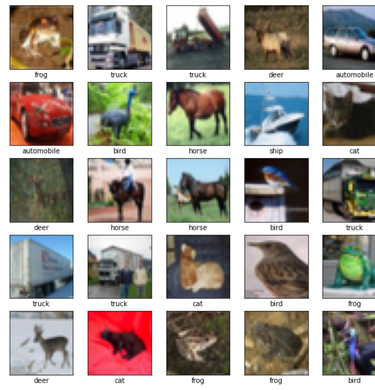
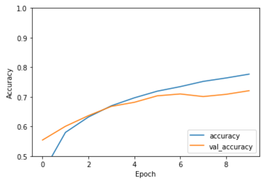
In [1]:
```
#Implementation Of CIFAR10 Using TensorFlow & keras Sequential API In Python
#Import TensorFlow
import tensorflow as tf

from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt

#Download and prepare the CIFAR10 dataset
#The CIFAR10 dataset contains 60,000 color images in 10 classes, with 6,000 images in each c
lass.
#The dataset is divided into 50,000 training images and 10,000 testing images.
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0


#Verify the data
#To verify that the dataset looks let's plot the first 25 images
#from the training set and display the class name below each image.
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']

plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    # The CIFAR labels happen to be arrays,
    # which is why you need the extra index
    plt.xlabel(class_names[train_labels[i][0]])
plt.show()

#Create the convolutional base
#As input, a CNN takes tensors of shape (image_height, image_width, color_channels)
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))

#Let's display the architecture of our model so far.
model.summary()

#Adding dense layers on top
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))

#Again checking architecture of model
model.summary()

#Compile and train the model
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=10,
                    validation_data=(test_images, test_labels))

#Evaluate the model
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)

#Testing the accuracy
print(test_acc)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 860s 5us/step
```




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928
=================================================================
Total params: 56,320
Trainable params: 56,320
Non-trainable params: 0
_________________________________________________________________
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 64)          36928
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 64)                65600
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
1563/1563 [==============================] - 35s 22ms/step - loss: 1.7542 - accuracy: 0.3496
- val_loss: 1.2608 - val_accuracy: 0.5541
Epoch 2/10
1563/1563 [==============================] - 31s 20ms/step - loss: 1.2109 - accuracy: 0.5700
- val_loss: 1.1249 - val_accuracy: 0.6006
Epoch 3/10
1563/1563 [==============================] - 34s 22ms/step - loss: 1.0514 - accuracy: 0.6288
- val_loss: 1.0512 - val_accuracy: 0.6365
Epoch 4/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.9512 - accuracy: 0.6655
- val_loss: 0.9620 - val_accuracy: 0.6678
Epoch 5/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.8651 - accuracy: 0.6960
- val_loss: 0.9086 - val_accuracy: 0.6816
Epoch 6/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.7900 - accuracy: 0.7256
- val_loss: 0.8666 - val_accuracy: 0.7035
Epoch 7/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.7455 - accuracy: 0.7401
- val_loss: 0.8556 - val_accuracy: 0.7098
Epoch 8/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.7017 - accuracy: 0.7561
- val_loss: 0.8814 - val_accuracy: 0.7010
Epoch 9/10
1563/1563 [==============================] - 36s 23ms/step - loss: 0.6559 - accuracy: 0.7710
- val_loss: 0.8667 - val_accuracy: 0.7086
Epoch 10/10
1563/1563 [==============================] - 35s 22ms/step - loss: 0.6260 - accuracy: 0.7811
- val_loss: 0.8394 - val_accuracy: 0.7208
313/313 - 2s - loss: 0.8394 - accuracy: 0.7208
0.72079998254776
```



In [ ]: