SELECTED -2 PRESENTATION

ALI EHAB ALI YOUSEF	201900483
NEHAD AHMED ALI NOURELDEIN	201900902
NEHAL ASHRAF ELSAYED ELSAYD	201900903
MOHAMED ASHRAF ABDELAZIZ IBRAHIM	201900635
MOHAMED HASSAN ALI AMEN HENDY	201900654
OMAR ABDEL NASSER TAWFIK ADAM	201900517
HESHAM MAHMOUD IBRAHIM	201900942
NOURHAN MOUHAMED RADWAN	201900916

ARCHITECTURE USED IN THE PAPER

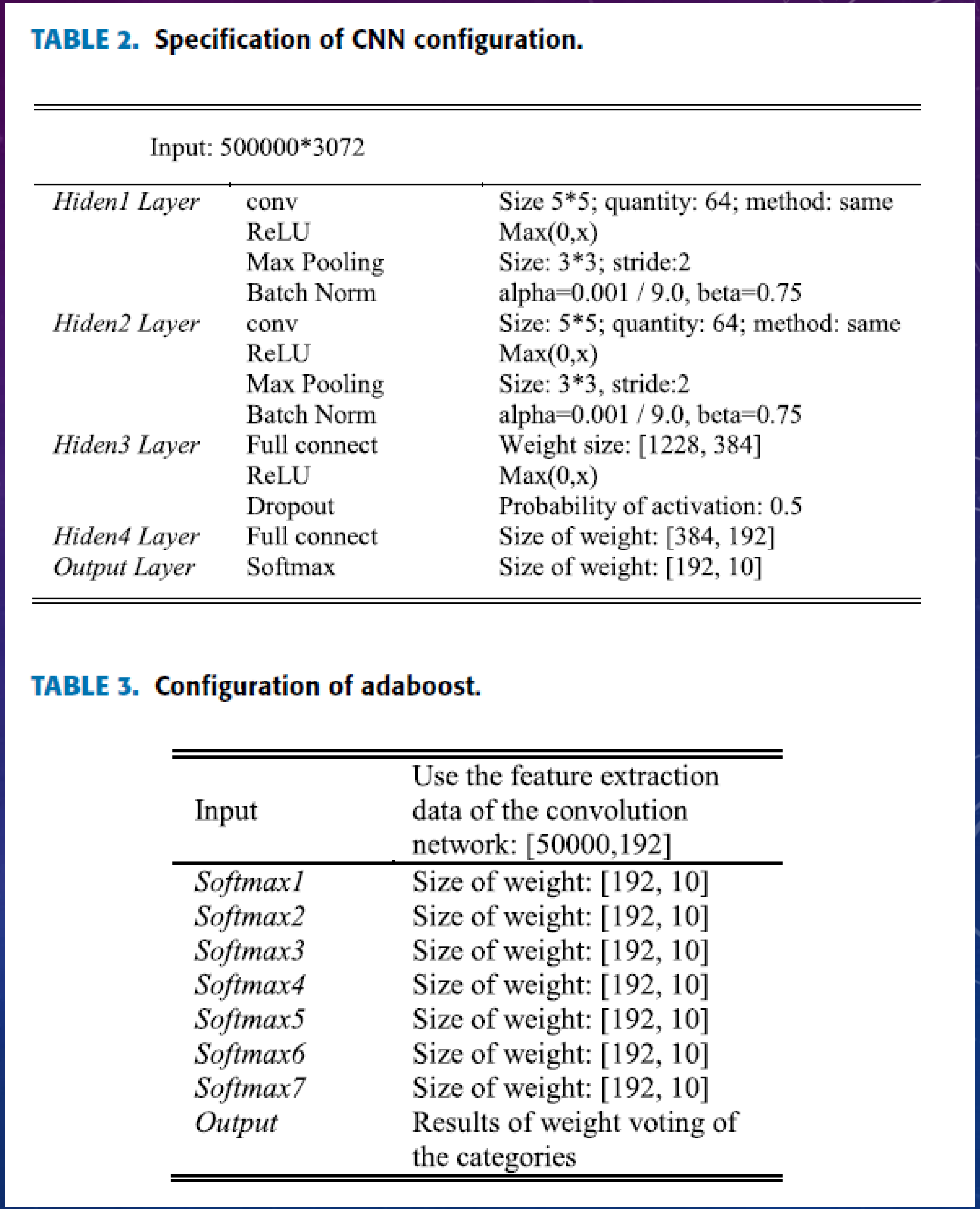


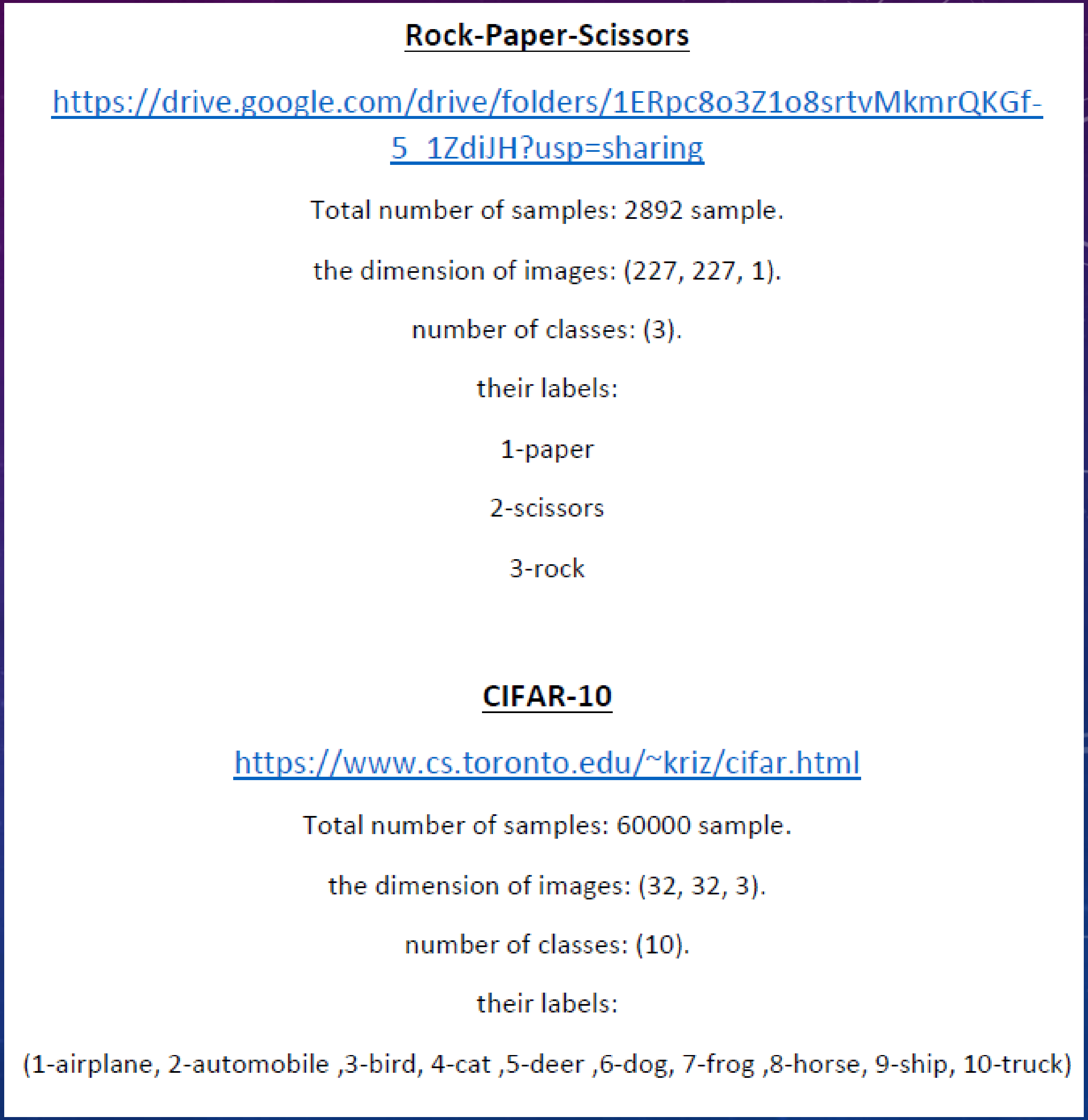
TABLE 2. Specification of CNN configuration.

Input: 500000*3072		
Hiden1 Layer	conv	Size 5*5; quantity: 64; method: same
	ReLU	Max(0,x)
	Max Pooling	Size: 3*3; stride:2
	Batch Norm	alpha=0.001 / 9.0, beta=0.75
Hiden2 Layer	conv	Size: 5*5; quantity: 64; method: same
	ReLU	Max(0,x)
	Max Pooling	Size: 3*3, stride:2
	Batch Norm	alpha=0.001 / 9.0, beta=0.75
Hiden3 Layer	Full connect	Weight size: [1228, 384]
	ReLU	Max(0,x)
	Dropout	Probability of activation: 0.5
Hiden4 Layer	Full connect	Size of weight: [384, 192]
Output Layer	Softmax	Size of weight: [192, 10]

TABLE 3. Configuration of adaboost.

Input	Use the feature extraction data of the convolution network: [50000,192]
Softmax1	Size of weight: [192, 10]
Softmax2	Size of weight: [192, 10]
Softmax3	Size of weight: [192, 10]
Softmax4	Size of weight: [192, 10]
Softmax5	Size of weight: [192, 10]
Softmax6	Size of weight: [192, 10]
Softmax7	Size of weight: [192, 10]
Output	Results of weight voting of the categories

DATASET DETAILS



Rock-Paper-Scissors

https://drive.google.com/drive/folders/1ERpc8o3Z1o8srtvMkmrQKGf-5_1ZdiJH?usp=sharing

Total number of samples: 2892 sample.

the dimension of images: (227, 227, 1).

number of classes: (3).

their labels:

1-paper

2-scissors

3-rock

CIFAR-10

https://www.cs.toronto.edu/~kriz/cifar.html

Total number of samples: 60000 sample.

