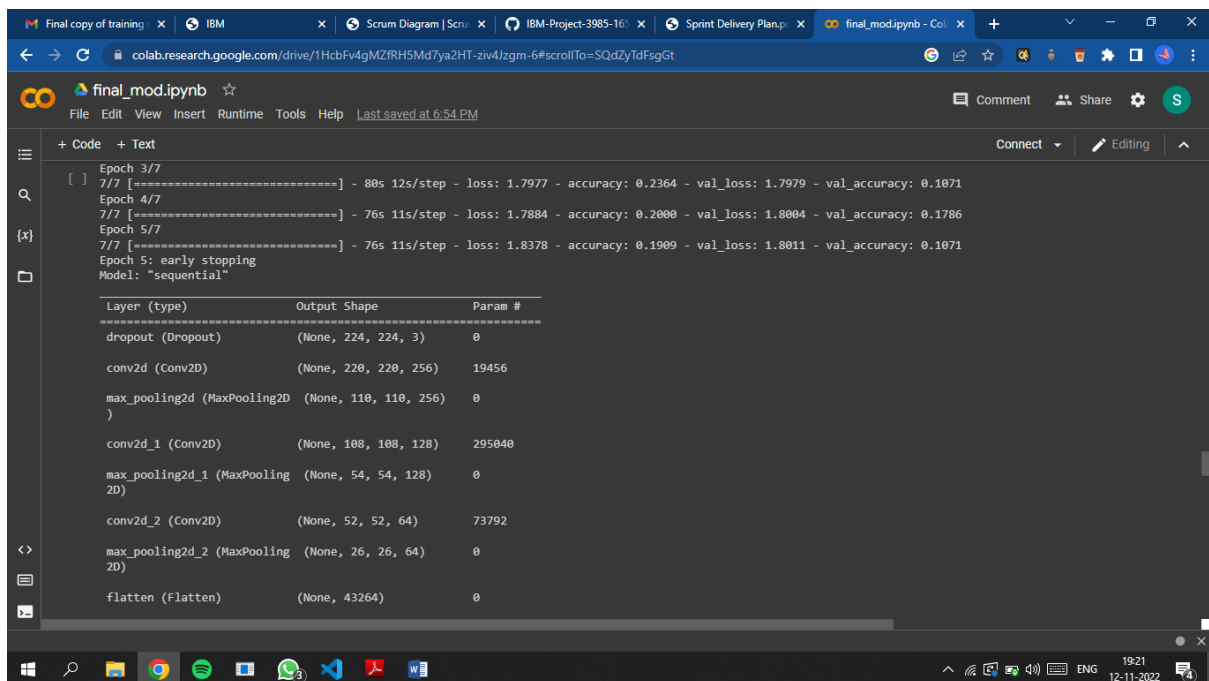
```
#Fitting Model

[ ] #some arrays to store the result of each model (model trained on each bath size)
histories_acc = []
histories_val_acc = []
histories_loss = []
histories_val_loss = []

model.set_weights(weights)
h=model.fit(X_train,y_train,
            batch_size=16,
            epochs=7,
            verbose=1,
            callbacks=[early_stop_loss],
            shuffle=True,
            validation_data=(X_test, y_test))

model.summary() #to print model summary

Epoch 1/7
7/7 [=====] - 86s 12s/step - loss: 3.0162 - accuracy: 0.1909 - val_loss: 1.8112 - val_accuracy: 0.0714
Epoch 2/7
7/7 [=====] - 78s 11s/step - loss: 1.7801 - accuracy: 0.1818 - val_loss: 1.7979 - val_accuracy: 0.1429
Epoch 3/7
7/7 [=====] - 80s 12s/step - loss: 1.7977 - accuracy: 0.2364 - val_loss: 1.7979 - val_accuracy: 0.1071
```



```
Epoch 3/7
7/7 [=====] - 80s 12s/step - loss: 1.7977 - accuracy: 0.2364 - val_loss: 1.7979 - val_accuracy: 0.1071
Epoch 4/7
7/7 [=====] - 76s 11s/step - loss: 1.7884 - accuracy: 0.2000 - val_loss: 1.8004 - val_accuracy: 0.1786
Epoch 5/7
7/7 [=====] - 76s 11s/step - loss: 1.8378 - accuracy: 0.1909 - val_loss: 1.8011 - val_accuracy: 0.1071
Epoch 5: early stopping
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dropout (Dropout)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 220, 220, 256)	19456
max_pooling2d (MaxPooling2D)	(None, 110, 110, 256)	0
conv2d_1 (Conv2D)	(None, 108, 108, 128)	295040
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 128)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	73792
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
flatten (Flatten)	(None, 43264)	0

Final copy of training x IBM x Scrum Diagram | Scr x IBM-Project-3985-16 x Sprint Delivery Plan x final_mod.ipynb - Colab

colab.research.google.com/drive/1HcbFv4gMZrRH5Md7ya2HT-ziv4Jzgm-6#scrollTo=SQdZyTdfsgGt

final_mod.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:54 PM

+ Code + Text

```
[ ] max_pooling2d_1 (MaxPooling (None, 54, 54, 128) 0
2D)

conv2d_2 (Conv2D) (None, 52, 52, 64) 73792

max_pooling2d_2 (MaxPooling (None, 26, 26, 64) 0
2D)

flatten (Flatten) (None, 43264) 0

dense (Dense) (None, 512) 22151680

dropout_1 (Dropout) (None, 512) 0

dense_1 (Dense) (None, 256) 131328

dropout_2 (Dropout) (None, 256) 0

dense_2 (Dense) (None, 128) 32896

dropout_3 (Dropout) (None, 128) 0

dense_3 (Dense) (None, 6) 774

=====
Total params: 22,704,966
Trainable params: 22,704,966
Non-trainable params: 0
```

19:21 12-11-2022

Final copy of training x IBM x Scrum Diagram | Scr x IBM-Project-3985-16 x Sprint Delivery Plan x final_mod.ipynb - Colab

colab.research.google.com/drive/1HcbFv4gMZrRH5Md7ya2HT-ziv4Jzgm-6#scrollTo=SQdZyTdfsgGt

final_mod.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:54 PM

+ Code + Text

