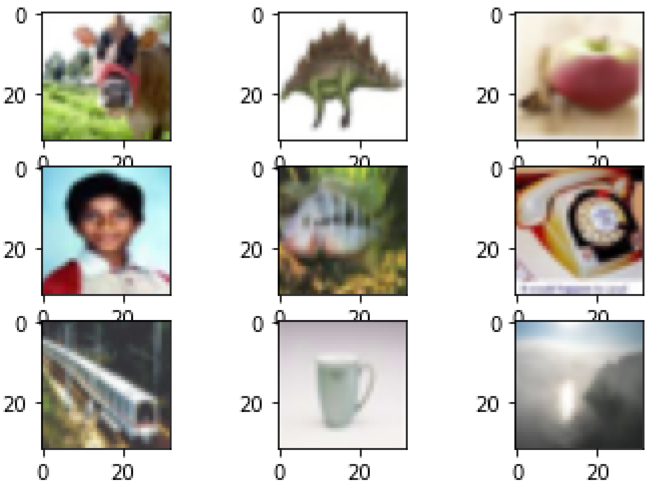
```
from keras.datasets import cifar100
(X_train,y_train),(X_test,y_test)=cifar100.load_data()
import matplotlib.pyplot as plt
for i in range(9):
 plt.subplot(330+i+1)
 plt.imshow(X_train[i])
plt.show()
```



```
X = X_test
X_train = X_train.astype('float')
X_test = X_test.astype('float')
X_train/=255
X_test/=255
```

```
from keras.utils import np_utils
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
```

```
from keras.models import Sequential
from keras.layers import Dense,Activation,Dropout,Flatten
model = Sequential()
model.add(Flatten(input_shape=(32,32,3)))
model.add(Dropout(0.2))
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(100,activation="softmax"))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 3072)              0

 dropout (Dropout)           (None, 3072)              0
 dense (Dense)              (None, 512)               1573376

 dropout_1 (Dropout)        (None, 512)               0

 dense_1 (Dense)            (None, 512)               262656

 dropout_2 (Dropout)        (None, 512)               0

 dense_2 (Dense)            (None, 100)               51300

=================================================================
Total params: 1,887,332
Trainable params: 1,887,332
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.optimizers import SGD
opt = SGD(lr = 0.001, momentum = 0.9)
model.compile(loss = 'categorical_crossentropy', optimizer = opt , metrics = ['accuracy
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/gradient_descent.py:102: Us
  super(SGD, self).__init__(name, **kwargs)
```

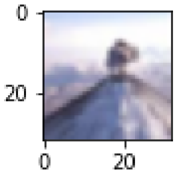
```
history = model.fit(X_train,y_train,batch_size=128,epochs=50,verbose=1,validation_data=
```

```
Epoch 1/50
391/391 [==============================] - 7s 10ms/step - loss: 4.5611 - accuracy
Epoch 2/50
391/391 [==============================] - 3s 9ms/step - loss: 4.3560 - accuracy:
Epoch 3/50
391/391 [==============================] - 3s 8ms/step - loss: 4.1774 - accuracy:
Epoch 4/50
391/391 [==============================] - 3s 8ms/step - loss: 4.0791 - accuracy:
Epoch 5/50
391/391 [==============================] - 3s 8ms/step - loss: 4.0016 - accuracy:
Epoch 6/50
391/391 [==============================] - 3s 8ms/step - loss: 3.9416 - accuracy:
Epoch 7/50
391/391 [==============================] - 3s 8ms/step - loss: 3.8898 - accuracy:
Epoch 8/50
391/391 [==============================] - 3s 8ms/step - loss: 3.8455 - accuracy:
Epoch 9/50
391/391 [==============================] - 3s 8ms/step - loss: 3.8082 - accuracy:
Epoch 10/50
391/391 [==============================] - 3s 8ms/step - loss: 3.7764 - accuracy:
Epoch 11/50
391/391 [==============================] - 3s 8ms/step - loss: 3.7492 - accuracy:
Epoch 12/50
391/391 [==============================] - 3s 8ms/step - loss: 3.7231 - accuracy:
Epoch 13/50
391/391 [==============================] - 3s 8ms/step - loss: 3.6988 - accuracy:
Epoch 14/50
391/391 [==============================] - 3s 8ms/step - loss: 3.6767 - accuracy:
Epoch 15/50
391/391 [==============================] - 3s 8ms/step - loss: 3.6571 - accuracy:
Epoch 16/50
```

