6.2.b ConvNet: CIFGAR10 image classifier

In [1]:
```
import os
from google.colab import drive
drive.mount('/content/drive', force_remount = True)
os.chdir('/content/drive/My Drive/DSC650/assignment06')
!pwd
```

```
Mounted at /content/drive
/content/drive/My Drive/DSC650/assignment06
```

In [28]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pickle

from keras import layers, models
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import (
    Dense, Dropout, Activation,
    Conv2D, MaxPooling2D, Flatten,
    BatchNormalization
)
from keras.utils import np_utils, to_categorical
from keras.optimizers import SGD
from keras.preprocessing.image import ImageDataGenerator
```

In [3]:
```
(trainX, trainy), (testX, testy) = cifar10.load_data()
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 2s 0us/step
```

In [4]:
```
# get the size of the data sets
print(f'train_images: {trainX.shape}')
print(f'test_images: {testX.shape}')
print(f'train_labels: {trainy.shape}')
print(f'test_labels: {testy.shape}')
```

```
train_images: (50000, 32, 32, 3)
test_images: (10000, 32, 32, 3)
train_labels: (50000, 1)
test_labels: (10000, 1)
```

In [5]:
```
# Assignment classes for visualization
cifar10_classes = ['airplane', 'automobile', 'bird', 'cat',
                   'deer', 'frog', 'horse', 'ship', 'truck']
```

Visualize sample images

In [6]:
```
fig, ax = plt.subplots(5, 5)
k = 0

for i in range(5):
  for j in range(5):
    ax[i][j].imshow(trainX[k], aspect = 'auto')
```

```
    k += 1

plt.show()
```

In [7]:

```
# normalize datasets
train_images = trainX.astype('float32') / 255.0
test_images = testX.astype('float32') / 255.0
```

In [8]:

```
# convert labels to numeric
train_labels = to_categorical(trainy)
test_labels = to_categorical(testy)
```

Split training data into training and validation datasets

In [9]:

```
x_val = train_images[:10000]
partial_x_train = train_images[10000:]

y_val = train_labels[:10000]
partial_y_train = train_labels[10000:]
```

In [10]:

```
# get the size of the data sets
print(f'x_val: {x_val.shape}')
print(f'y_val: {y_val.shape}')
print(f'partial_x_train: {partial_x_train.shape}')
print(f'partial_y_train: {partial_y_train.shape}')
```

```
x_val: (10000, 32, 32, 3)
y_val: (10000, 10)
partial_x_train: (40000, 32, 32, 3)
partial_y_train: (40000, 10)
```

Build the Model

In [29]:

```
# Instantiate a convnet
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', paddi
model.add(BatchNormalization())
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', paddi
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', paddi
```

```python
model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', paddi
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padd
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padd
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.4))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
```

In [30]:

```python
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 32, 32, 32)        896

 batch_normalization (BatchN  (None, 32, 32, 32)       128
 ormalization)

 conv2d_13 (Conv2D)          (None, 32, 32, 32)        9248

 batch_normalization_1 (Batc  (None, 32, 32, 32)       128
 hNormalization)

 max_pooling2d_6 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 dropout_8 (Dropout)         (None, 16, 16, 32)        0

 conv2d_14 (Conv2D)          (None, 16, 16, 64)        18496

 batch_normalization_2 (Batc  (None, 16, 16, 64)       256
 hNormalization)

 conv2d_15 (Conv2D)          (None, 16, 16, 64)        36928

 batch_normalization_3 (Batc  (None, 16, 16, 64)       256
 hNormalization)

 max_pooling2d_7 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 dropout_9 (Dropout)         (None, 8, 8, 64)          0

 conv2d_16 (Conv2D)          (None, 8, 8, 128)         73856

 batch_normalization_4 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 conv2d_17 (Conv2D)          (None, 8, 8, 128)         147584

 batch_normalization_5 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 max_pooling2d_8 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout_10 (Dropout)        (None, 4, 4, 128)         0

 flatten_2 (Flatten)         (None, 2048)              0

 dense_4 (Dense)             (None, 128)               262272

 batch_normalization_6 (Batc  (None, 128)              512
 hNormalization)

 dropout_11 (Dropout)        (None, 128)               0

 dense_5 (Dense)             (None, 10)                1290
```

```
=================================================================
Total params: 552,874
Trainable params: 551,722
Non-trainable params: 1,152
_________________________________________________________________
```

Compile the Model

In [31]:
```
opt = SGD(lr = 0.001, momentum = 0.9)

model.compile(loss = 'categorical_crossentropy',
              optimizer = opt,
              metrics = ['accuracy'])
```

```
/usr/local/lib/python3.8/dist-packages/keras/optimizers/optimizer_v2/gradient_descen
t.py:108: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(SGD, self).__init__(name, **kwargs)
```

Train the model

In [32]:
```
# create data generator
datagen = ImageDataGenerator(width_shift_range=0.1, height_shift_range=0.1, horizontal

# prepare iterator
it_train = datagen.flow(partial_x_train, partial_y_train, batch_size=64)

steps = int(partial_x_train.shape[0] / 64)
```

In [34]:
```
history = model.fit(it_train, steps_per_epoch = steps,
                    epochs = 200, validation_data = (x_val, y_val), verbose = 1)
```