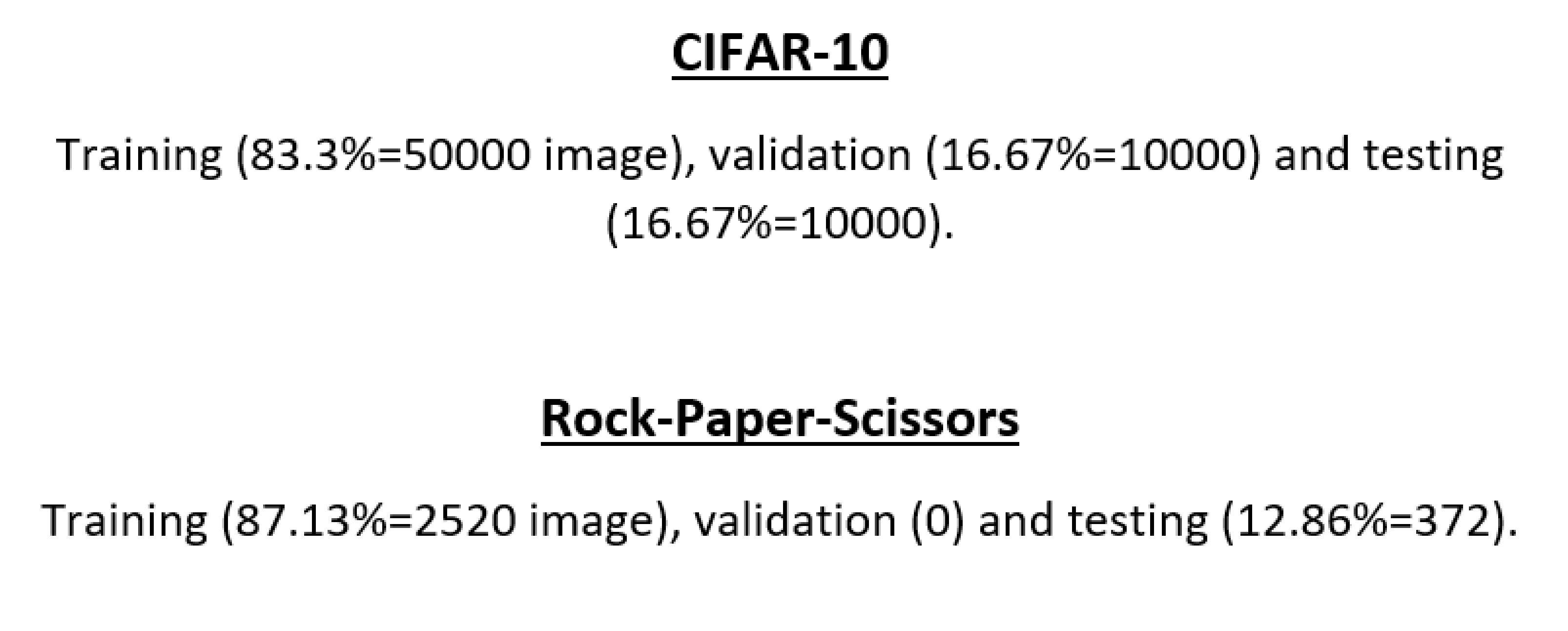
the dimension of images: (32, 32, 3).

number of classes: (10).

their labels:

(1-airplane, 2-automobile ,3-bird, 4-cat ,5-deer ,6-dog, 7-frog ,8-horse, 9-ship, 10-truck)

IMPLEMENTATION DETAILS



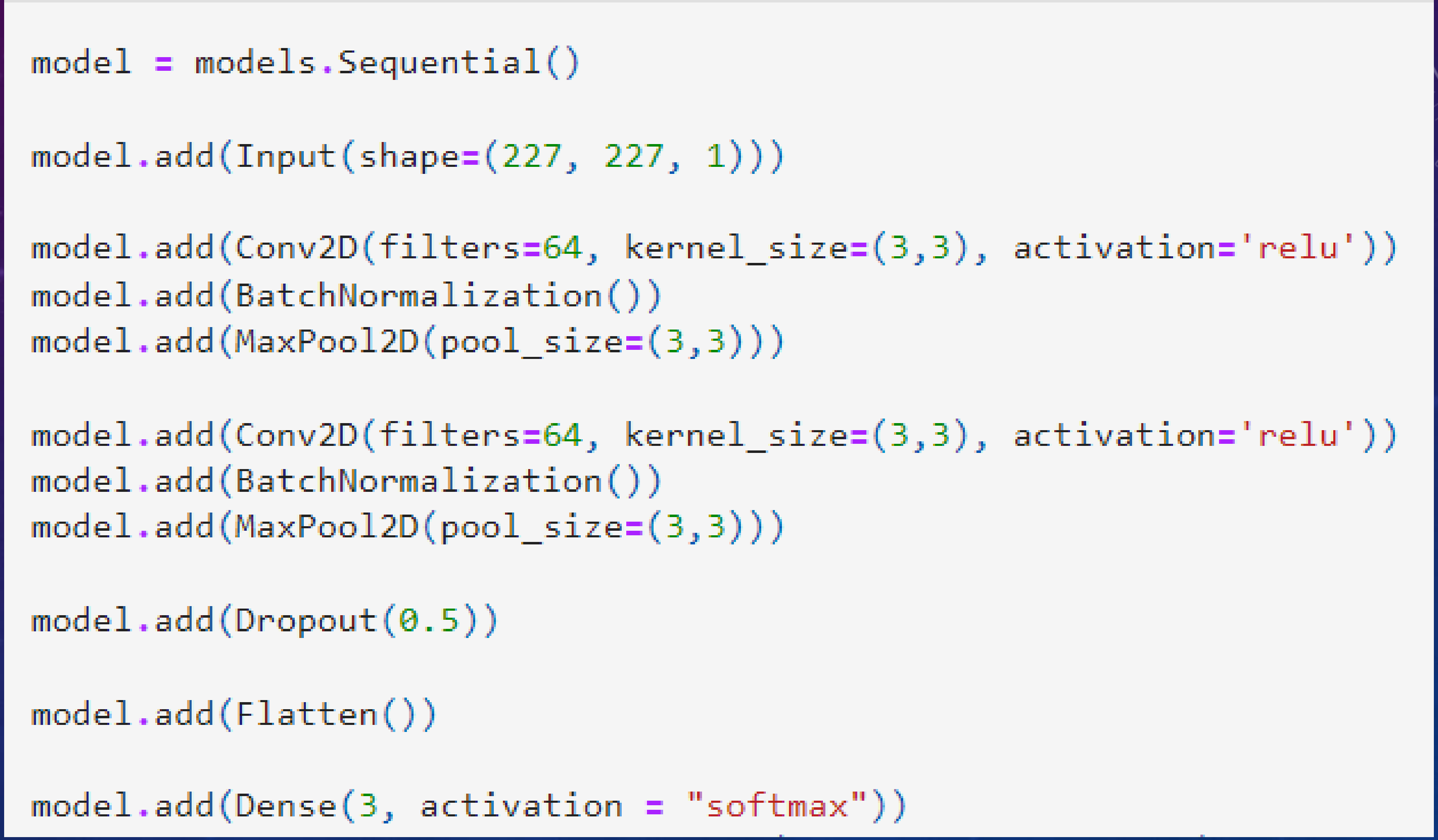
CIFAR-10

Training (83.3%=50000 image), validation (16.67%=10000) and testing (16.67%=10000).

