

Assignment 12

In [1]: *#import libraries*

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

In [2]: *#download and prepare the CIFAR10 dataset*

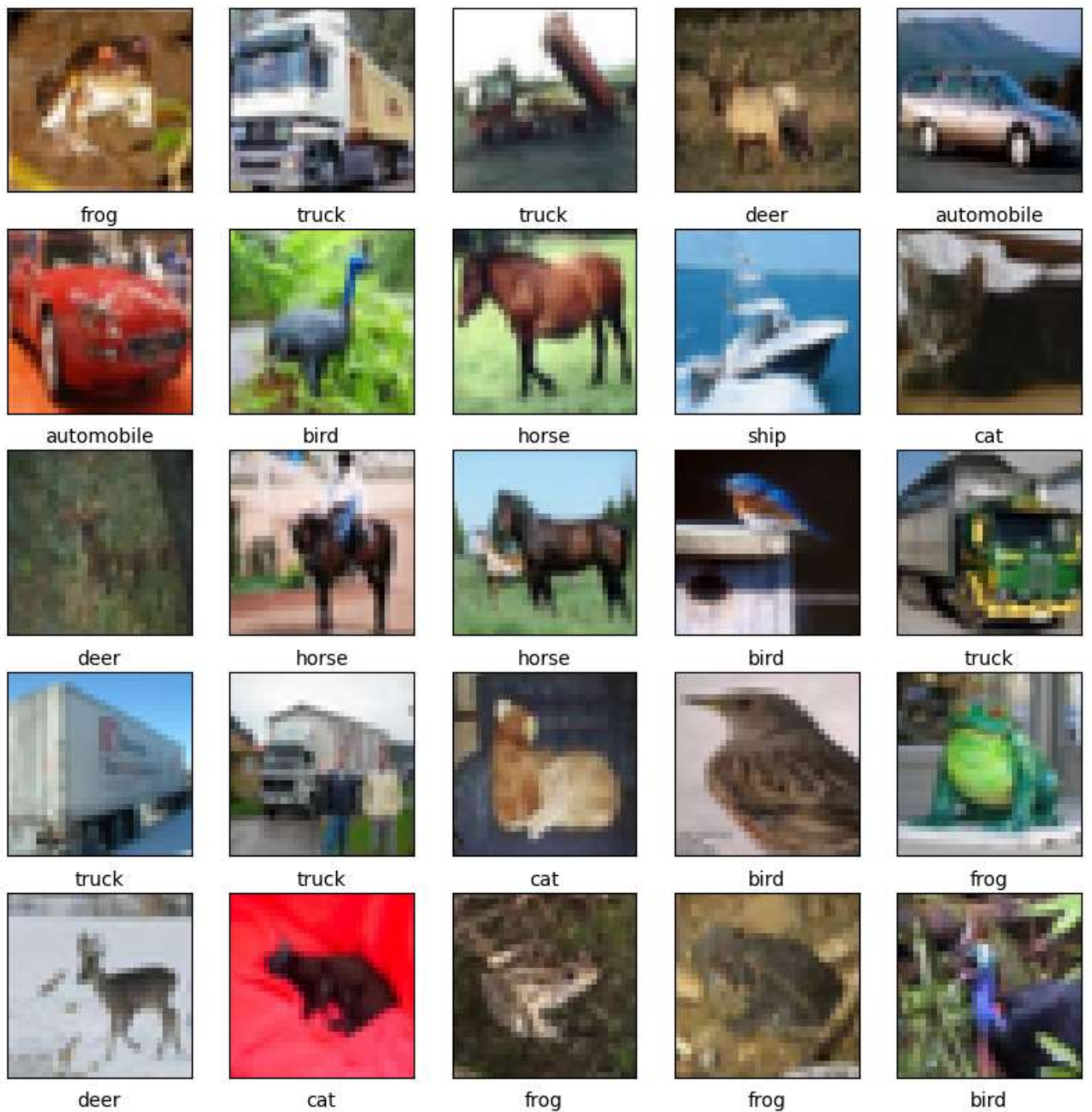
```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

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

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz> (h
<https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>)
170498071/170498071 [=====] - 208s 1us/step

In [3]: *#verify the data*

```
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])  
    # 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()
```



```
In [5]: #create CNN model

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'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))
```

```
In [6]: model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====		
conv2d_3 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_2 (MaxPooling 2D)	(None, 15, 15, 32)	0
conv2d_4 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 6, 6, 64)	0
conv2d_5 (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		