```
Epoch 1/200
625/625 [==============================] - 23s 36ms/step - loss: 0.9010 - accuracy:
0.6829 - val_loss: 0.8049 - val_accuracy: 0.7176
Epoch 2/200
625/625 [==============================] - 25s 39ms/step - loss: 0.8899 - accuracy:
0.6870 - val_loss: 0.8892 - val_accuracy: 0.6920
Epoch 3/200
625/625 [==============================] - 23s 37ms/step - loss: 0.8796 - accuracy:
0.6900 - val_loss: 0.9128 - val_accuracy: 0.6839
Epoch 4/200
625/625 [==============================] - 24s 38ms/step - loss: 0.8768 - accuracy:
0.6937 - val_loss: 0.8745 - val_accuracy: 0.6956
Epoch 5/200
625/625 [==============================] - 24s 39ms/step - loss: 0.8647 - accuracy:
0.6944 - val_loss: 0.8409 - val_accuracy: 0.7104
Epoch 6/200
625/625 [==============================] - 25s 41ms/step - loss: 0.8595 - accuracy:
0.6999 - val_loss: 0.8192 - val_accuracy: 0.7164
Epoch 7/200
625/625 [==============================] - 28s 45ms/step - loss: 0.8521 - accuracy:
0.7020 - val_loss: 0.7516 - val_accuracy: 0.7400
Epoch 8/200
625/625 [==============================] - 23s 36ms/step - loss: 0.8418 - accuracy:
0.7068 - val_loss: 0.8637 - val_accuracy: 0.7044
Epoch 9/200
625/625 [==============================] - 23s 37ms/step - loss: 0.8388 - accuracy:
0.7026 - val_loss: 0.7549 - val_accuracy: 0.7382
Epoch 10/200
625/625 [==============================] - 23s 36ms/step - loss: 0.8346 - accuracy:
0.7082 - val_loss: 0.7974 - val_accuracy: 0.7259
Epoch 11/200
625/625 [==============================] - 23s 36ms/step - loss: 0.8236 - accuracy:
0.7110 - val_loss: 0.8216 - val_accuracy: 0.7216
Epoch 12/200
625/625 [==============================] - 23s 36ms/step - loss: 0.8131 - accuracy:
0.7140 - val_loss: 0.7091 - val_accuracy: 0.7507
Epoch 13/200
625/625 [==============================] - 23s 36ms/step - loss: 0.8090 - accuracy:
0.7175 - val_loss: 0.7178 - val_accuracy: 0.7470
Epoch 14/200
625/625 [==============================] - 24s 39ms/step - loss: 0.7996 - accuracy:
0.7200 - val_loss: 0.8014 - val_accuracy: 0.7233
Epoch 15/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7989 - accuracy:
0.7226 - val_loss: 0.7716 - val_accuracy: 0.7268
Epoch 16/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7917 - accuracy:
0.7225 - val_loss: 0.7219 - val_accuracy: 0.7497
Epoch 17/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7828 - accuracy:
0.7265 - val_loss: 0.7204 - val_accuracy: 0.7493
Epoch 18/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7777 - accuracy:
0.7290 - val_loss: 0.8124 - val_accuracy: 0.7218
Epoch 19/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7722 - accuracy:
0.7303 - val_loss: 0.7087 - val_accuracy: 0.7532
Epoch 20/200
625/625 [==============================] - 24s 39ms/step - loss: 0.7665 - accuracy:
0.7326 - val_loss: 0.7312 - val_accuracy: 0.7474
```

```
Epoch 21/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7596 - accuracy:
0.7356 - val_loss: 0.7128 - val_accuracy: 0.7516
Epoch 22/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7510 - accuracy:
0.7356 - val_loss: 0.7540 - val_accuracy: 0.7419
Epoch 23/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7503 - accuracy:
0.7367 - val_loss: 0.7482 - val_accuracy: 0.7408
Epoch 24/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7505 - accuracy:
0.7388 - val_loss: 0.7363 - val_accuracy: 0.7461
Epoch 25/200
625/625 [==============================] - 22s 36ms/step - loss: 0.7392 - accuracy:
0.7417 - val_loss: 0.7111 - val_accuracy: 0.7558
Epoch 26/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7398 - accuracy:
0.7421 - val_loss: 0.6891 - val_accuracy: 0.7606
Epoch 27/200
625/625 [==============================] - 24s 39ms/step - loss: 0.7318 - accuracy:
0.7454 - val_loss: 0.7574 - val_accuracy: 0.7386
Epoch 28/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7232 - accuracy:
0.7479 - val_loss: 0.6414 - val_accuracy: 0.7759
Epoch 29/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7275 - accuracy:
0.7452 - val_loss: 0.6657 - val_accuracy: 0.7685
Epoch 30/200
625/625 [==============================] - 23s 36ms/step - loss: 0.7194 - accuracy:
0.7505 - val_loss: 0.6352 - val_accuracy: 0.7796
Epoch 31/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7137 - accuracy:
0.7525 - val_loss: 0.6344 - val_accuracy: 0.7791
Epoch 32/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7043 - accuracy:
0.7552 - val_loss: 0.6781 - val_accuracy: 0.7639
Epoch 33/200
625/625 [==============================] - 23s 37ms/step - loss: 0.7021 - accuracy:
0.7574 - val_loss: 0.7103 - val_accuracy: 0.7595
Epoch 34/200
625/625 [==============================] - 24s 39ms/step - loss: 0.7013 - accuracy:
0.7594 - val_loss: 0.6145 - val_accuracy: 0.7881
Epoch 35/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6972 - accuracy:
0.7583 - val_loss: 0.6162 - val_accuracy: 0.7877
Epoch 36/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6948 - accuracy:
0.7581 - val_loss: 0.6305 - val_accuracy: 0.7828
Epoch 37/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6867 - accuracy:
0.7594 - val_loss: 0.6609 - val_accuracy: 0.7751
Epoch 38/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6816 - accuracy:
0.7627 - val_loss: 0.5848 - val_accuracy: 0.7973
Epoch 39/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6731 - accuracy:
0.7668 - val_loss: 0.6277 - val_accuracy: 0.7828
Epoch 40/200
625/625 [==============================] - 24s 39ms/step - loss: 0.6788 - accuracy:
0.7677 - val_loss: 0.7577 - val_accuracy: 0.7485
```