Rock-Paper-Scissors

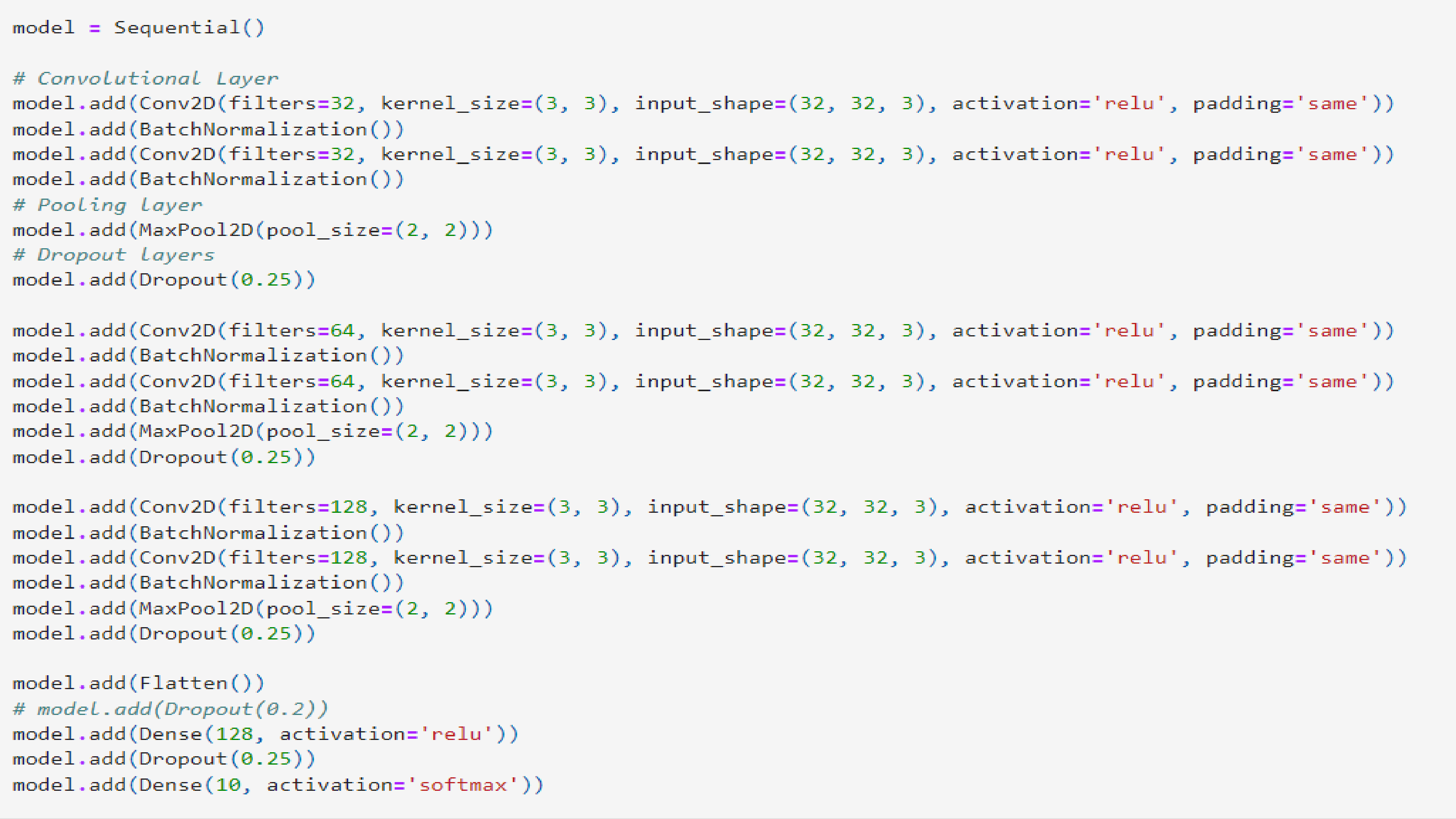
Training (87.13%=2520 image), validation (0) and testing (12.86%=372).

ROCK-PAPER-SCISSORS MODEL



```
model = models.Sequential()

model.add(Input(shape=(227, 227, 1)))

model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(3,3)))

model.add(Conv2D(filters=64, kernel_size=(3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(3,3)))

model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(3, activation = "softmax"))
```

