```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
import numpy as np
```

```
(X_train, y_train), (X_test,y_test) = datasets.cifar10.load_data()
X_train.shape
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 4s 0us/step
(50000, 32, 32, 3)
```

```
X_test.shape
```

```
(10000, 32, 32, 3)
```

```
y_train.shape
```

```
(50000, 1)
```

```
y_train[:5]
```

```
array([[6],
       [9],
       [9],
       [4],
       [1]], dtype=uint8)
```

```
y_train=y_train.reshape(-1,)
y_train[:5]
```

```
array([6, 9, 9, 4, 1], dtype=uint8)
```

```
y_test=y_test.reshape(-1)
```

classes=["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"]

```
X_train=X_train/255.0
X_test=X_test/255.0
```

```
cnn = models.Sequential([
    layers.Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(filters=64, kernel_size=(3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),

    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])
```

```
cnn.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
cnn.fit(X_train, y_train, epochs=10)
```

```
Epoch 1/10
1563/1563 [==============================] - 74s 47ms/step - loss: 1.4715 - accuracy: 0.4746
Epoch 2/10
1563/1563 [==============================] - 71s 46ms/step - loss: 1.1128 - accuracy: 0.6111
Epoch 3/10
1563/1563 [==============================] - 74s 47ms/step - loss: 0.9891 - accuracy: 0.6550
Epoch 4/10
1563/1563 [==============================] - 71s 45ms/step - loss: 0.9130 - accuracy: 0.6831
Epoch 5/10
1563/1563 [==============================] - 72s 46ms/step - loss: 0.8471 - accuracy: 0.7067
Epoch 6/10
```

```
1563/1563 [==============================] - 71s 46ms/step - loss: 0.7969 - accuracy: 0.7213
Epoch 7/10
1563/1563 [==============================] - 72s 46ms/step - loss: 0.7492 - accuracy: 0.7405
Epoch 8/10
1563/1563 [==============================] - 70s 45ms/step - loss: 0.7056 - accuracy: 0.7539
Epoch 9/10
1563/1563 [==============================] - 71s 46ms/step - loss: 0.6679 - accuracy: 0.7647
Epoch 10/10
1563/1563 [==============================] - 71s 46ms/step - loss: 0.6292 - accuracy: 0.7790
<keras.callbacks.History at 0x7f12f3f1e020>
```

```
cnn.evaluate(X_test,y_test)
```

```
313/313 [==============================] - 4s 12ms/step - loss: 0.9143 - accuracy: 0.6996
[0.9143370985984802, 0.6995999813079834]
```

```
y_pred = cnn.predict(X_test)
y_pred[:5]
```

```
313/313 [==============================] - 5s 14ms/step
array([[2.4866988e-04, 5.8901311e-05, 3.2999991e-03, 7.4168807e-01,
        7.6619821e-04, 4.1707762e-02, 1.1600843e-01, 8.6237542e-06,
        9.5880494e-02, 3.3277879e-04],
       [5.1231316e-04, 1.1186218e-02, 1.6227300e-06, 1.5418141e-06,
        9.2761667e-08, 3.2809743e-07, 8.2088794e-11, 2.5798197e-10,
        9.8727310e-01, 1.0247014e-03],
       [4.0158039e-01, 3.7449601e-01, 9.0038992e-04, 1.5566789e-02,
        1.7654404e-02, 1.2133983e-03, 1.3520199e-04, 2.8759893e-03,
        1.8019880e-01, 5.3785765e-03],
       [9.2160785e-01, 1.4405651e-03, 8.6524747e-03, 2.4899506e-04,
        8.2117843e-04, 4.9398677e-06, 6.7698420e-07, 1.8228039e-06,
        6.7072809e-02, 1.4872088e-04],
       [3.4482905e-06, 1.4903485e-05, 7.1817990e-03, 3.8286109e-02,
        9.9152215e-02, 6.0704304e-04, 8.5471010e-01, 4.6863684e-07,
        4.3715892e-05, 1.4918402e-07]], dtype=float32)
```

```
y_classes = [np.argmax(element) for element in y_pred]
y_classes[:5]
```

```
[3, 8, 0, 0, 6]
```

```
classes[y_classes[3]]
```

```
'airplane'
```