In [7]: *#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))
```

Epoch 1/10

1563/1563 [=====] - 60s 38ms/step - loss: 1.5448 - accuracy: 0.4337 - val_loss: 1.2659 - val_accuracy: 0.5439

Epoch 2/10

1563/1563 [=====] - 47s 30ms/step - loss: 1.1759 - accuracy: 0.5836 - val_loss: 1.1252 - val_accuracy: 0.6042

Epoch 3/10

1563/1563 [=====] - 57s 37ms/step - loss: 1.0229 - accuracy: 0.6408 - val_loss: 0.9674 - val_accuracy: 0.6622

Epoch 4/10

1563/1563 [=====] - 55s 35ms/step - loss: 0.9282 - accuracy: 0.6740 - val_loss: 0.9622 - val_accuracy: 0.6669

Epoch 5/10

1563/1563 [=====] - 49s 31ms/step - loss: 0.8482 - accuracy: 0.7043 - val_loss: 0.9111 - val_accuracy: 0.6895

Epoch 6/10

1563/1563 [=====] - 52s 33ms/step - loss: 0.7854 - accuracy: 0.7261 - val_loss: 0.8815 - val_accuracy: 0.6948

Epoch 7/10

1563/1563 [=====] - 60s 39ms/step - loss: 0.7364 - accuracy: 0.7413 - val_loss: 0.8587 - val_accuracy: 0.7084

Epoch 8/10

1563/1563 [=====] - 59s 38ms/step - loss: 0.6959 - accuracy: 0.7549 - val_loss: 0.8696 - val_accuracy: 0.7006

Epoch 9/10

1563/1563 [=====] - 60s 38ms/step - loss: 0.6586 - accuracy: 0.7663 - val_loss: 0.8427 - val_accuracy: 0.7112

Epoch 10/10

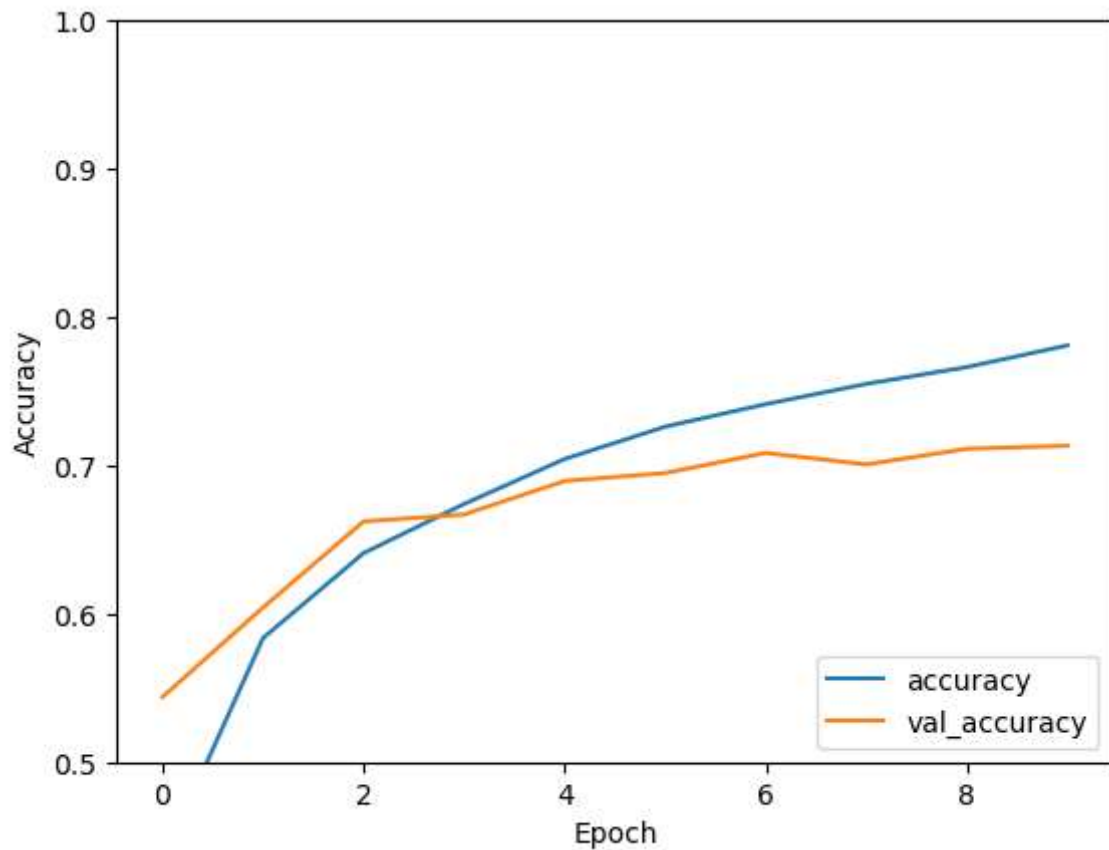
1563/1563 [=====] - 57s 36ms/step - loss: 0.6211 - accuracy: 0.7809 - val_loss: 0.8494 - val_accuracy: 0.7132

In [8]: *#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)
```

313/313 - 5s - loss: 0.8494 - accuracy: 0.7132 - 5s/epoch - 15ms/step



In [9]: `print(test_acc)`

0.7131999731063843