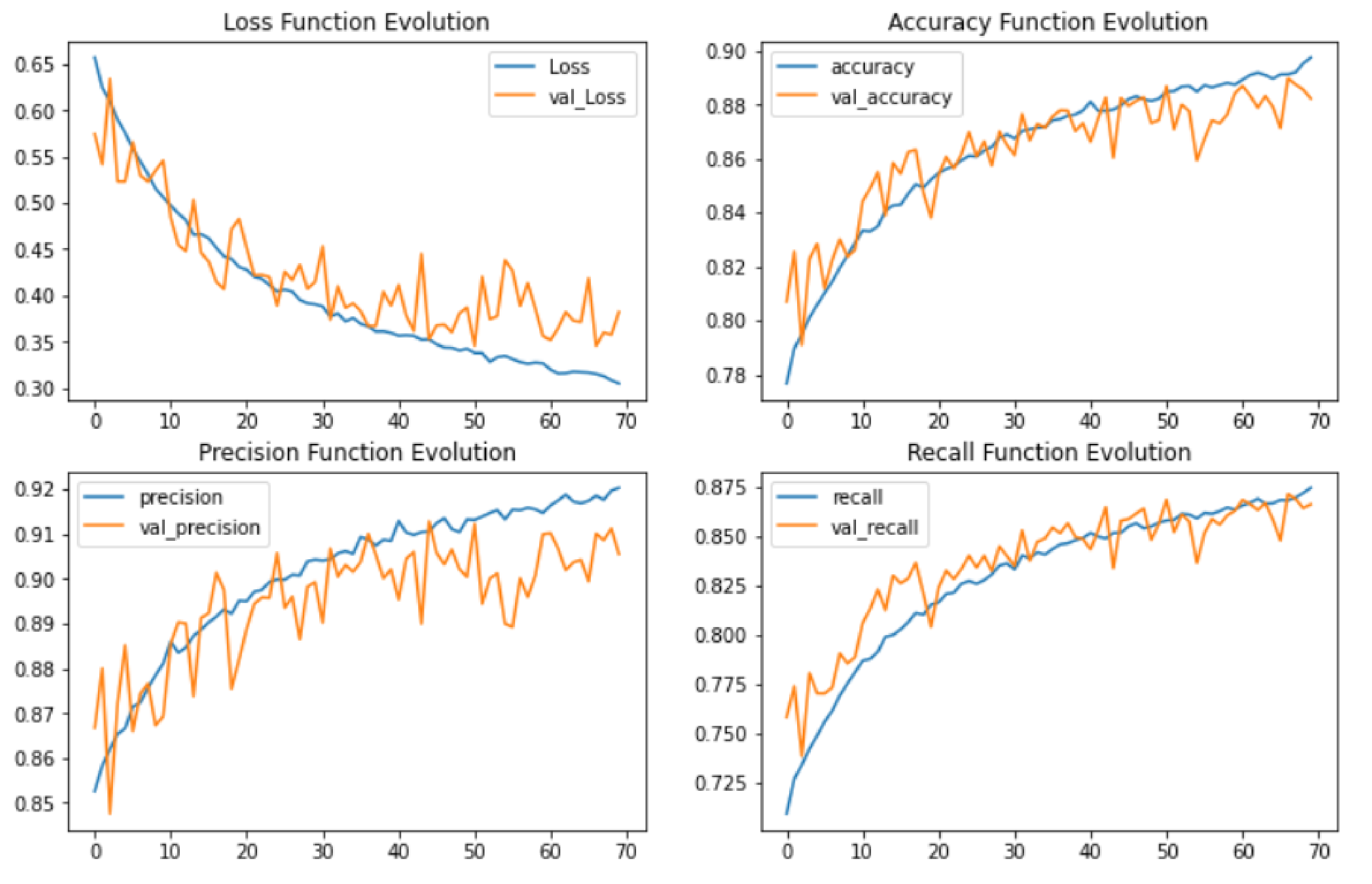
CIFAR-10 MODEL

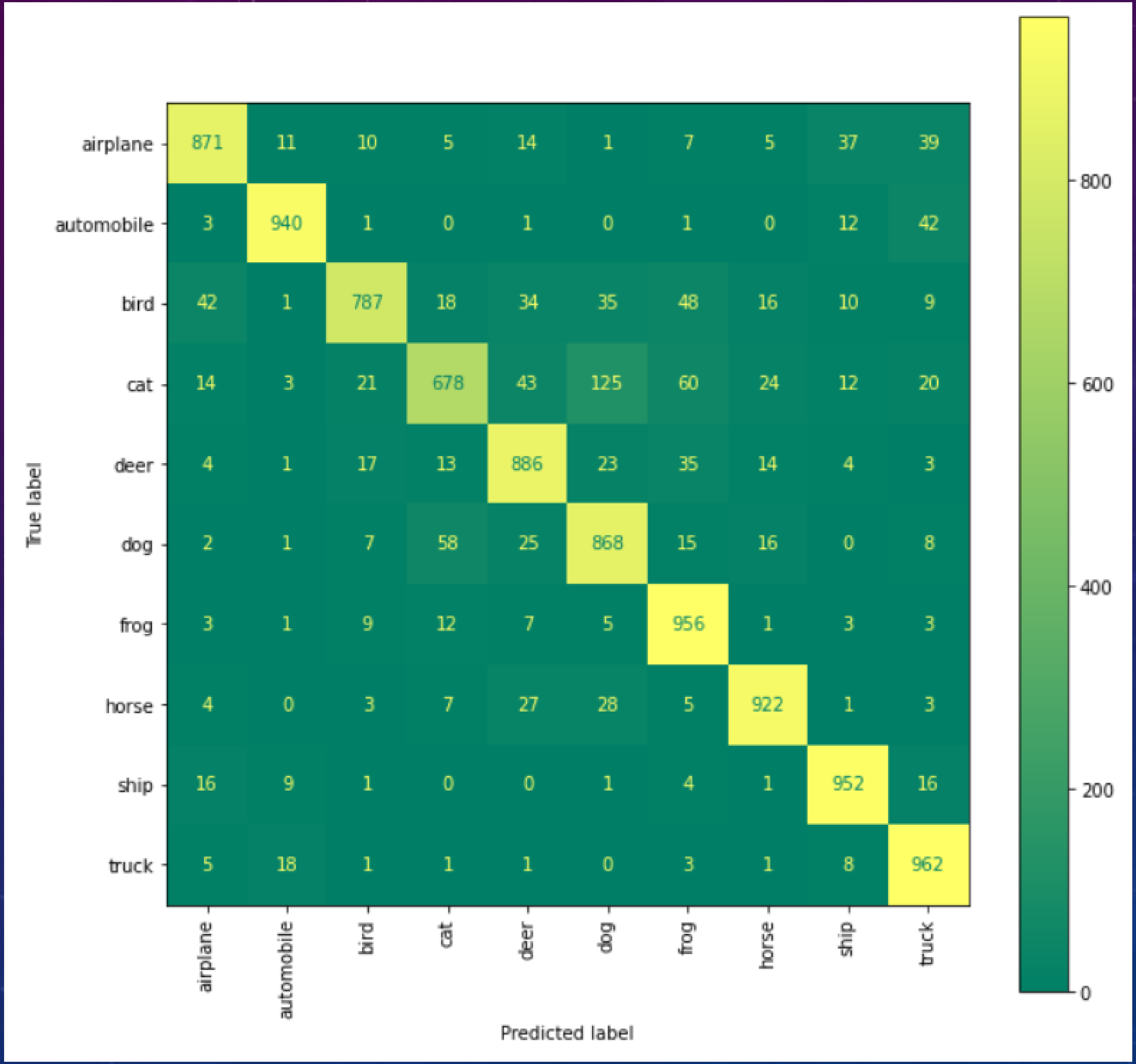


```
model = Sequential()

# Convolutional Layer
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
# Pooling layer
model.add(MaxPool2D(pool_size=(2, 2)))
# Dropout layers
model.add(Dropout(0.25))

model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters=128, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(filters=128, kernel_size=(3, 3), input_shape=(32, 32, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
# model.add(Dropout(0.2))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(10, activation='softmax'))
```

