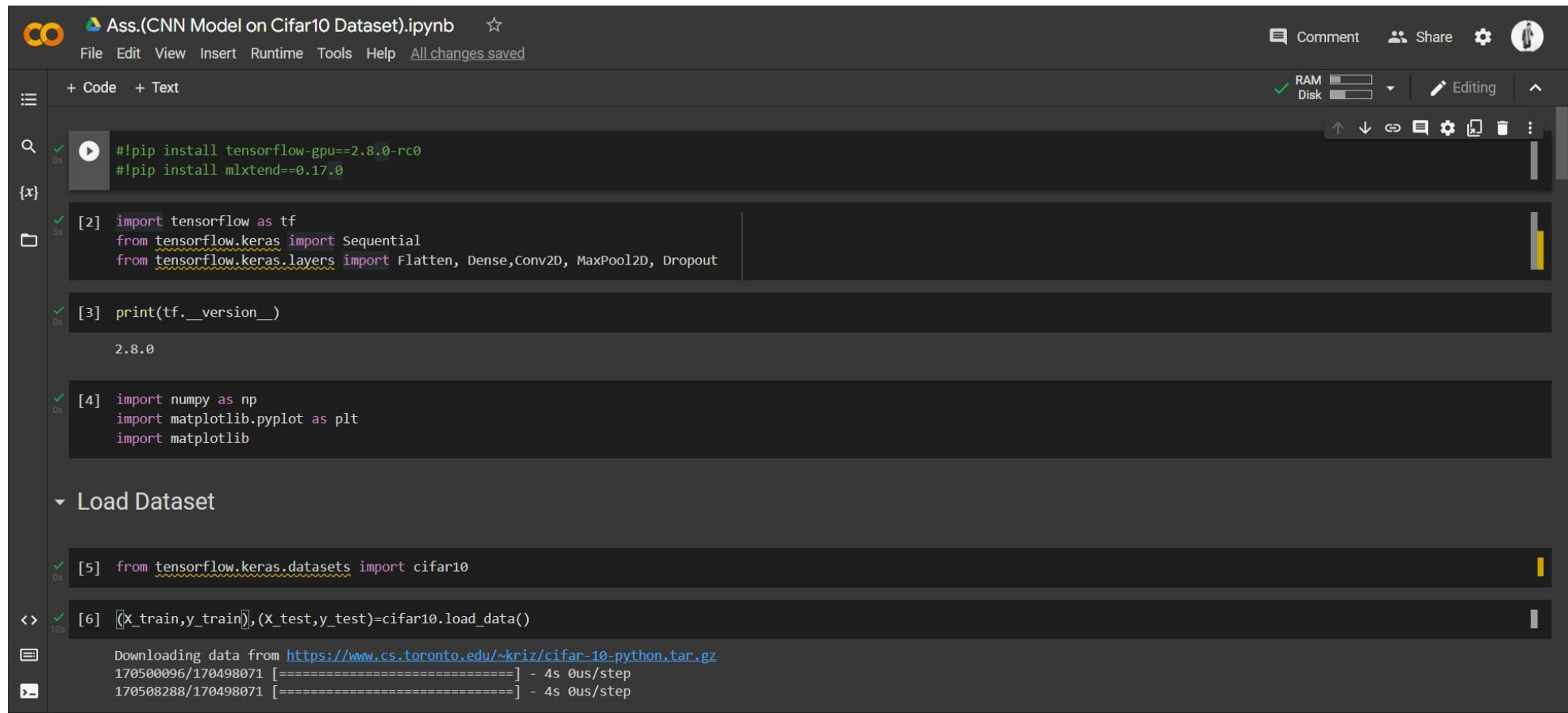


# CNN Model on Cifar10



The screenshot displays a Jupyter Notebook titled "Ass.(CNN Model on Cifar10 Dataset).ipynb". The interface includes a top menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. On the right, there are icons for Comment, Share, and a user profile. Below the menu, a toolbar shows RAM and Disk usage, an "Editing" mode indicator, and various action icons. The notebook contains several code cells:

- Cell 1:** A code cell with a play button icon, containing pip installation commands:

```
#!pip install tensorflow-gpu==2.8.0-rc0
#!pip install mxlxtend==0.17.0
```
- Cell 2:** A code cell with a checkmark icon, containing Keras imports:

```
[2] import tensorflow as tf
    from tensorflow.keras import Sequential
    from tensorflow.keras.layers import Flatten, Dense, Conv2D, MaxPool2D, Dropout
```
- Cell 3:** A code cell with a checkmark icon, containing a print statement and its output:

```
[3] print(tf.__version__)

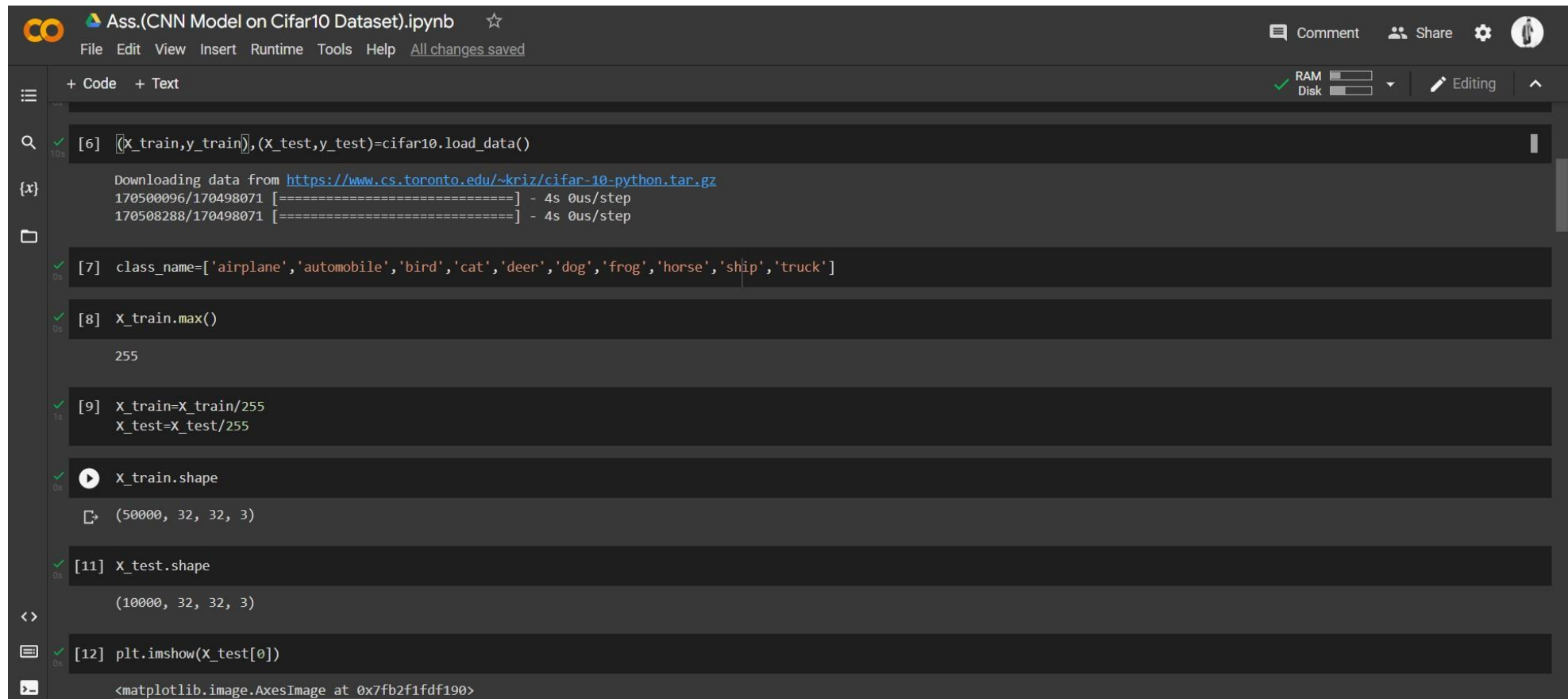
2.8.0
```
- Cell 4:** A code cell with a checkmark icon, containing imports for numpy, matplotlib, and matplotlib.pyplot:

```
[4] import numpy as np
    import matplotlib.pyplot as plt
    import matplotlib
```
- Section Header:** A section titled "Load Dataset" with a downward arrow icon.
- Cell 5:** A code cell with a checkmark icon, containing the import for the CIFAR-10 dataset:

```
[5] from tensorflow.keras.datasets import cifar10
```
- Cell 6:** A code cell with a checkmark icon, containing the data loading command:

```
[6] (X_train, y_train), (X_test, y_test) = cifar10.load_data()
```
- Output:** Below the code cells, the output of the data loading process is shown:

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [=====] - 4s 0us/step
170508288/170498071 [=====] - 4s 0us/step
```



The screenshot shows a Jupyter Notebook titled "Ass.(CNN Model on Cifar10 Dataset).ipynb". The interface includes a top menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. On the right, there are icons for Comment, Share, and a user profile. Below the menu, there are status indicators for RAM and Disk usage, and a button for Editing. The notebook content consists of several code cells, each with a green checkmark icon on the left. The code cells are as follows:

```
[6] [(X_train,y_train),(X_test,y_test)]=cifar10.load_data()

Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [=====] - 4s 0us/step
170508288/170498071 [=====] - 4s 0us/step

[7] class_name=['airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck']

[8] X_train.max()

255

[9] X_train=X_train/255
    X_test=X_test/255

[10] X_train.shape


(50000, 32, 32, 3)

[11] X_test.shape

(10000, 32, 32, 3)

[12] plt.imshow(X_test[0])

<matplotlib.image.AxesImage at 0x7fb2f1fdf190>
```

 Ass.(CNN Model on Cifar10 Dataset).ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

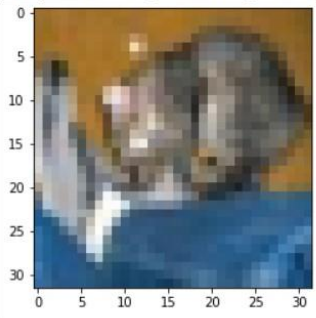
+ Code + Text

Comment Share Settings Profile

RAM Disk Editing

plt.imshow(X\_test[0])

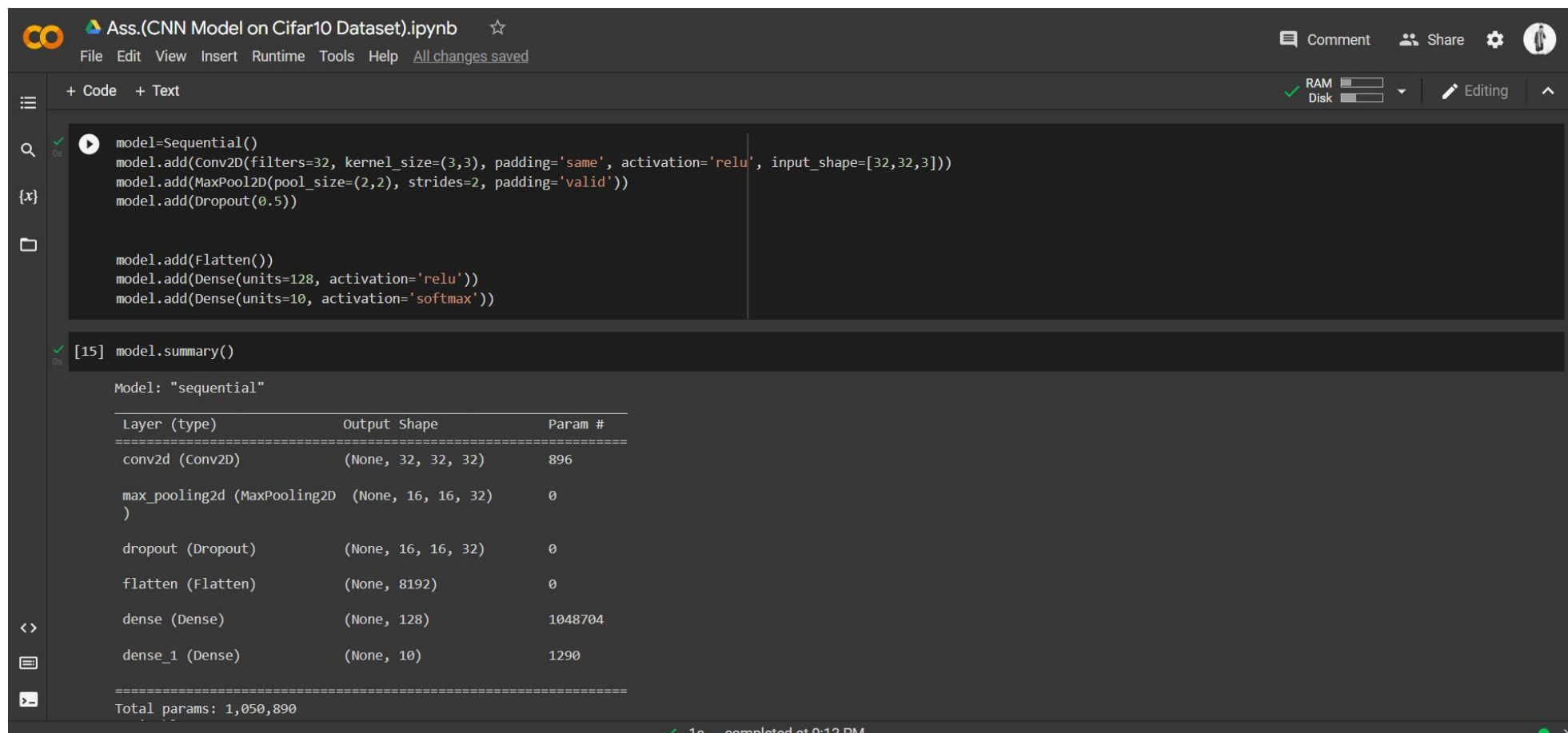
<matplotlib.image.AxesImage at 0x7fb2f1fdf190>



[13] y\_test

array([[3],  
[8],  
[8],  
...,  
[5],  
[1],  
[7]], dtype=uint8)

[14] model=Sequential()  
model.add(Conv2D(filters=32, kernel\_size=(3,3), padding='same', activation='relu', input\_shape=[32,32,3]))



The screenshot shows a Jupyter Notebook titled "Ass.(CNN Model on Cifar10 Dataset).ipynb". The code defines a sequential model with the following layers:

- Conv2D(filters=32, kernel\_size=(3,3), padding='same', activation='relu', input\_shape=[32,32,3])
- MaxPool2D(pool\_size=(2,2), strides=2, padding='valid')
- Dropout(0.5)
- Flatten()
- Dense(units=128, activation='relu')
- Dense(units=10, activation='softmax')

The model summary is displayed below the code:

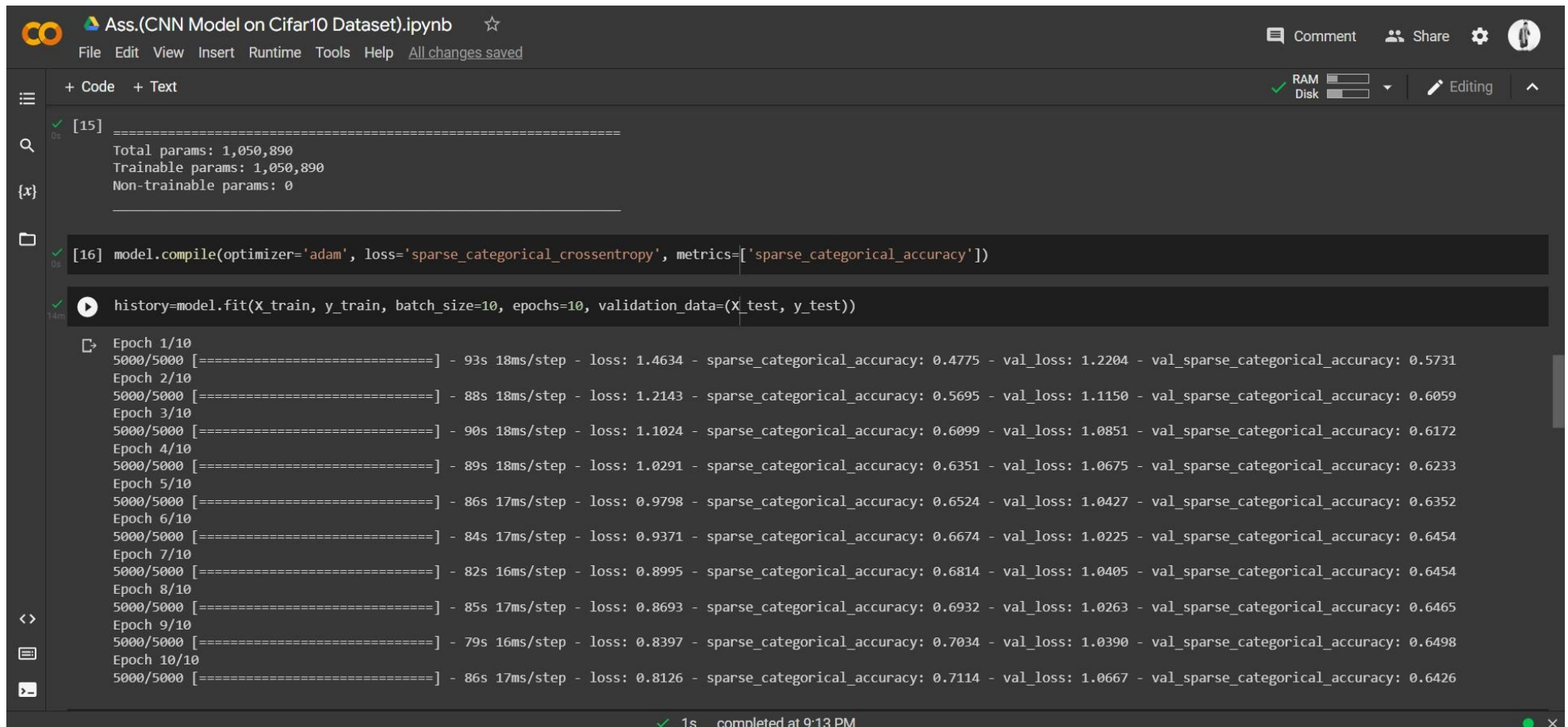
```
[15]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 128)	1048704
dense_1 (Dense)	(None, 10)	1290

