In [1]:
```
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
```

In [2]:
```
import numpy as np
```

In [3]:
```
data = np.load('cifar-10-python.npz')
```

In [4]:
```
filenames = ["x_train","y_train","x_test","y_test"]
nps = []
for filename in filenames:
    nps.append(data[filename])
train_images,train_labels,test_images,test_labels = nps
```

Shuffling, resizing images

In [5]:
```
from tensorflow.keras.utils import to_categorical
train_images, test_images = train_images / 255.0, test_images / 255.0
y_train_cat = to_categorical(train_labels,10)
y_test_cat  = to_categorical(test_labels,10)
```

In [6]:
```
train_images
```

Out[6]:
```
array([[[[0.23137255, 0.24313725, 0.24705882],
         [0.16862745, 0.18039216, 0.17647059],
         [0.19607843, 0.18823529, 0.16862745],
         ...,
         [0.61960784, 0.51764706, 0.42352941],
         [0.59607843, 0.49019608, 0.4       ],
         [0.58039216, 0.48627451, 0.40392157]],

        [[0.0627451 , 0.07843137, 0.07843137],
         [0.        , 0.        , 0.        ],
         [0.07058824, 0.03137255, 0.        ],
         ...,
         [0.48235294, 0.34509804, 0.21568627],
         [0.46666667, 0.3254902 , 0.19607843],
         [0.47843137, 0.34117647, 0.22352941]],

        [[0.09803922, 0.09411765, 0.08235294],
         [0.0627451 , 0.02745098, 0.        ],
         [0.19215686, 0.10588235, 0.03137255],
         ...,
         [0.4627451 , 0.32941176, 0.19607843],
         [0.47058824, 0.32941176, 0.19607843],
         [0.42745098, 0.28627451, 0.16470588]],

        ...,

        [[0.81568627, 0.66666667, 0.37647059],
         [0.78823529, 0.6       , 0.13333333],
         [0.77647059, 0.63137255, 0.10196078],
         ...,
         [0.62745098, 0.52156863, 0.2745098 ],
         [0.21960784, 0.12156863, 0.02745098],
         [0.20784314, 0.13333333, 0.07843137]],

        [[0.70588235, 0.54509804, 0.37647059],
         [0.67843137, 0.48235294, 0.16470588],
         [0.72941176, 0.56470588, 0.11764706],
         ...,
         [0.72156863, 0.58039216, 0.36862745],
         [0.38039216, 0.24313725, 0.13333333],
         [0.3254902 , 0.20784314, 0.13333333]],

        [[0.69411765, 0.56470588, 0.45490196],
         [0.65882353, 0.50588235, 0.36862745],
         [0.70196078, 0.55686275, 0.34117647],
         ...,
         [0.84705882, 0.72156863, 0.54901961],
         [0.59215686, 0.4627451 , 0.32941176],
         [0.48235294, 0.36078431, 0.28235294]]],


       [[[0.60392157, 0.69411765, 0.73333333],
         [0.49411765, 0.5372549 , 0.53333333],
         [0.41176471, 0.40784314, 0.37254902],
         ...,
         [0.35686275, 0.37254902, 0.27843137],
```

```
 [0.34117647, 0.35294118, 0.27843137],
 [0.30980392, 0.31764706, 0.2745098 ]],

[[0.54901961, 0.62745098, 0.6627451 ],
 [0.56862745, 0.6       , 0.60392157],
 [0.49019608, 0.49019608, 0.4627451 ],
 ...,
 [0.37647059, 0.38823529, 0.30588235],
 [0.30196078, 0.31372549, 0.24313725],
 [0.27843137, 0.28627451, 0.23921569]],

[[0.54901961, 0.60784314, 0.64313725],
 [0.54509804, 0.57254902, 0.58431373],
 [0.45098039, 0.45098039, 0.43921569],
 ...,
 [0.30980392, 0.32156863, 0.25098039],
 [0.26666667, 0.2745098 , 0.21568627],
 [0.2627451 , 0.27058824, 0.21568627]],

...,

[[0.68627451, 0.65490196, 0.65098039],
 [0.61176471, 0.60392157, 0.62745098],
 [0.60392157, 0.62745098, 0.66666667],
 ...,
 [0.16470588, 0.13333333, 0.14117647],