RESULTS AND VISUALIZATION

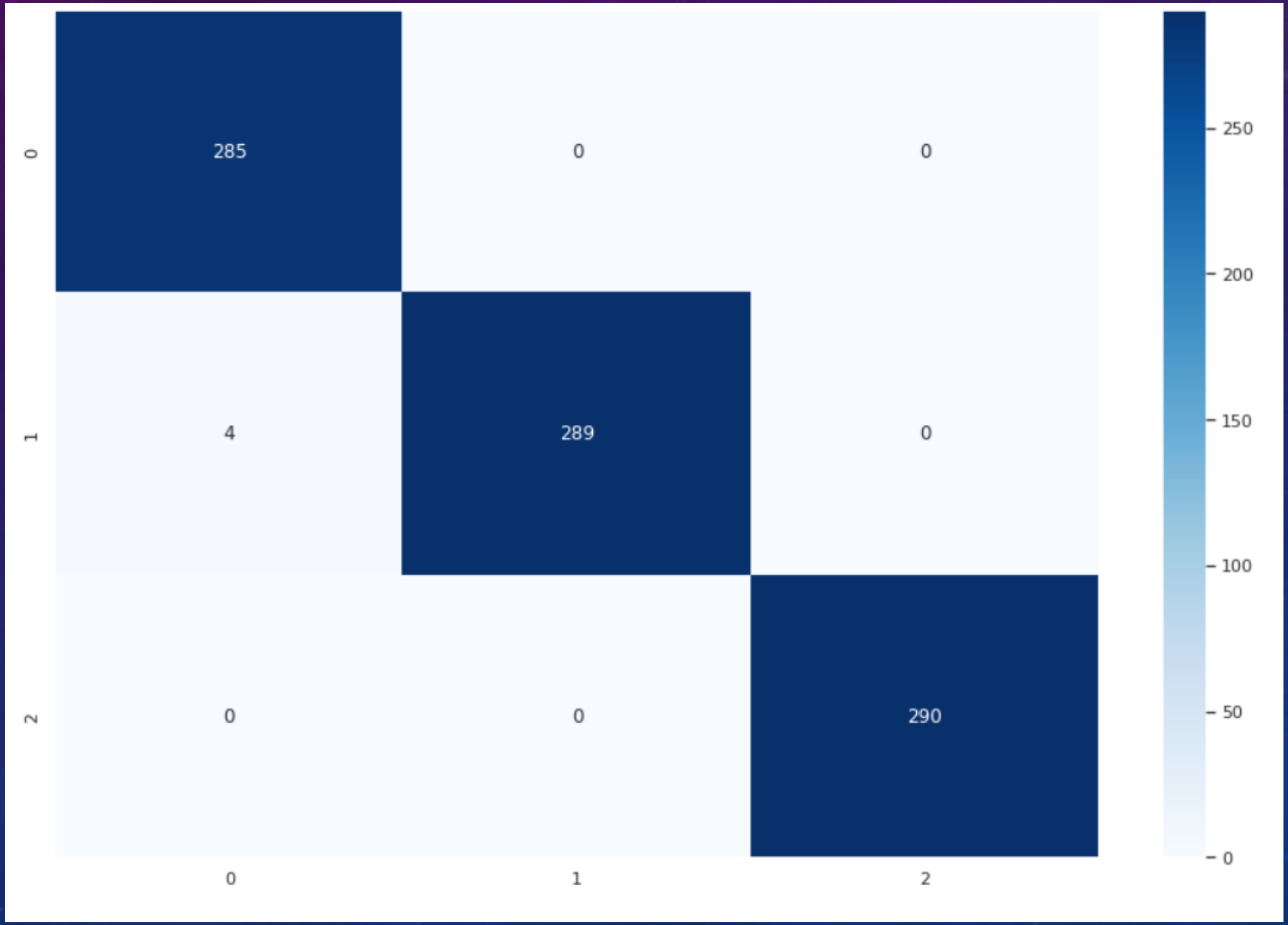
CIFAR-10 RESULT



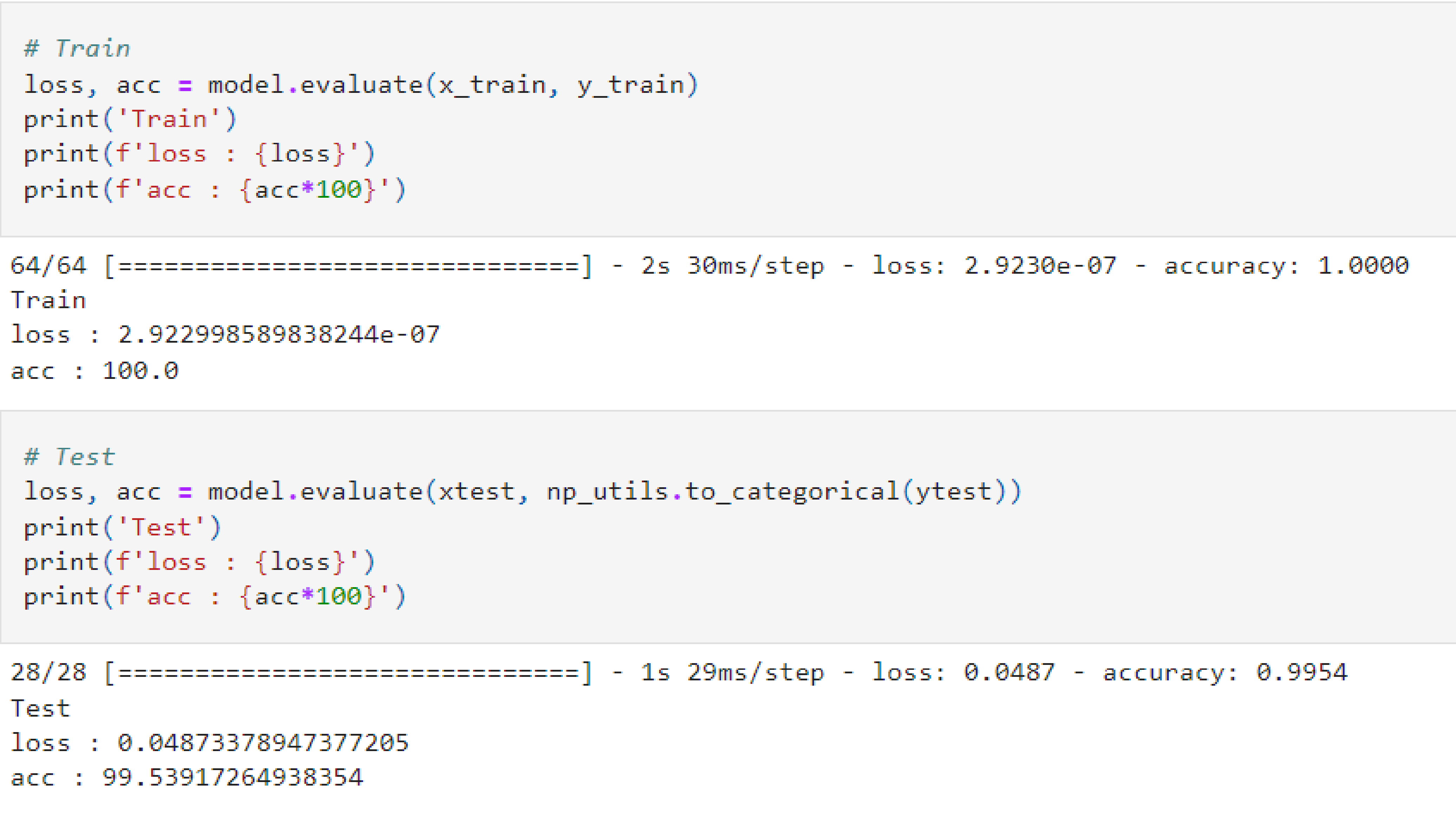


	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	871	11	10	5	14	1	7	5	37	39
automobile	3	940	1	0	1	0	1	0	12	42
bird	42	1	787	18	34	35	48	16	10	9
cat	14	3	21	678	43	125	60	24	12	20
deer	4	1	17	13	886	23	35	14	4	3
dog	2	1	7	58	25	868	15	16	0	8