```
Epoch 41/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6690 - accuracy: 0.7677 - val_loss: 0.5739 - val_accuracy: 0.8012
Epoch 42/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6712 - accuracy: 0.7676 - val_loss: 0.6415 - val_accuracy: 0.7736
Epoch 43/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6609 - accuracy: 0.7721 - val_loss: 0.6147 - val_accuracy: 0.7894
Epoch 44/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6554 - accuracy: 0.7737 - val_loss: 0.5942 - val_accuracy: 0.7930
Epoch 45/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6583 - accuracy: 0.7705 - val_loss: 0.5538 - val_accuracy: 0.8069
Epoch 46/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6439 - accuracy: 0.7770 - val_loss: 0.5977 - val_accuracy: 0.7953
Epoch 47/200
625/625 [==============================] - 24s 39ms/step - loss: 0.6498 - accuracy: 0.7735 - val_loss: 0.5624 - val_accuracy: 0.8075
Epoch 48/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6505 - accuracy: 0.7725 - val_loss: 0.6075 - val_accuracy: 0.7894
Epoch 49/200
625/625 [==============================] - 23s 37ms/step - loss: 0.6442 - accuracy: 0.7779 - val_loss: 0.5940 - val_accuracy: 0.7946
Epoch 50/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6395 - accuracy: 0.7781 - val_loss: 0.5716 - val_accuracy: 0.8015
Epoch 51/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6437 - accuracy: 0.7773 - val_loss: 0.5910 - val_accuracy: 0.7994
Epoch 52/200
625/625 [==============================] - 23s 37ms/step - loss: 0.6286 - accuracy: 0.7835 - val_loss: 0.5769 - val_accuracy: 0.7987
Epoch 53/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6320 - accuracy: 0.7829 - val_loss: 0.5811 - val_accuracy: 0.7978
Epoch 54/200
625/625 [==============================] - 24s 39ms/step - loss: 0.6334 - accuracy: 0.7815 - val_loss: 0.5570 - val_accuracy: 0.8097
Epoch 55/200
625/625 [==============================] - 23s 37ms/step - loss: 0.6336 - accuracy: 0.7806 - val_loss: 0.5696 - val_accuracy: 0.8039
Epoch 56/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6255 - accuracy: 0.7851 - val_loss: 0.5400 - val_accuracy: 0.8125
Epoch 57/200
625/625 [==============================] - 23s 37ms/step - loss: 0.6262 - accuracy: 0.7835 - val_loss: 0.5845 - val_accuracy: 0.8004
Epoch 58/200
625/625 [==============================] - 23s 37ms/step - loss: 0.6172 - accuracy: 0.7864 - val_loss: 0.5949 - val_accuracy: 0.7960
Epoch 59/200
625/625 [==============================] - 24s 39ms/step - loss: 0.6116 - accuracy: 0.7888 - val_loss: 0.5647 - val_accuracy: 0.8071
Epoch 60/200
625/625 [==============================] - 31s 50ms/step - loss: 0.6152 - accuracy: 0.7895 - val_loss: 0.5658 - val_accuracy: 0.8069
```

```
Epoch 61/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6058 - accuracy:
0.7883 - val_loss: 0.6429 - val_accuracy: 0.7823
Epoch 62/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6069 - accuracy:
0.7907 - val_loss: 0.5695 - val_accuracy: 0.8057
Epoch 63/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6022 - accuracy:
0.7920 - val_loss: 0.6617 - val_accuracy: 0.7773
Epoch 64/200
625/625 [==============================] - 24s 38ms/step - loss: 0.6015 - accuracy:
0.7929 - val_loss: 0.5013 - val_accuracy: 0.8270
Epoch 65/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5980 - accuracy:
0.7931 - val_loss: 0.5704 - val_accuracy: 0.8053
Epoch 66/200
625/625 [==============================] - 23s 36ms/step - loss: 0.6004 - accuracy:
0.7925 - val_loss: 0.5569 - val_accuracy: 0.8130
Epoch 67/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5955 - accuracy:
0.7944 - val_loss: 0.5095 - val_accuracy: 0.8241
Epoch 68/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5879 - accuracy:
0.7974 - val_loss: 0.5500 - val_accuracy: 0.8109
Epoch 69/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5823 - accuracy:
0.7985 - val_loss: 0.5404 - val_accuracy: 0.8133
Epoch 70/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5911 - accuracy:
0.7980 - val_loss: 0.5288 - val_accuracy: 0.8207
Epoch 71/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5734 - accuracy:
0.8026 - val_loss: 0.5065 - val_accuracy: 0.8243
Epoch 72/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5831 - accuracy:
0.7975 - val_loss: 0.5680 - val_accuracy: 0.8073
Epoch 73/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5799 - accuracy:
0.7993 - val_loss: 0.5203 - val_accuracy: 0.8224
Epoch 74/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5737 - accuracy:
0.8037 - val_loss: 0.5160 - val_accuracy: 0.8230
Epoch 75/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5747 - accuracy:
0.8034 - val_loss: 0.5450 - val_accuracy: 0.8154
Epoch 76/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5649 - accuracy:
0.8060 - val_loss: 0.5977 - val_accuracy: 0.8014
Epoch 77/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5696 - accuracy:
0.8061 - val_loss: 0.4726 - val_accuracy: 0.8393
Epoch 78/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5670 - accuracy:
0.8044 - val_loss: 0.5153 - val_accuracy: 0.8270
Epoch 79/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5666 - accuracy:
0.8046 - val_loss: 0.5333 - val_accuracy: 0.8193
Epoch 80/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5603 - accuracy:
0.8086 - val_loss: 0.5267 - val_accuracy: 0.8193
```