 [0.23921569, 0.20784314, 0.22352941],
 [0.36470588, 0.3254902 , 0.35686275]],

[[0.64705882, 0.60392157, 0.50196078],
 [0.61176471, 0.59607843, 0.50980392],
 [0.62352941, 0.63137255, 0.55686275],
 ...,
 [0.40392157, 0.36470588, 0.37647059],
 [0.48235294, 0.44705882, 0.47058824],
 [0.51372549, 0.4745098 , 0.51372549]],

[[0.63921569, 0.58039216, 0.47058824],
 [0.61960784, 0.58039216, 0.47843137],
 [0.63921569, 0.61176471, 0.52156863],
 ...,
 [0.56078431, 0.52156863, 0.54509804],
 [0.56078431, 0.5254902 , 0.55686275],
 [0.56078431, 0.52156863, 0.56470588]]],


[[[1.        , 1.        , 1.        ],
  [0.99215686, 0.99215686, 0.99215686],
  [0.99215686, 0.99215686, 0.99215686],
  ...,
  [0.99215686, 0.99215686, 0.99215686],
  [0.99215686, 0.99215686, 0.99215686],
  [0.99215686, 0.99215686, 0.99215686]],

 [[1.        , 1.        , 1.        ],
  [1.        , 1.        , 1.        ],
```

```
  [1.        , 1.        , 1.        ],
  ...,
  [1.        , 1.        , 1.        ],
  [1.        , 1.        , 1.        ],
  [1.        , 1.        , 1.        ]],

 [[1.        , 1.        , 1.        ],
  [0.99607843, 0.99607843, 0.99607843],
  [0.99607843, 0.99607843, 0.99607843],
  ...,
  [0.99607843, 0.99607843, 0.99607843],
  [0.99607843, 0.99607843, 0.99607843],
  [0.99607843, 0.99607843, 0.99607843]],

 ...,

 [[0.44313725, 0.47058824, 0.43921569],
  [0.43529412, 0.4627451 , 0.43529412],
  [0.41176471, 0.43921569, 0.41568627],
  ...,
  [0.28235294, 0.31764706, 0.31372549],
  [0.28235294, 0.31372549, 0.30980392],
  [0.28235294, 0.31372549, 0.30980392]],

 [[0.43529412, 0.4627451 , 0.43137255],
  [0.40784314, 0.43529412, 0.40784314],
  [0.38823529, 0.41568627, 0.38431373],
  ...,
  [0.26666667, 0.29411765, 0.28627451],
  [0.2745098 , 0.29803922, 0.29411765],
  [0.30588235, 0.32941176, 0.32156863]],

 [[0.41568627, 0.44313725, 0.41176471],
  [0.38823529, 0.41568627, 0.38431373],
  [0.37254902, 0.4       , 0.36862745],
  ...,
  [0.30588235, 0.33333333, 0.3254902 ],
  [0.30980392, 0.33333333, 0.3254902 ],
  [0.31372549, 0.3372549 , 0.32941176]]],


...,


[[[0.1372549 , 0.69803922, 0.92156863],
  [0.15686275, 0.69019608, 0.9372549 ],
  [0.16470588, 0.69019608, 0.94509804],
  ...,
  [0.38823529, 0.69411765, 0.85882353],
  [0.30980392, 0.57647059, 0.77254902],
  [0.34901961, 0.58039216, 0.74117647]],

 [[0.22352941, 0.71372549, 0.91764706],
  [0.17254902, 0.72156863, 0.98039216],
  [0.19607843, 0.71764706, 0.94117647],
  ...,
```

```
  [0.61176471, 0.71372549, 0.78431373],
  [0.55294118, 0.69411765, 0.80784314],
  [0.45490196, 0.58431373, 0.68627451]],

 [[0.38431373, 0.77254902, 0.92941176],
  [0.25098039, 0.74117647, 0.98823529],
  [0.27058824, 0.75294118, 0.96078431],
  ...,
  [0.7372549 , 0.76470588, 0.80784314],
  [0.46666667, 0.52941176, 0.57647059],
  [0.23921569, 0.30980392, 0.35294118]],

 ...,

 [[0.28627451, 0.30980392, 0.30196078],
  [0.20784314, 0.24705882, 0.26666667],
  [0.21176471, 0.26666667, 0.31372549],
  ...,
  [0.06666667, 0.15686275, 0.25098039],
  [0.08235294, 0.14117647, 0.2       ],
  [0.12941176, 0.18823529, 0.19215686]],

 [[0.23921569, 0.26666667, 0.29411765],
  [0.21568627, 0.2745098 , 0.3372549 ],
  [0.22352941, 0.30980392, 0.40392157],
  ...,
  [0.09411765, 0.18823529, 0.28235294],
  [0.06666667, 0.1372549 , 0.20784314],
  [0.02745098, 0.09019608, 0.1254902 ]],

 [[0.17254902, 0.21960784, 0.28627451],