frog	3	1	9	12	7	5	956	1	3	3
horse	4	0	3	7	27	28	5	922	1	3
ship	16	9	1	0	0	1	4	1	952	16
truck	5	18	1	1	1	0	3	1	8	962

ROCK-PAPER-SCISSORS MODEL RESULT



	0	1	2
0	285	0	0
1	4	289	0
2	0	0	290

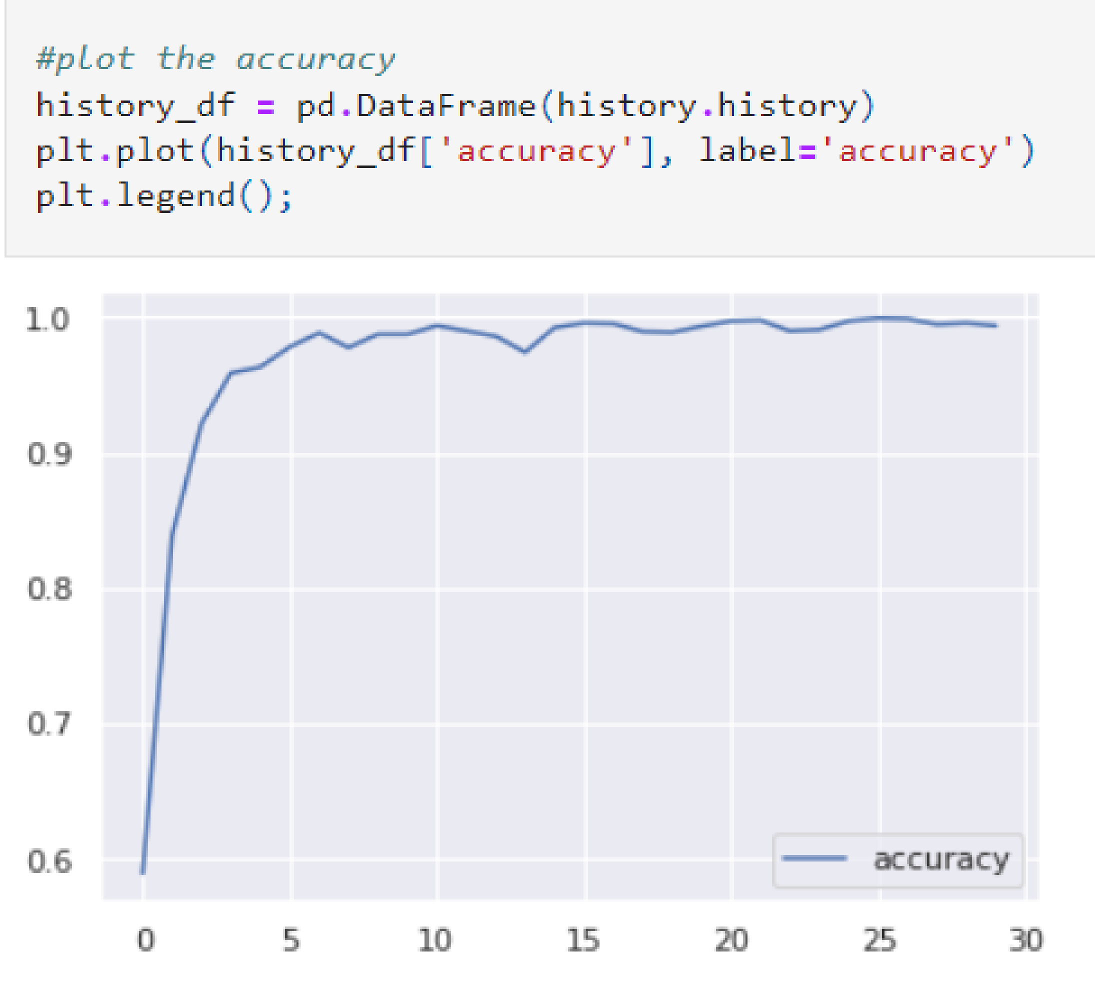


```
# Train
loss, acc = model.evaluate(x_train, y_train)
print('Train')
print(f'loss : {loss}')
print(f'acc : {acc*100}')
```