```
Epoch 81/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5524 - accuracy:
0.8106 - val_loss: 0.4927 - val_accuracy: 0.8298
Epoch 82/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5605 - accuracy:
0.8066 - val_loss: 0.5335 - val_accuracy: 0.8178
Epoch 83/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5596 - accuracy:
0.8077 - val_loss: 0.5066 - val_accuracy: 0.8277
Epoch 84/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5528 - accuracy:
0.8096 - val_loss: 0.4922 - val_accuracy: 0.8286
Epoch 85/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5509 - accuracy:
0.8091 - val_loss: 0.5142 - val_accuracy: 0.8255
Epoch 86/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5462 - accuracy:
0.8131 - val_loss: 0.5218 - val_accuracy: 0.8237
Epoch 87/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5400 - accuracy:
0.8150 - val_loss: 0.4908 - val_accuracy: 0.8323
Epoch 88/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5415 - accuracy:
0.8122 - val_loss: 0.5605 - val_accuracy: 0.8123
Epoch 89/200
625/625 [==============================] - 24s 38ms/step - loss: 0.5421 - accuracy:
0.8146 - val_loss: 0.5072 - val_accuracy: 0.8247
Epoch 90/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5365 - accuracy:
0.8165 - val_loss: 0.4598 - val_accuracy: 0.8425
Epoch 91/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5419 - accuracy:
0.8138 - val_loss: 0.4602 - val_accuracy: 0.8439
Epoch 92/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5409 - accuracy:
0.8134 - val_loss: 0.4767 - val_accuracy: 0.8387
Epoch 93/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5381 - accuracy:
0.8164 - val_loss: 0.5334 - val_accuracy: 0.8193
Epoch 94/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5315 - accuracy:
0.8157 - val_loss: 0.5331 - val_accuracy: 0.8204
Epoch 95/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5306 - accuracy:
0.8177 - val_loss: 0.4889 - val_accuracy: 0.8330
Epoch 96/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5301 - accuracy:
0.8169 - val_loss: 0.4609 - val_accuracy: 0.8434
Epoch 97/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5289 - accuracy:
0.8185 - val_loss: 0.4808 - val_accuracy: 0.8372
Epoch 98/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5270 - accuracy:
0.8208 - val_loss: 0.4871 - val_accuracy: 0.8344
Epoch 99/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5273 - accuracy:
0.8196 - val_loss: 0.4780 - val_accuracy: 0.8367
Epoch 100/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5216 - accuracy:
0.8194 - val_loss: 0.4753 - val_accuracy: 0.8367
```

```
Epoch 101/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5229 - accuracy:
0.8218 - val_loss: 0.4677 - val_accuracy: 0.8414
Epoch 102/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5111 - accuracy:
0.8252 - val_loss: 0.4437 - val_accuracy: 0.8470
Epoch 103/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5202 - accuracy:
0.8220 - val_loss: 0.4807 - val_accuracy: 0.8402
Epoch 104/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5166 - accuracy:
0.8204 - val_loss: 0.4849 - val_accuracy: 0.8373
Epoch 105/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5089 - accuracy:
0.8260 - val_loss: 0.4492 - val_accuracy: 0.8464
Epoch 106/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5084 - accuracy:
0.8270 - val_loss: 0.4671 - val_accuracy: 0.8420
Epoch 107/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5072 - accuracy:
0.8259 - val_loss: 0.4614 - val_accuracy: 0.8451
Epoch 108/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5056 - accuracy:
0.8271 - val_loss: 0.4512 - val_accuracy: 0.8508
Epoch 109/200
625/625 [==============================] - 24s 39ms/step - loss: 0.5058 - accuracy:
0.8245 - val_loss: 0.4629 - val_accuracy: 0.8417
Epoch 110/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5077 - accuracy:
0.8235 - val_loss: 0.5193 - val_accuracy: 0.8274
Epoch 111/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5001 - accuracy:
0.8285 - val_loss: 0.5383 - val_accuracy: 0.8203
Epoch 112/200
625/625 [==============================] - 23s 36ms/step - loss: 0.5018 - accuracy:
0.8262 - val_loss: 0.4530 - val_accuracy: 0.8490
Epoch 113/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4949 - accuracy:
0.8313 - val_loss: 0.5055 - val_accuracy: 0.8317
Epoch 114/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4977 - accuracy:
0.8291 - val_loss: 0.4649 - val_accuracy: 0.8421
Epoch 115/200
625/625 [==============================] - 23s 37ms/step - loss: 0.5011 - accuracy:
0.8280 - val_loss: 0.4320 - val_accuracy: 0.8539
Epoch 116/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4934 - accuracy:
0.8331 - val_loss: 0.4430 - val_accuracy: 0.8482
Epoch 117/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4984 - accuracy:
0.8291 - val_loss: 0.4335 - val_accuracy: 0.8544
Epoch 118/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4953 - accuracy:
0.8294 - val_loss: 0.4645 - val_accuracy: 0.8426
Epoch 119/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4921 - accuracy:
0.8317 - val_loss: 0.4854 - val_accuracy: 0.8382
Epoch 120/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4917 - accuracy:
0.8309 - val_loss: 0.4869 - val_accuracy: 0.8358
```

