

In [15]:

```
"""
erick heredia

This program was written by josheph lee wei en on medium
this program is being used for learning

link to website and github

https://github.com/josephlee94/intuitive-deep-learning

https://medium.com/intuitive-deep-learning/build-your-first-neural-network-to-predi
"""


```

Out[15]: '\n\nerick heredia\n\nThis program was written by josheph lee wei en on medium \n\nlink to website and github\n\nhttps://github.com/josephlee94/intuitive-deep-learning\n\nhttps://medium.com/intuitive-deep-learning/build-your-first-neural-network-to-predict-house-prices-with-keras-eb5db60232c\n\n'

In [2]:

```
import keras
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from skimage.transform import resize
import numpy as np
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras import regularizers
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
```

In [3]:

```
df = pd.read_csv("housepricedata.csv")

data = df.values

X = data[:,0:10]
Y = data[:,10]

scaler = preprocessing.MinMaxScaler()
X_scale = scaler.fit_transform(X)

X_train, X_val_and_test, Y_train, Y_val_and_test = train_test_split(X_scale, Y, test_size=0.2)
X_val, X_test, Y_val, Y_test = train_test_split(X_val_and_test, Y_val_and_test, test_size=0.5)

print(X_train.shape, X_val.shape, Y_train.shape, Y_val.shape, X_test.shape, Y_test.shape)

model = Sequential([
    Dense(32, activation="relu", input_shape=(10,)),
    Dense(32, activation="relu"),
```

```
Dense(1, activation="sigmoid"),
])

model.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])

hist = model.fit(X_train, Y_train, batch_size=32, epochs=100, validation_data=(X_val,
model.evaluate(X_test, Y_test)[1]

model_2 = Sequential([
    Dense(1000, activation='relu', input_shape=(10,)),
    Dense(1000, activation='relu'),
    Dense(1000, activation='relu')])

model_2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
hist_2 = model_2.fit(X_train, Y_train, batch_size=32, epochs=100, validation_data=(X_val,
model_3 = Sequential([
    Dense(1000, activation='relu', kernel_regularizer=regularizers.l2(0.01), input_
    Dense(1000, activation='relu', kernel_regularizer=regularizers.l2(0.01)),
    Dropout(0.3),
    Dense(1000, activation='relu', kernel_regularizer=regularizers.l2(0.01)),
    Dropout(0.3),
    Dense(1000, activation='relu', kernel_regularizer=regularizers.l2(0.01)),
    Dropout(0.3),
    Dense(1, activation='sigmoid', kernel_regularizer=regularizers.l2(0.01)),
])

model_3.compile(optimizer='adam',
                 loss='binary_crossentropy',
                 metrics=['accuracy']))
hist_3 = model_3.fit(X_train, Y_train,
                      batch_size=32, epochs=100,
                      validation_data=(X_val, Y_val))
```

(1022, 10) (219, 10) (1022,) (219,) (219, 10) (219,)
Epoch 1/100

C:\Users\erick\anaconda3\envs\lol\Lib\site-packages\keras\src\layers\core\dense.py:8
6: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When u
sing Sequential models, prefer using an `Input(shape)` object as the first layer in
the model instead.