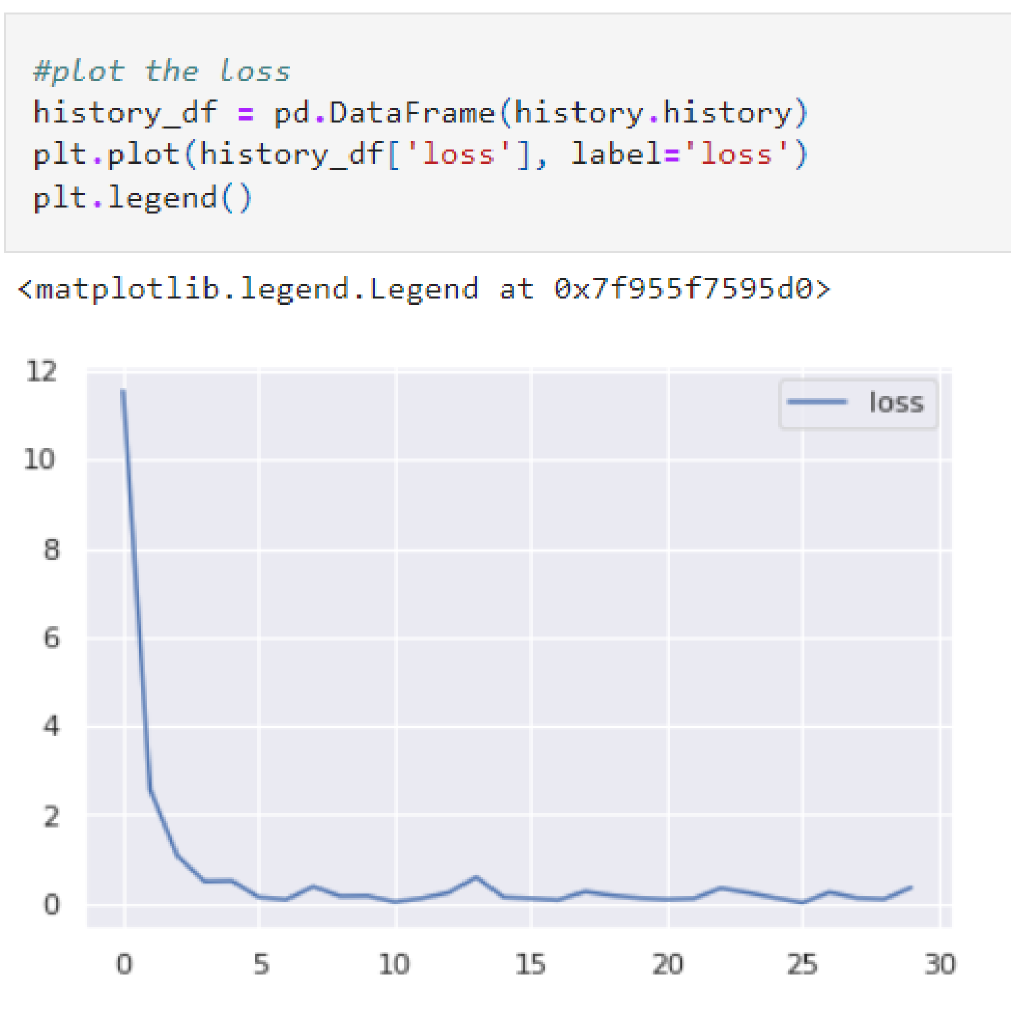
```
64/64 [==============================] - 2s 30ms/step - loss: 2.9230e-07 - accuracy: 1.0000
Train
loss : 2.9229985898382 44e-07
acc : 100.0
```

```
# Test
loss, acc = model.evaluate(xtest, np_utils.to_categorical(ytest))
print('Test')
print(f'loss : {loss}')
print(f'acc : {acc*100}')
```

```
28/28 [==============================] - 1s 29ms/step - loss: 0.0487 - accuracy: 0.9954
Test
loss : 0.04873378947377205
acc : 99.53917264938354
```



```python
#plot the accuracy
history_df = pd.DataFrame(history.history)
plt.plot(history_df['accuracy'], label='accuracy')
plt.legend();
```



```python
#plot the loss
history_df = pd.DataFrame(history.history)
plt.plot(history_df['loss'], label='loss')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f955f7595d0>
```