```
Epoch 121/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4858 - accuracy:
0.8324 - val_loss: 0.5399 - val_accuracy: 0.8220
Epoch 122/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4883 - accuracy:
0.8320 - val_loss: 0.4853 - val_accuracy: 0.8359
Epoch 123/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4830 - accuracy:
0.8348 - val_loss: 0.4683 - val_accuracy: 0.8421
Epoch 124/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4931 - accuracy:
0.8315 - val_loss: 0.4550 - val_accuracy: 0.8461
Epoch 125/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4848 - accuracy:
0.8355 - val_loss: 0.4388 - val_accuracy: 0.8519
Epoch 126/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4784 - accuracy:
0.8389 - val_loss: 0.4568 - val_accuracy: 0.8470
Epoch 127/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4780 - accuracy:
0.8380 - val_loss: 0.4377 - val_accuracy: 0.8514
Epoch 128/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4802 - accuracy:
0.8360 - val_loss: 0.4632 - val_accuracy: 0.8454
Epoch 129/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4735 - accuracy:
0.8378 - val_loss: 0.4314 - val_accuracy: 0.8526
Epoch 130/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4761 - accuracy:
0.8345 - val_loss: 0.4488 - val_accuracy: 0.8496
Epoch 131/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4763 - accuracy:
0.8376 - val_loss: 0.4630 - val_accuracy: 0.8458
Epoch 132/200
625/625 [==============================] - 25s 39ms/step - loss: 0.4750 - accuracy:
0.8376 - val_loss: 0.4401 - val_accuracy: 0.8540
Epoch 133/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4707 - accuracy:
0.8389 - val_loss: 0.4454 - val_accuracy: 0.8505
Epoch 134/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4702 - accuracy:
0.8409 - val_loss: 0.4341 - val_accuracy: 0.8554
Epoch 135/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4698 - accuracy:
0.8405 - val_loss: 0.4515 - val_accuracy: 0.8501
Epoch 136/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4676 - accuracy:
0.8391 - val_loss: 0.4851 - val_accuracy: 0.8417
Epoch 137/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4734 - accuracy:
0.8364 - val_loss: 0.4384 - val_accuracy: 0.8519
Epoch 138/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4694 - accuracy:
0.8404 - val_loss: 0.4450 - val_accuracy: 0.8490
Epoch 139/200
625/625 [==============================] - 25s 39ms/step - loss: 0.4667 - accuracy:
0.8390 - val_loss: 0.4519 - val_accuracy: 0.8493
Epoch 140/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4596 - accuracy:
0.8407 - val_loss: 0.4318 - val_accuracy: 0.8559
```

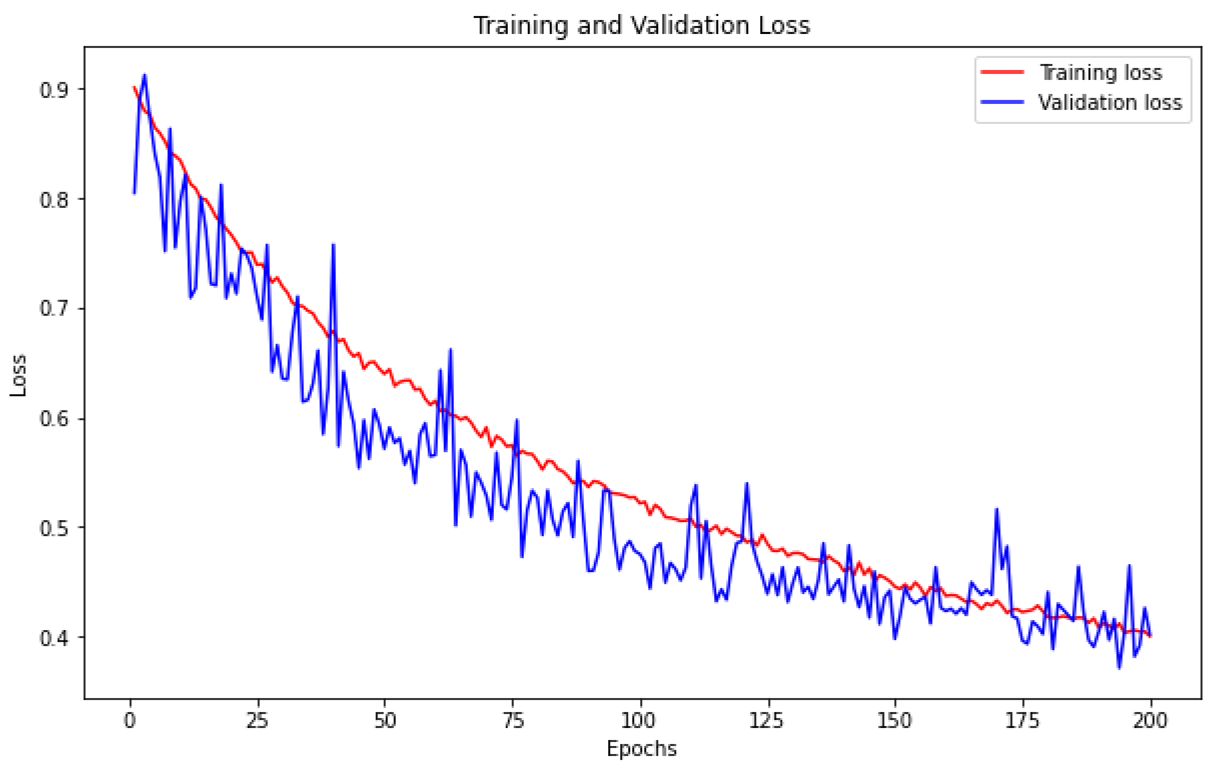
```
Epoch 141/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4628 - accuracy:
0.8400 - val_loss: 0.4833 - val_accuracy: 0.8375
Epoch 142/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4559 - accuracy:
0.8440 - val_loss: 0.4434 - val_accuracy: 0.8532
Epoch 143/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4676 - accuracy:
0.8406 - val_loss: 0.4266 - val_accuracy: 0.8549
Epoch 144/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4564 - accuracy:
0.8429 - val_loss: 0.4459 - val_accuracy: 0.8519
Epoch 145/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4623 - accuracy:
0.8433 - val_loss: 0.4171 - val_accuracy: 0.8583
Epoch 146/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4493 - accuracy:
0.8461 - val_loss: 0.4593 - val_accuracy: 0.8460
Epoch 147/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4559 - accuracy:
0.8430 - val_loss: 0.4113 - val_accuracy: 0.8628
Epoch 148/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4536 - accuracy:
0.8440 - val_loss: 0.4357 - val_accuracy: 0.8572
Epoch 149/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4504 - accuracy:
0.8479 - val_loss: 0.4419 - val_accuracy: 0.8527
Epoch 150/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4454 - accuracy:
0.8480 - val_loss: 0.3977 - val_accuracy: 0.8637
Epoch 151/200
625/625 [==============================] - 25s 40ms/step - loss: 0.4433 - accuracy:
0.8479 - val_loss: 0.4185 - val_accuracy: 0.8626
Epoch 152/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4475 - accuracy:
0.8462 - val_loss: 0.4445 - val_accuracy: 0.8519
Epoch 153/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4427 - accuracy:
0.8482 - val_loss: 0.4344 - val_accuracy: 0.8556
Epoch 154/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4493 - accuracy:
0.8473 - val_loss: 0.4301 - val_accuracy: 0.8570
Epoch 155/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4444 - accuracy:
0.8474 - val_loss: 0.4334 - val_accuracy: 0.8550
Epoch 156/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4367 - accuracy:
0.8495 - val_loss: 0.4362 - val_accuracy: 0.8557
Epoch 157/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4452 - accuracy:
0.8477 - val_loss: 0.4117 - val_accuracy: 0.8609
Epoch 158/200
625/625 [==============================] - 25s 40ms/step - loss: 0.4414 - accuracy:
0.8475 - val_loss: 0.4632 - val_accuracy: 0.8475
Epoch 159/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4441 - accuracy:
0.8477 - val_loss: 0.4262 - val_accuracy: 0.8583
Epoch 160/200
625/625 [==============================] - 25s 39ms/step - loss: 0.4369 - accuracy:
0.8507 - val_loss: 0.4231 - val_accuracy: 0.8599
```