  [0.18039216, 0.25882353, 0.34509804],
  [0.19215686, 0.30196078, 0.41176471],
  ...,
  [0.10588235, 0.20392157, 0.30196078],
  [0.08235294, 0.16862745, 0.25882353],
  [0.04705882, 0.12156863, 0.19607843]]],


[[[0.74117647, 0.82745098, 0.94117647],
  [0.72941176, 0.81568627, 0.9254902 ],
  [0.7254902 , 0.81176471, 0.92156863],
  ...,
  [0.68627451, 0.76470588, 0.87843137],
  [0.6745098 , 0.76078431, 0.87058824],
  [0.6627451 , 0.76078431, 0.8627451 ]],

 [[0.76078431, 0.82352941, 0.9372549 ],
  [0.74901961, 0.81176471, 0.9254902 ],
  [0.74509804, 0.80784314, 0.92156863],
  ...,
  [0.67843137, 0.75294118, 0.8627451 ],
  [0.67058824, 0.74901961, 0.85490196],
  [0.65490196, 0.74509804, 0.84705882]],

 [[0.81568627, 0.85882353, 0.95686275],
```

```
  [0.80392157, 0.84705882, 0.94117647],
  [0.8       , 0.84313725, 0.9372549 ],
  ...,
  [0.68627451, 0.74901961, 0.85098039],
  [0.6745098 , 0.74509804, 0.84705882],
  [0.6627451 , 0.74901961, 0.84313725]],

 ...,

 [[0.81176471, 0.78039216, 0.70980392],
  [0.79607843, 0.76470588, 0.68627451],
  [0.79607843, 0.76862745, 0.67843137],
  ...,
  [0.52941176, 0.51764706, 0.49803922],
  [0.63529412, 0.61960784, 0.58823529],
  [0.65882353, 0.63921569, 0.59215686]],

 [[0.77647059, 0.74509804, 0.66666667],
  [0.74117647, 0.70980392, 0.62352941],
  [0.70588235, 0.6745098 , 0.57647059],
  ...,
  [0.69803922, 0.67058824, 0.62745098],
  [0.68627451, 0.6627451 , 0.61176471],
  [0.68627451, 0.6627451 , 0.60392157]],

 [[0.77647059, 0.74117647, 0.67843137],
  [0.74117647, 0.70980392, 0.63529412],
  [0.69803922, 0.66666667, 0.58431373],
  ...,
  [0.76470588, 0.72156863, 0.6627451 ],
  [0.76862745, 0.74117647, 0.67058824],
  [0.76470588, 0.74509804, 0.67058824]]],


[[[0.89803922, 0.89803922, 0.9372549 ],
  [0.9254902 , 0.92941176, 0.96862745],
  [0.91764706, 0.9254902 , 0.96862745],
  ...,
  [0.85098039, 0.85882353, 0.91372549],
  [0.86666667, 0.8745098 , 0.91764706],
  [0.87058824, 0.8745098 , 0.91372549]],

 [[0.87058824, 0.86666667, 0.89803922],
  [0.9372549 , 0.9372549 , 0.97647059],
  [0.91372549, 0.91764706, 0.96470588],
  ...,
  [0.8745098 , 0.8745098 , 0.9254902 ],
  [0.89019608, 0.89411765, 0.93333333],
  [0.82352941, 0.82745098, 0.8627451 ]],

 [[0.83529412, 0.80784314, 0.82745098],
  [0.91764706, 0.90980392, 0.9372549 ],
  [0.90588235, 0.91372549, 0.95686275],
  ...,
  [0.8627451 , 0.8627451 , 0.90980392],
  [0.8627451 , 0.85882353, 0.90980392],
```

```
        [0.79215686, 0.79607843, 0.84313725]],

       ...,

       [[0.58823529, 0.56078431, 0.52941176],
        [0.54901961, 0.52941176, 0.49803922],
        [0.51764706, 0.49803922, 0.47058824],
        ...,
        [0.87843137, 0.87058824, 0.85490196],
        [0.90196078, 0.89411765, 0.88235294],
        [0.94509804, 0.94509804, 0.93333333]],

       [[0.5372549 , 0.51764706, 0.49411765],
        [0.50980392, 0.49803922, 0.47058824],
        [0.49019608, 0.4745098 , 0.45098039],
        ...,
        [0.70980392, 0.70588235, 0.69803922],
        [0.79215686, 0.78823529, 0.77647059],
        [0.83137255, 0.82745098, 0.81176471]],

       [[0.47843137, 0.46666667, 0.44705882],
        [0.4627451 , 0.45490196, 0.43137255],
        [0.47058824, 0.45490196, 0.43529412],
        ...,
        [0.70196078, 0.69411765, 0.67843137],
        [0.64313725, 0.64313725, 0.63529412],
        [0.63921569, 0.63921569, 0.63137255]]]], shape=(50000, 32, 32, 3))
```

