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

Double-click (or enter) to edit

```
(train_images,train_labels),(test_images,test_labels)=datasets.cifar10.load_data()
```

```
print('x_tain: ', train_images.shape)
print('y_tain: ', train_labels.shape)
print('x_test: ', test_images.shape)
print('y_test: ', test_labels.shape)
```

```
x_tain:  (50000, 32, 32, 3)
y_tain:  (50000, 1)
x_test:  (10000, 32, 32, 3)
y_test:  (10000, 1)
```

```
pd.DataFrame(train_images[1])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-50f17b396f26> in <cell line: 1>()
----> 1 pd.DataFrame(train_images[1])

  2 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/internals/construction.py in
_prep_ndarraylike(values, copy)
    581         values = values.reshape((values.shape[0], 1))
    582     elif values.ndim != 2:
--> 583         raise ValueError(f"Must pass 2-d input. shape={values.shape}")
    584 
    585     return values

ValueError: Must pass 2-d input. shape=(32, 32, 3)
```

SEARCH STACK OVERFLOW

```
train_images=train_images/255
test_images=test_images/255
```

Double-click (or enter) to edit

```
model=models.Sequential()
model.add(layers.Flatten(input_shape=(32,32,3)))
# model.add(layers.Dense(2048,activation='relu'))
# model.add(layers.Dense(1024,activation='relu'))
model.add(layers.Dense(512,activation='relu'))
model.add(layers.Dense(128,activation='relu'))
model.add(layers.Dense(32,activation='relu'))
model.add(layers.Dense(16,activation='relu'))
model.add(layers.Dense(10,activation='softmax'))
```