```
Epoch 161/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4378 - accuracy:
0.8486 - val_loss: 0.4253 - val_accuracy: 0.8589
Epoch 162/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4373 - accuracy:
0.8490 - val_loss: 0.4206 - val_accuracy: 0.8631
Epoch 163/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4345 - accuracy:
0.8505 - val_loss: 0.4257 - val_accuracy: 0.8581
Epoch 164/200
625/625 [==============================] - 26s 42ms/step - loss: 0.4314 - accuracy:
0.8523 - val_loss: 0.4198 - val_accuracy: 0.8609
Epoch 165/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4327 - accuracy:
0.8518 - val_loss: 0.4494 - val_accuracy: 0.8539
Epoch 166/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4295 - accuracy:
0.8552 - val_loss: 0.4431 - val_accuracy: 0.8572
Epoch 167/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4250 - accuracy:
0.8537 - val_loss: 0.4380 - val_accuracy: 0.8598
Epoch 168/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4302 - accuracy:
0.8513 - val_loss: 0.4422 - val_accuracy: 0.8552
Epoch 169/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4284 - accuracy:
0.8536 - val_loss: 0.4380 - val_accuracy: 0.8550
Epoch 170/200
625/625 [==============================] - 26s 42ms/step - loss: 0.4328 - accuracy:
0.8511 - val_loss: 0.5164 - val_accuracy: 0.8352
Epoch 171/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4278 - accuracy:
0.8517 - val_loss: 0.4616 - val_accuracy: 0.8474
Epoch 172/200
625/625 [==============================] - 24s 39ms/step - loss: 0.4213 - accuracy:
0.8569 - val_loss: 0.4823 - val_accuracy: 0.8464
Epoch 173/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4245 - accuracy:
0.8539 - val_loss: 0.4184 - val_accuracy: 0.8600
Epoch 174/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4247 - accuracy:
0.8539 - val_loss: 0.4163 - val_accuracy: 0.8619
Epoch 175/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4222 - accuracy:
0.8541 - val_loss: 0.3961 - val_accuracy: 0.8687
Epoch 176/200
625/625 [==============================] - 25s 39ms/step - loss: 0.4232 - accuracy:
0.8554 - val_loss: 0.3932 - val_accuracy: 0.8687
Epoch 177/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4243 - accuracy:
0.8531 - val_loss: 0.4136 - val_accuracy: 0.8631
Epoch 178/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4281 - accuracy:
0.8549 - val_loss: 0.4092 - val_accuracy: 0.8635
Epoch 179/200
625/625 [==============================] - 23s 36ms/step - loss: 0.4231 - accuracy:
0.8541 - val_loss: 0.4021 - val_accuracy: 0.8690
Epoch 180/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4175 - accuracy:
0.8568 - val_loss: 0.4406 - val_accuracy: 0.8524
```