```
[ ] model.set_weights(weights)
h=model.fit(X_train,y_train,
            batch_size=16,
            epochs=15,
            verbose=1,
            callbacks=[early_stop_loss],
            shuffle=True,
            validation_data=(X_test, y_test))

Epoch 1/15
7/7 [=====] - 91s 13s/step - loss: 1.9735 - accuracy: 0.1182 - val_loss: 1.7991 - val_accuracy: 0.1071
Epoch 2/15
7/7 [=====] - 79s 11s/step - loss: 1.8036 - accuracy: 0.1727 - val_loss: 1.8049 - val_accuracy: 0.1786
Epoch 3/15
7/7 [=====] - 81s 11s/step - loss: 1.8095 - accuracy: 0.1091 - val_loss: 1.7935 - val_accuracy: 0.1429
Epoch 4/15
7/7 [=====] - 80s 12s/step - loss: 1.8223 - accuracy: 0.2545 - val_loss: 1.7937 - val_accuracy: 0.1786
Epoch 5/15
7/7 [=====] - 79s 11s/step - loss: 1.7786 - accuracy: 0.2364 - val_loss: 1.7892 - val_accuracy: 0.1786
Epoch 6/15
7/7 [=====] - 78s 11s/step - loss: 1.7422 - accuracy: 0.3000 - val_loss: 1.7855 - val_accuracy: 0.2143
Epoch 7/15
7/7 [=====] - 81s 12s/step - loss: 1.7244 - accuracy: 0.3182 - val_loss: 1.7824 - val_accuracy: 0.1429
Epoch 8/15
7/7 [=====] - 84s 12s/step - loss: 1.5872 - accuracy: 0.3545 - val_loss: 1.7056 - val_accuracy: 0.2500
Epoch 9/15
7/7 [=====] - 77s 11s/step - loss: 1.2719 - accuracy: 0.5091 - val_loss: 1.6680 - val_accuracy: 0.3571
Epoch 10/15
```

19:21 12-11-2022

Final copy of training x IBM x Scrum Diagram | Scr x IBM-Project-3985-16 x Sprint Delivery Plan.p x final_mod.ipynb - Col x +

colab.research.google.com/drive/1HcbFv4gMZRH5Md7ya2HT-ziv4Izgm-6#scrollTo=SQdZyTdFsgGt

final_mod.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at 6:54 PM

Comment Share S

+ Code + Text Connect Editing

```
7/7 [=====] - 91s 13s/step - loss: 1.9735 - accuracy: 0.1182 - val_loss: 1.7991 - val_accuracy: 0.1071
Epoch 2/15
7/7 [=====] - 79s 11s/step - loss: 1.8036 - accuracy: 0.1727 - val_loss: 1.8049 - val_accuracy: 0.1786
Epoch 3/15
7/7 [=====] - 81s 11s/step - loss: 1.8095 - accuracy: 0.1091 - val_loss: 1.7935 - val_accuracy: 0.1429
Epoch 4/15
7/7 [=====] - 80s 12s/step - loss: 1.8223 - accuracy: 0.2545 - val_loss: 1.7937 - val_accuracy: 0.1786
Epoch 5/15
7/7 [=====] - 79s 11s/step - loss: 1.7786 - accuracy: 0.2364 - val_loss: 1.7892 - val_accuracy: 0.1786
Epoch 6/15
7/7 [=====] - 78s 11s/step - loss: 1.7422 - accuracy: 0.3000 - val_loss: 1.7855 - val_accuracy: 0.2143
Epoch 7/15
7/7 [=====] - 81s 12s/step - loss: 1.7244 - accuracy: 0.3182 - val_loss: 1.7824 - val_accuracy: 0.1429
Epoch 8/15
7/7 [=====] - 84s 12s/step - loss: 1.5872 - accuracy: 0.3545 - val_loss: 1.7056 - val_accuracy: 0.2500
Epoch 9/15
7/7 [=====] - 77s 11s/step - loss: 1.2719 - accuracy: 0.5091 - val_loss: 1.6680 - val_accuracy: 0.3571
Epoch 10/15
7/7 [=====] - 77s 11s/step - loss: 1.0414 - accuracy: 0.6273 - val_loss: 1.5946 - val_accuracy: 0.5000
Epoch 11/15
7/7 [=====] - 78s 11s/step - loss: 0.6647 - accuracy: 0.7727 - val_loss: 1.6963 - val_accuracy: 0.3571
Epoch 12/15
7/7 [=====] - 84s 12s/step - loss: 0.4295 - accuracy: 0.8636 - val_loss: 2.2821 - val_accuracy: 0.3929
Epoch 13/15
7/7 [=====] - 78s 11s/step - loss: 0.3337 - accuracy: 0.8545 - val_loss: 2.2652 - val_accuracy: 0.2500
Epoch 14/15
7/7 [=====] - 78s 11s/step - loss: 0.3625 - accuracy: 0.9273 - val_loss: 2.7422 - val_accuracy: 0.3571
Epoch 15/15
7/7 [=====] - 80s 12s/step - loss: 0.1846 - accuracy: 0.9273 - val_loss: 3.5091 - val_accuracy: 0.2857
```

19:21 12-11-2022