Total params: 1,050,890

The notebook interface includes a top bar with the file name, a star icon, and buttons for Comment, Share, and settings. The left sidebar shows the file explorer and code editor tabs. The bottom status bar indicates the execution time as 1s and the completion time as 9:13 PM.



```
Ass.(CNN Model on Cifar10 Dataset).ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
RAM 100%
Disk 100%
Editing

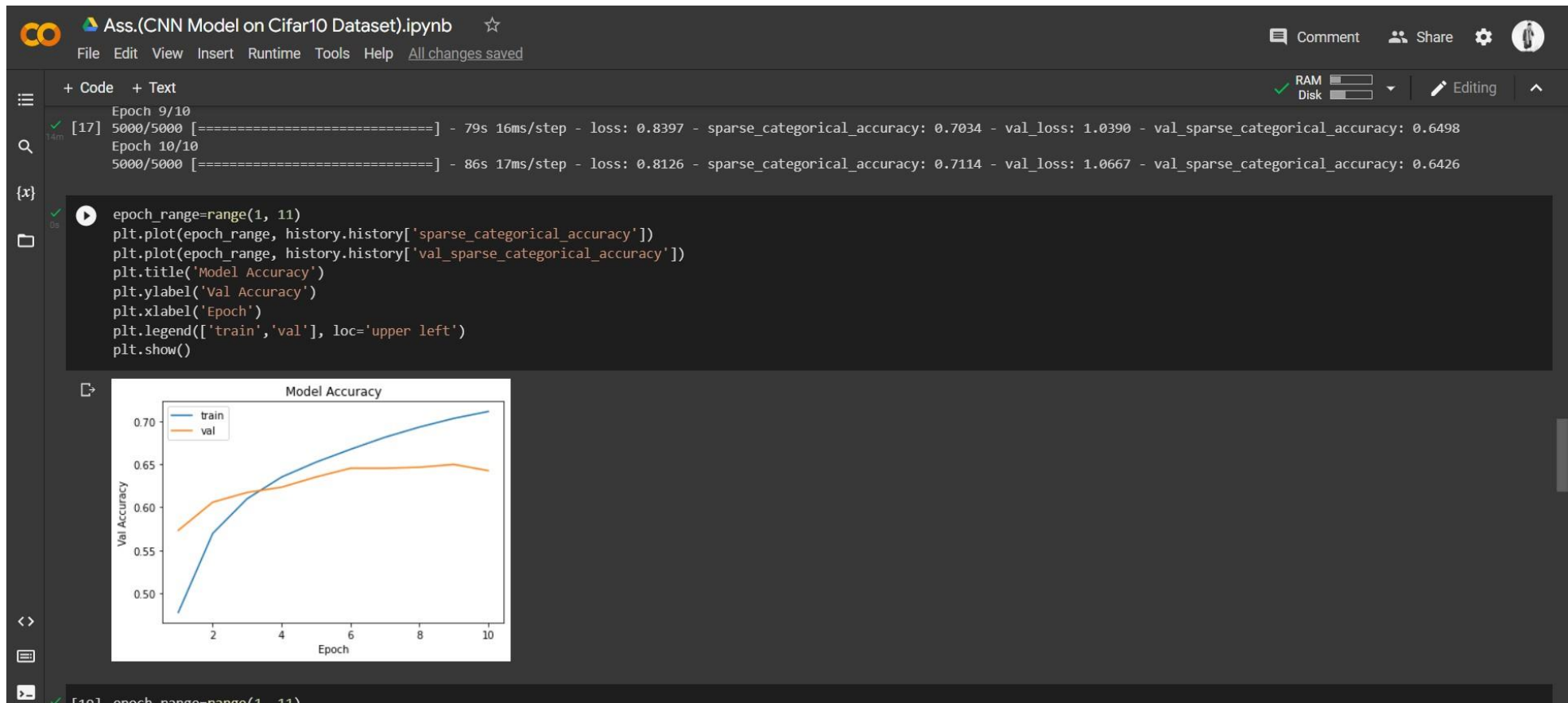
[15] =====
Total params: 1,050,890
Trainable params: 1,050,890
Non-trainable params: 0


[16] model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['sparse_categorical_accuracy'])

[17] history=model.fit(X_train, y_train, batch_size=10, epochs=10, validation_data=(X_test, y_test))

Epoch 1/10
5000/5000 [=====] - 93s 18ms/step - loss: 1.4634 - sparse_categorical_accuracy: 0.4775 - val_loss: 1.2204 - val_sparse_categorical_accuracy: 0.5731
Epoch 2/10
5000/5000 [=====] - 88s 18ms/step - loss: 1.2143 - sparse_categorical_accuracy: 0.5695 - val_loss: 1.1150 - val_sparse_categorical_accuracy: 0.6059
Epoch 3/10
5000/5000 [=====] - 90s 18ms/step - loss: 1.1024 - sparse_categorical_accuracy: 0.6099 - val_loss: 1.0851 - val_sparse_categorical_accuracy: 0.6172
Epoch 4/10