```
Epoch 181/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4169 - accuracy:
0.8579 - val_loss: 0.3883 - val_accuracy: 0.8688
Epoch 182/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4172 - accuracy:
0.8560 - val_loss: 0.4296 - val_accuracy: 0.8572
Epoch 183/200
625/625 [==============================] - 25s 40ms/step - loss: 0.4185 - accuracy:
0.8573 - val_loss: 0.4243 - val_accuracy: 0.8609
Epoch 184/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4172 - accuracy:
0.8565 - val_loss: 0.4191 - val_accuracy: 0.8642
Epoch 185/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4161 - accuracy:
0.8572 - val_loss: 0.4141 - val_accuracy: 0.8613
Epoch 186/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4172 - accuracy:
0.8550 - val_loss: 0.4638 - val_accuracy: 0.8526
Epoch 187/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4168 - accuracy:
0.8549 - val_loss: 0.4225 - val_accuracy: 0.8635
Epoch 188/200
625/625 [==============================] - 25s 40ms/step - loss: 0.4124 - accuracy:
0.8571 - val_loss: 0.3965 - val_accuracy: 0.8698
Epoch 189/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4165 - accuracy:
0.8567 - val_loss: 0.3904 - val_accuracy: 0.8704
Epoch 190/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4077 - accuracy:
0.8583 - val_loss: 0.4048 - val_accuracy: 0.8674
Epoch 191/200
625/625 [==============================] - 23s 38ms/step - loss: 0.4121 - accuracy:
0.8609 - val_loss: 0.4224 - val_accuracy: 0.8610
Epoch 192/200
625/625 [==============================] - 23s 37ms/step - loss: 0.4099 - accuracy:
0.8602 - val_loss: 0.3968 - val_accuracy: 0.8677
Epoch 193/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4065 - accuracy:
0.8579 - val_loss: 0.4161 - val_accuracy: 0.8668
Epoch 194/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4118 - accuracy:
0.8596 - val_loss: 0.3711 - val_accuracy: 0.8777
Epoch 195/200
625/625 [==============================] - 23s 38ms/step - loss: 0.4032 - accuracy:
0.8634 - val_loss: 0.4039 - val_accuracy: 0.8671
Epoch 196/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4045 - accuracy:
0.8620 - val_loss: 0.4647 - val_accuracy: 0.8484
Epoch 197/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4055 - accuracy:
0.8618 - val_loss: 0.3816 - val_accuracy: 0.8742
Epoch 198/200
625/625 [==============================] - 25s 40ms/step - loss: 0.4044 - accuracy:
0.8598 - val_loss: 0.3916 - val_accuracy: 0.8706
Epoch 199/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4044 - accuracy:
0.8615 - val_loss: 0.4259 - val_accuracy: 0.8608
Epoch 200/200
625/625 [==============================] - 24s 38ms/step - loss: 0.4001 - accuracy:
0.8618 - val_loss: 0.4018 - val_accuracy: 0.8690
```

Plot Training and Validation Loss

In [35]:
```
plt.figure(figsize = (10, 6))

loss_values = history.history['loss']
val_loss_values = history.history['val_loss']

epochs = range(1, len(loss_values) + 1)

plt.plot(epochs, loss_values, 'r', label = 'Training loss')
plt.plot(epochs, val_loss_values, 'b', label = 'Validation loss')

plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')

plt.legend()

fig = plt.gcf()
fig.savefig('results/CIFGAR10/yes/train_val_loss.png')

plt.show()
```

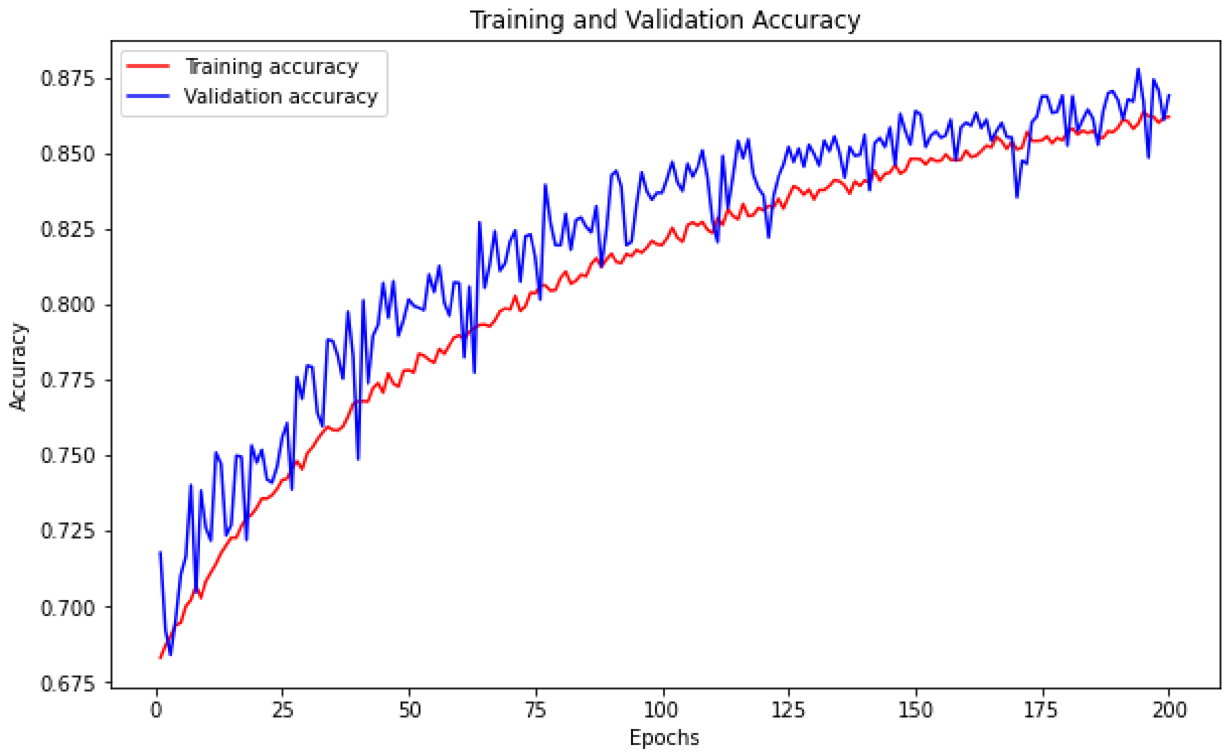


Plot Training and Validation accuracy

In [36]:
```
plt.clf()