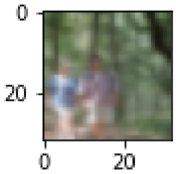
```
391/391 [==============================] - 4s 10ms/step - loss: 3.6371 - accuracy
Epoch 17/50
391/391 [==============================] - 4s 11ms/step - loss: 3.6222 - accuracy
Epoch 18/50
391/391 [==============================] - 3s 8ms/step - loss: 3.6023 - accuracy:
Epoch 19/50
391/391 [==============================] - 3s 8ms/step - loss: 3.5894 - accuracy:
Epoch 20/50
391/391 [==============================] - 3s 8ms/step - loss: 3.5682 - accuracy:
Epoch 21/50
391/391 [==============================] - 3s 9ms/step - loss: 3.5535 - accuracy:
Epoch 22/50
391/391 [==============================] - 3s 8ms/step - loss: 3.5372 - accuracy:
Epoch 23/50
391/391 [==============================] - 3s 9ms/step - loss: 3.5259 - accuracy:
Epoch 24/50
391/391 [==============================] - 3s 8ms/step - loss: 3.5119 - accuracy:
Epoch 25/50
391/391 [==============================] - 3s 8ms/step - loss: 3.4951 - accuracy:
Epoch 26/50
391/391 [==============================] - 3s 8ms/step - loss: 3.4888 - accuracy:
Epoch 27/50
391/391 [==============================] - 3s 8ms/step - loss: 3.4669 - accuracy:
Epoch 28/50
391/391 [==============================] - 3s 8ms/step - loss: 3.4591 - accuracy:
```

```
model.save('final_cifar100_ann.h5')
```

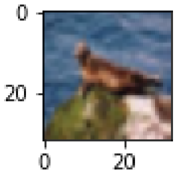
```
import numpy as np
y_pred=model.predict(X_test)
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(X[i])
  plt.show()
  print(np.round(y_pred[i]))
```



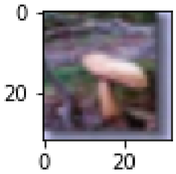
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



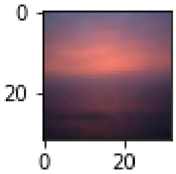
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



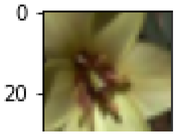
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

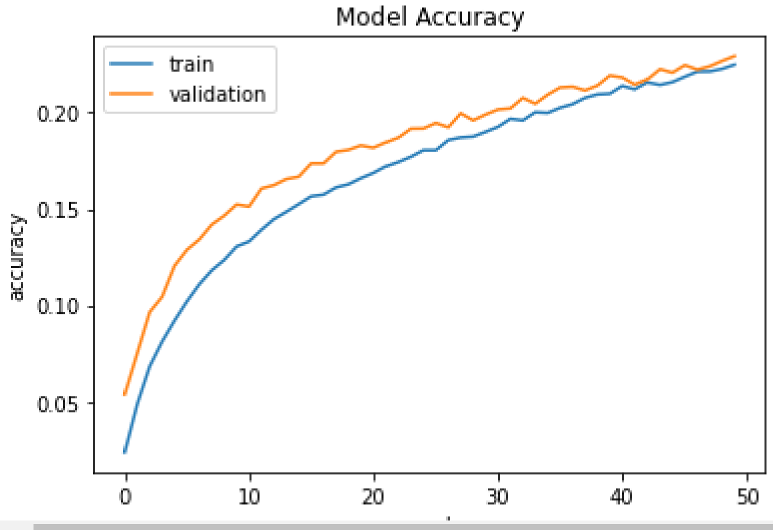


```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
score=model.evaluate(X_test,y_test,verbose=1)
print('Test loss=',score[0])
print('Test accuracy=',score[1])
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','validation'],loc='upper left')
```

```
313/313 [==============================] - 1s 4ms/step - loss: 3.2609 - accuracy: 0.2
Test loss= 3.2608563899993896
Test accuracy= 0.22920000553131104
<matplotlib.legend.Legend at 0x7f8b28622c90>
```


Model Accuracy
train
validation
accuracy