5000/5000 [=====] - 89s 18ms/step - loss: 1.0291 - sparse_categorical_accuracy: 0.6351 - val_loss: 1.0675 - val_sparse_categorical_accuracy: 0.6233
Epoch 5/10
5000/5000 [=====] - 86s 17ms/step - loss: 0.9798 - sparse_categorical_accuracy: 0.6524 - val_loss: 1.0427 - val_sparse_categorical_accuracy: 0.6352
Epoch 6/10
5000/5000 [=====] - 84s 17ms/step - loss: 0.9371 - sparse_categorical_accuracy: 0.6674 - val_loss: 1.0225 - val_sparse_categorical_accuracy: 0.6454
Epoch 7/10
5000/5000 [=====] - 82s 16ms/step - loss: 0.8995 - sparse_categorical_accuracy: 0.6814 - val_loss: 1.0405 - val_sparse_categorical_accuracy: 0.6454
Epoch 8/10
5000/5000 [=====] - 85s 17ms/step - loss: 0.8693 - sparse_categorical_accuracy: 0.6932 - val_loss: 1.0263 - val_sparse_categorical_accuracy: 0.6465
Epoch 9/10
5000/5000 [=====] - 79s 16ms/step - loss: 0.8397 - sparse_categorical_accuracy: 0.7034 - val_loss: 1.0390 - val_sparse_categorical_accuracy: 0.6498
Epoch 10/10
5000/5000 [=====] - 86s 17ms/step - loss: 0.8126 - sparse_categorical_accuracy: 0.7114 - val_loss: 1.0667 - val_sparse_categorical_accuracy: 0.6426



1s completed at 9:13 PM
```



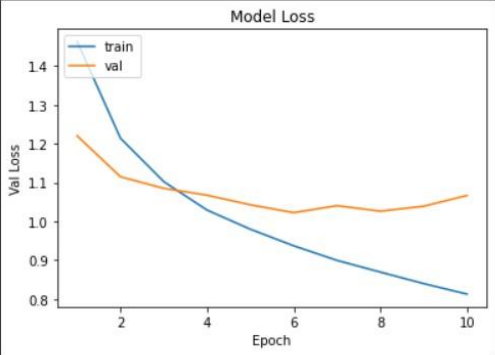
 Ass.(CNN Model on Cifar10 Dataset).ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