In [7]:
```
import matplotlib.pyplot as plt
```

In [8]:
```
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer','dog', 'frog', 'hors
plt.figure(figsize=(10,10))
for i in range(10):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
 #     plt.grid(False)
    plt.imshow(train_images[i])
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```

In [9]:
```
from tensorflow.keras.layers import BatchNormalization, Dropout
```

In [10]:
```
model = models.Sequential()

# Conv block 1
model.add(layers.Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=
model.add(BatchNormalization())
model.add(layers.Conv2D(32, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(layers.MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

# Conv block 2
model.add(layers.Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(layers.Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(layers.MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

# Conv block 3
model.add(layers.Conv2D(128, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(layers.Conv2D(128, (3, 3), padding='same', activation='relu'))
model.add(BatchNormalization())
model.add(layers.MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

# Fully connected layers
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(layers.Dense(10, activation='softmax'))
```

```
C:\Users\SMIT DESHMUKH\AppData\Local\Programs\Python\Python313\Lib\site-packages\ker
as\src\layers\convolutional\base_conv.py:113: UserWarning: Do not pass an `input_sha
pe`/`input_dim` argument to a layer. When using Sequential models, prefer using an `
Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [11]:
```
model.compile(optimizer='adam',
 loss='categorical_crossentropy',
 metrics=['accuracy'])
```

In [12]:
```
epochs=15
history=model.fit(train_images, y_train_cat,
                  validation_data=(test_images,y_test_cat),epochs=epochs)
```

```
Epoch 1/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 258s 159ms/step - accuracy: 0.4881 - loss: 1.5081 - v
al_accuracy: 0.6154 - val_loss: 1.1117
Epoch 2/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 236s 151ms/step - accuracy: 0.6552 - loss: 0.9796 - v
al_accuracy: 0.6983 - val_loss: 0.8738
Epoch 3/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 242s 155ms/step - accuracy: 0.7089 - loss: 0.8386 - v
al_accuracy: 0.7030 - val_loss: 0.8835
Epoch 4/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 235s 150ms/step - accuracy: 0.7349 - loss: 0.7617 - v
al_accuracy: 0.7005 - val_loss: 0.8646
Epoch 5/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 249s 159ms/step - accuracy: 0.7664 - loss: 0.6788 - v
al_accuracy: 0.7663 - val_loss: 0.6815
Epoch 6/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 253s 154ms/step - accuracy: 0.7868 - loss: 0.6197 - v
al_accuracy: 0.7983 - val_loss: 0.5865
Epoch 7/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 242s 155ms/step - accuracy: 0.8004 - loss: 0.5841 - v
al_accuracy: 0.7814 - val_loss: 0.6456
Epoch 8/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 273s 162ms/step - accuracy: 0.8136 - loss: 0.5377 - v
al_accuracy: 0.8005 - val_loss: 0.5910
Epoch 9/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 258s 159ms/step - accuracy: 0.8269 - loss: 0.5024 - v
al_accuracy: 0.8236 - val_loss: 0.5349
Epoch 10/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 250s 160ms/step - accuracy: 0.8376 - loss: 0.4709 - v
al_accuracy: 0.8243 - val_loss: 0.5203
Epoch 11/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 258s 157ms/step - accuracy: 0.8473 - loss: 0.4389 - v
al_accuracy: 0.8347 - val_loss: 0.4918
Epoch 12/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 267s 160ms/step - accuracy: 0.8546 - loss: 0.4170 - v
al_accuracy: 0.8323 - val_loss: 0.5073
Epoch 13/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 250s 160ms/step - accuracy: 0.8638 - loss: 0.3898 - v
al_accuracy: 0.8378 - val_loss: 0.4838
Epoch 14/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 259s 158ms/step - accuracy: 0.8693 - loss: 0.3757 - v
al_accuracy: 0.8332 - val_loss: 0.5053
Epoch 15/15
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 252s 161ms/step - accuracy: 0.8759 - loss: 0.3519 - v
al_accuracy: 0.8536 - val_loss: 0.4580
```

In [13]:
```
test_loss,test_acc = model.evaluate(test_images,y_test_cat)
print("loss %.3f"%test_loss)
print("acc %.3f"%test_acc)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 14s 46ms/step - accuracy: 0.8536 - loss: 0.4580
loss 0.458
acc 0.854
```

In [14]:
```
predicted_values = model.predict(test_images)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 14s 43ms/step
```

In [15]:
```
predicted_values.shape
```

Out[15]:
```
(10000, 10)
```

In [16]:
```
import random
n = random.randint(0,9999)
plt.figure(figsize=(10,10))
plt.imshow(test_images[n])
plt.yticks([])
plt.xticks([])
plt.grid(False)
plt.title(class_names[np.argmax(predicted_values[n])])
```

Out[16]:
```
Text(0.5, 1.0, 'dog')
```

In [17]:
```
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')
plt.show()
```