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

```
model.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 flatten_7 (Flatten)         (None, 3072)              0         
                                                                 
 dense_35 (Dense)            (None, 512)               1573376   
                                                                 
 dense_36 (Dense)            (None, 128)               65664     
                                                                 
 dense_37 (Dense)            (None, 32)                4128      
                                                                 
 dense_38 (Dense)            (None, 16)                528       
 dense_39 (Dense)            (None, 10)                170

=================================================================
Total params: 1,643,866
Trainable params: 1,643,866
Non-trainable params: 0
_________________________________________________________________
```

```
h = model.fit(train_images,train_labels, epochs=50, validation_data = (test_images,test_labels))
```

```
Epoch 1/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1967 - accuracy: 0.5691 - val_loss: 1.4360 - val_accuracy: 0.
Epoch 2/50
1563/1563 [==============================] - 9s 5ms/step - loss: 1.1872 - accuracy: 0.5718 - val_loss: 1.4495 - val_accuracy: 0.
Epoch 3/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1801 - accuracy: 0.5770 - val_loss: 1.4978 - val_accuracy: 0.
Epoch 4/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1700 - accuracy: 0.5791 - val_loss: 1.4685 - val_accuracy: 0.
Epoch 5/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1658 - accuracy: 0.5807 - val_loss: 1.4681 - val_accuracy: 0.
Epoch 6/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1572 - accuracy: 0.5843 - val_loss: 1.4627 - val_accuracy: 0.
Epoch 7/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1515 - accuracy: 0.5881 - val_loss: 1.5164 - val_accuracy: 0.
Epoch 8/50
1563/1563 [==============================] - 10s 7ms/step - loss: 1.1476 - accuracy: 0.5865 - val_loss: 1.4680 - val_accuracy: 0
Epoch 9/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1367 - accuracy: 0.5918 - val_loss: 1.4628 - val_accuracy: 0.
Epoch 10/50
1563/1563 [==============================] - 9s 5ms/step - loss: 1.1324 - accuracy: 0.5921 - val_loss: 1.5246 - val_accuracy: 0.
Epoch 11/50
1563/1563 [==============================] - 9s 5ms/step - loss: 1.1304 - accuracy: 0.5917 - val_loss: 1.4706 - val_accuracy: 0.
Epoch 12/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1227 - accuracy: 0.5956 - val_loss: 1.4956 - val_accuracy: 0.
Epoch 13/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1143 - accuracy: 0.5999 - val_loss: 1.4866 - val_accuracy: 0.
Epoch 14/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.1107 - accuracy: 0.5995 - val_loss: 1.5244 - val_accuracy: 0.
Epoch 15/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1066 - accuracy: 0.6024 - val_loss: 1.4661 - val_accuracy: 0.
Epoch 16/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1001 - accuracy: 0.6026 - val_loss: 1.5276 - val_accuracy: 0.
Epoch 17/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0976 - accuracy: 0.6057 - val_loss: 1.4855 - val_accuracy: 0.
Epoch 18/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0922 - accuracy: 0.6061 - val_loss: 1.5227 - val_accuracy: 0.
Epoch 19/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0824 - accuracy: 0.6097 - val_loss: 1.5593 - val_accuracy: 0.
Epoch 20/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0775 - accuracy: 0.6153 - val_loss: 1.5244 - val_accuracy: 0.
Epoch 21/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0749 - accuracy: 0.6142 - val_loss: 1.5305 - val_accuracy: 0.
Epoch 22/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0708 - accuracy: 0.6146 - val_loss: 1.5456 - val_accuracy: 0.
Epoch 23/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0701 - accuracy: 0.6155 - val_loss: 1.5582 - val_accuracy: 0.
Epoch 24/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0598 - accuracy: 0.6188 - val_loss: 1.5214 - val_accuracy: 0.
Epoch 25/50
1563/1563 [==============================] - 9s 5ms/step - loss: 1.0531 - accuracy: 0.6219 - val_loss: 1.5532 - val_accuracy: 0.
Epoch 26/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0525 - accuracy: 0.6208 - val_loss: 1.5284 - val_accuracy: 0.
Epoch 27/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0530 - accuracy: 0.6213 - val_loss: 1.5290 - val_accuracy: 0.
Epoch 28/50
1563/1563 [==============================] - 9s 6ms/step - loss: 1.0445 - accuracy: 0.6239 - val_loss: 1.5136 - val_accuracy: 0.
Epoch 29/50
```

```
score = model.evaluate(test_images,test_labels)
print("test loss :", score[0])
print("test accuracy :", score[1])
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.1304 - accuracy: 0.9620
test loss : 0.1303500086069107
test accuracy : 0.9620000123977661
```

```
model_name="file.h5"
model.save(model_name,save_format='h5')
```

```
loaded_model = tf.keras.models.load_model(model_name)
```

```
predictions_one_hot = loaded_model.predict([test_images])
```

```
313/313 [==============================] - 1s 1ms/step
```

```
print("predications one hot :", predictions_one_hot.shape)
```

```
predications one hot : (10000, 10)
```

```
predictions=np.argmax(predictions_one_hot, axis=1)
pd.DataFrame(predictions)
```

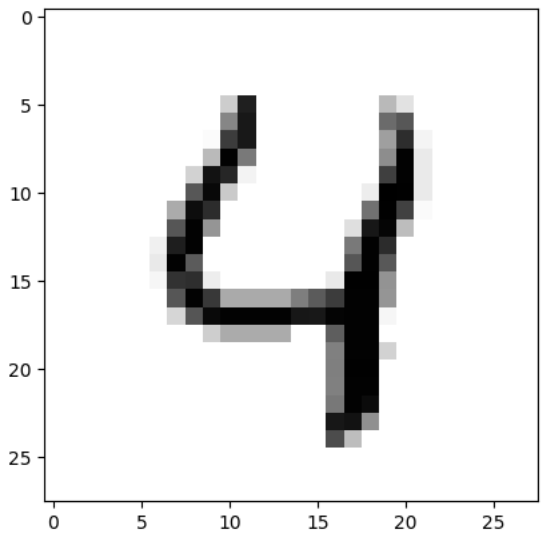
	0
0	7
1	2
2	1
3	0
4	4
...	...
9995	2
9996	3
9997	4
9998	5
9999	6

10000 rows × 1 columns

```
print(predictions[4])
```

```
4
```

```
plt.imshow(test_images[4].reshape((28,28)),cmap=plt.cm.binary)
plt.show()
```


0
5
10
15
20
25
0
5
10
15
20
25