plt.figure(figsize = (10, 6))

acc_values = history.history['accuracy']
val_acc_values = history.history['val_accuracy']

epochs = range(1, len(loss_values) + 1)
```

```python
plt.plot(epochs, acc_values, 'r', label = 'Training accuracy')
plt.plot(epochs, val_acc_values, 'b', label = 'Validation accuracy')

plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')

plt.legend()

fig = plt.gcf()
fig.savefig('results/CIFGAR10/yes/train_val_accuracy.png')

plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



Evaluate the Model

In [37]:
```python
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 1s 5ms/step - loss: 0.4343 - accuracy: 0.8
583
```

In [38]:
```python
print(f'Test accuracy: {test_acc * 100:.1f}%')
print(f'Test loss: {test_loss:.3f}')
```

```
Test accuracy: 85.8%
Test loss: 0.434
```

Predicting the test data

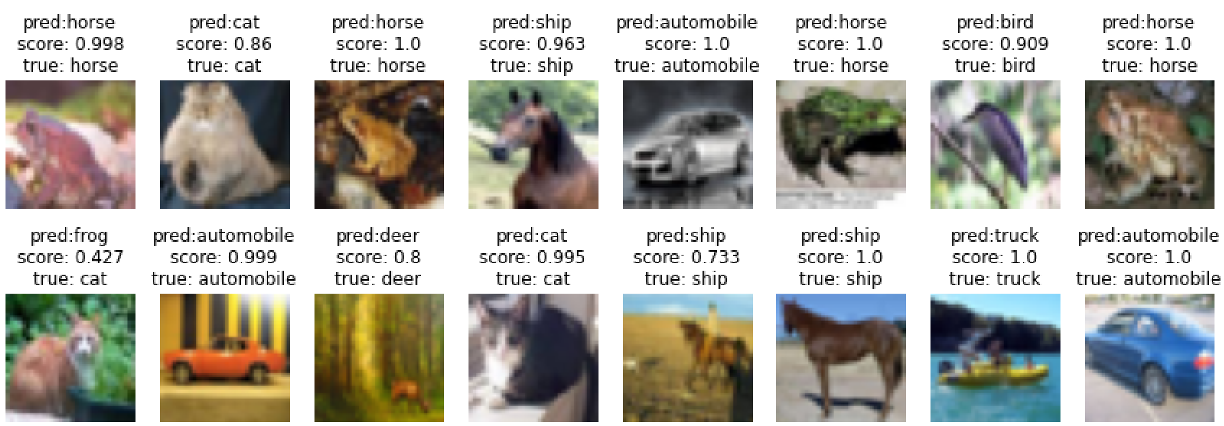
In [39]:
```python
label_pred_test = model.predict(test_images)
label_pred_test_classes = np.argmax(label_pred_test, axis = 1)
label_pred_test_max_probability = np.max(label_pred_test, axis = 1)
```

```
313/313 [==============================] - 1s 3ms/step
```

```
In [40]: # Reverse test_labels from categorical
         test_labels = np.argmax(test_labels, axis = 1)
```

Visualize predictions

```
In [41]: cols = 8
         rows = 2

         fig = plt.figure(figsize = (2 * cols - 1, 3 * rows - 1))

         for i in range(cols):
           for j in range(rows):
             random_index = np.random.randint(0, len(test_labels))

             ax = fig.add_subplot(rows, cols, i * rows + j + 1)

             ax.grid('off')
             ax.axis('off')

             ax.imshow(test_images[random_index, :])

             pred_label = cifar10_classes[label_pred_test_classes[random_index]]
             pred_probability = label_pred_test_max_probability[random_index]

             true_label = cifar10_classes[test_labels[random_index]]

             ax.set_title(f'pred:{pred_label}\nscore: {pred_probability:.3}\ntrue: {true_label
```



Save Model and Results

```
In [42]: model.save('results/CIFGAR10/yes/mnist.h5', history)
```

```
In [43]: pickle.dump({'test_accuracy': test_acc,
                      'test_loss': test_loss,
                      'history_dict': history.history},
                     open("results/CIFGAR10/yes/training_metrics", "wb"))
```