RAM  Disk  Editing 


```
[19] epoch_range=range(1, 11)
plt.plot(epoch_range, history.history['loss'])
plt.plot(epoch_range, history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Val Loss')
plt.xlabel('Epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



Epoch	Train Loss	Val Loss
1	1.35	1.22
2	1.22	1.12
3	1.12	1.08
4	1.05	1.05
5	0.98	1.02
6	0.92	1.02
7	0.88	1.05
8	0.85	1.03
9	0.83	1.04
10	0.82	1.05

```
predict_x=model.predict(X_test)
y_pred=np.argmax(predict_x,axis=1)
```

```
[21] y_pred
```

 Ass.(CNN Model on Cifar10 Dataset).ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

RAM  Disk  Editing

✓ [21] y\_pred

array([8, 8, 8, ..., 5, 1, 7])

✓ [22] y\_test

array([[3],  
[8],  
[8],  
...,  
[5],  
[1],  
[7]], dtype=uint8)

✓ [23]

from mlxtend.plotting import plot\_confusion\_matrix  
from sklearn.metrics import confusion\_matrix

✓ [24]

mat=confusion\_matrix(y\_test, y\_pred)

✓ [25]

plot\_confusion\_matrix(mat, figsize=(6,6), show\_normed=True)

<Figure size 432x432 with 1 Axes>  
<matplotlib.axes.\_subplots.AxesSubplot at 0x7fb2eaa22610>

0	683 (0.68)	22 (0.02)	48 (0.05)	40 (0.04)	14 (0.01)	16 (0.02)	16 (0.02)	16 (0.02)	67 (0.07)	78 (0.08)
1	14 (0.01)	760 (0.76)	5 (0.01)	12 (0.01)	0 (0.00)	2 (0.00)	8 (0.01)	5 (0.01)	20 (0.02)	174 (0.17)
2	71 (0.07)	15 (0.01)	418 (0.42)	84 (0.08)	95 (0.10)	139 (0.14)	105 (0.10)	41 (0.04)	16 (0.02)	16 (0.02)



Ass.(CNN Model on Cifar10 Dataset).ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

```
[23] from mlxtend.plotting import plot_confusion_matrix
      from sklearn.metrics import confusion_matrix

[24] mat=confusion_matrix(y_test, y_pred)

plot_confusion_matrix(mat, figsize=(6,6), show_normed=True)
```

<Figure size 432x432 with 1 Axes>  
<matplotlib.axes.\_subplots.AxesSubplot at 0x7fb2eaa22610>

	0	1	2	3	4	5	6	7	8	
0	683 (0.68)	22 (0.02)	48 (0.05)	40 (0.04)	14 (0.01)	16 (0.02)	16 (0.02)	16 (0.02)	67 (0.07)	78 (0.08)
1	14 (0.01)	760 (0.76)	5 (0.01)	12 (0.01)	0 (0.00)	2 (0.00)	8 (0.01)	5 (0.01)	20 (0.02)	174 (0.17)
2	71 (0.07)	15 (0.01)	418 (0.42)	84 (0.08)	95 (0.10)	139 (0.14)	105 (0.10)	41 (0.04)	16 (0.02)	16 (0.02)
3	19 (0.02)	21 (0.02)	46 (0.05)	442 (0.44)	49 (0.05)	251 (0.25)	95 (0.10)	33 (0.03)	21 (0.02)	23 (0.02)
4	34 (0.03)	9 (0.01)	43 (0.04)	104 (0.10)	524 (0.52)	83 (0.08)	107 (0.11)	79 (0.08)	12 (0.01)	5 (0.01)
5	15 (0.01)	5 (0.01)	45 (0.04)	155 (0.15)	43 (0.04)	615 (0.61)	38 (0.04)	58 (0.06)	11 (0.01)	15 (0.01)
6	5 (0.01)	8 (0.01)	32 (0.03)	80 (0.08)	34 (0.03)	47 (0.05)	776 (0.78)	7 (0.01)	9 (0.01)	2 (0.00)
7	11 (0.01)	6 (0.01)	28 (0.03)	58 (0.06)	45 (0.04)	94 (0.09)	14 (0.01)	708 (0.71)	7 (0.01)	29 (0.03)
8	97 (0.10)	52 (0.05)	8 (0.01)	32 (0.03)	5 (0.01)	15 (0.01)	8 (0.01)	9 (0.01)	717 (0.72)	57 (0.06)
9	23 (0.02)	91 (0.09)	9 (0.01)	29 (0.03)	0 (0.00)	13 (0.01)	6 (0.01)	12 (0.01)	34 (0.03)	783 (0.78)

1s completed at 9:13 PM