super().__init__(activity_regularizer=activity_regularizer, **kwargs)

```
32/32 ━━━━━━━━ 1s 5ms/step - accuracy: 0.4381 - loss: 0.7049 - val_accuracy: 0.5205 - val_loss: 0.6924
Epoch 2/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.5278 - loss: 0.6928 - val_accuracy: 0.5982 - val_loss: 0.6806
Epoch 3/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.6356 - loss: 0.6789 - val_accuracy: 0.7717 - val_loss: 0.6714
Epoch 4/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.7660 - loss: 0.6712 - val_accuracy: 0.8265 - val_loss: 0.6645
Epoch 5/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8313 - loss: 0.6634 - val_accuracy: 0.8493 - val_loss: 0.6583
Epoch 6/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8276 - loss: 0.6597 - val_accuracy: 0.8493 - val_loss: 0.6524
Epoch 7/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8164 - loss: 0.6497 - val_accuracy: 0.8539 - val_loss: 0.6463
Epoch 8/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8152 - loss: 0.6448 - val_accuracy: 0.8493 - val_loss: 0.6402
Epoch 9/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8159 - loss: 0.6407 - val_accuracy: 0.8493 - val_loss: 0.6338
Epoch 10/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8098 - loss: 0.6331 - val_accuracy: 0.8447 - val_loss: 0.6272
Epoch 11/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8038 - loss: 0.6271 - val_accuracy: 0.8265 - val_loss: 0.6203
Epoch 12/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8125 - loss: 0.6165 - val_accuracy: 0.8356 - val_loss: 0.6132
Epoch 13/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8152 - loss: 0.6094 - val_accuracy: 0.8356 - val_loss: 0.6059
Epoch 14/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8158 - loss: 0.6035 - val_accuracy: 0.8356 - val_loss: 0.5983
Epoch 15/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8277 - loss: 0.5943 - val_accuracy: 0.8356 - val_loss: 0.5905
Epoch 16/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8406 - loss: 0.5842 - val_accuracy: 0.8356 - val_loss: 0.5823
Epoch 17/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8360 - loss: 0.5778 - val_accuracy: 0.8447 - val_loss: 0.5739
Epoch 18/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8426 - loss: 0.5729 - val_accuracy: 0.8447 - val_loss: 0.5653
Epoch 19/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8551 - loss: 0.5591 - val_accuracy: 0.8311 - val_loss: 0.5567
```

```
Epoch 20/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8493 - loss: 0.5530 - val_accuracy: 0.8402 - val_loss: 0.5474
Epoch 21/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8699 - loss: 0.5380 - val_accuracy: 0.8447 - val_loss: 0.5380
Epoch 22/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8459 - loss: 0.5390 - val_accuracy: 0.8447 - val_loss: 0.5282
Epoch 23/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8693 - loss: 0.5210 - val_accuracy: 0.8447 - val_loss: 0.5184
Epoch 24/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8591 - loss: 0.5198 - val_accuracy: 0.8356 - val_loss: 0.5091
Epoch 25/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8612 - loss: 0.5075 - val_accuracy: 0.8402 - val_loss: 0.4995
Epoch 26/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8593 - loss: 0.4837 - val_accuracy: 0.8356 - val_loss: 0.4894
Epoch 27/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8675 - loss: 0.4895 - val_accuracy: 0.8356 - val_loss: 0.4799
Epoch 28/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8644 - loss: 0.4803 - val_accuracy: 0.8447 - val_loss: 0.4712
Epoch 29/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8582 - loss: 0.4733 - val_accuracy: 0.8447 - val_loss: 0.4618
Epoch 30/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8907 - loss: 0.4396 - val_accuracy: 0.8356 - val_loss: 0.4519
Epoch 31/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8857 - loss: 0.4326 - val_accuracy: 0.8311 - val_loss: 0.4432
Epoch 32/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8583 - loss: 0.4373 - val_accuracy: 0.8402 - val_loss: 0.4355
Epoch 33/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8697 - loss: 0.4277 - val_accuracy: 0.8402 - val_loss: 0.4278
Epoch 34/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8765 - loss: 0.4096 - val_accuracy: 0.8311 - val_loss: 0.4195
Epoch 35/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8983 - loss: 0.3885 - val_accuracy: 0.8356 - val_loss: 0.4118
Epoch 36/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8786 - loss: 0.3971 - val_accuracy: 0.8356 - val_loss: 0.4063
Epoch 37/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8755 - loss: 0.4017 - val_accuracy: 0.8356 - val_loss: 0.3995
Epoch 38/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8753 - loss: 0.3911 - val_accuracy:
```

```
acy: 0.8402 - val_loss: 0.3955
Epoch 39/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8861 - loss: 0.3714 - val_accur
acy: 0.8402 - val_loss: 0.3894
Epoch 40/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8774 - loss: 0.3651 - val_accur
acy: 0.8356 - val_loss: 0.3849
Epoch 41/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8817 - loss: 0.3599 - val_accur
acy: 0.8311 - val_loss: 0.3774
Epoch 42/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8866 - loss: 0.3536 - val_accur
acy: 0.8356 - val_loss: 0.3733
Epoch 43/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8637 - loss: 0.3609 - val_accur
acy: 0.8402 - val_loss: 0.3716
Epoch 44/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8826 - loss: 0.3487 - val_accur
acy: 0.8402 - val_loss: 0.3672
Epoch 45/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8802 - loss: 0.3479 - val_accur
acy: 0.8493 - val_loss: 0.3659
Epoch 46/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8710 - loss: 0.3445 - val_accur
acy: 0.8402 - val_loss: 0.3603
Epoch 47/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8601 - loss: 0.3606 - val_accur
acy: 0.8356 - val_loss: 0.3546
Epoch 48/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8891 - loss: 0.3366 - val_accur
acy: 0.8402 - val_loss: 0.3546
Epoch 49/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8779 - loss: 0.3340 - val_accur
acy: 0.8402 - val_loss: 0.3489
Epoch 50/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8620 - loss: 0.3418 - val_accur
acy: 0.8402 - val_loss: 0.3458
Epoch 51/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8891 - loss: 0.3225 - val_accur
acy: 0.8402 - val_loss: 0.3448
Epoch 52/100
32/32 ━━━━━━━━ 0s 2ms/step - accuracy: 0.8473 - loss: 0.3476 - val_accur
acy: 0.8402 - val_loss: 0.3436
Epoch 53/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8732 - loss: 0.3223 - val_accur
acy: 0.8402 - val_loss: 0.3414
Epoch 54/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.9059 - loss: 0.2831 - val_accur
acy: 0.8447 - val_loss: 0.3348
Epoch 55/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8791 - loss: 0.3212 - val_accur
acy: 0.8356 - val_loss: 0.3366
Epoch 56/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8920 - loss: 0.2973 - val_accur
acy: 0.8356 - val_loss: 0.3344
Epoch 57/100
```

```
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8677 - loss: 0.3137 - val_accuracy: 0.8402 - val_loss: 0.3319
Epoch 58/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8617 - loss: 0.3298 - val_accuracy: 0.8447 - val_loss: 0.3340
Epoch 59/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8636 - loss: 0.3343 - val_accuracy: 0.8493 - val_loss: 0.3276
Epoch 60/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8819 - loss: 0.3060 - val_accuracy: 0.8356 - val_loss: 0.3294
Epoch 61/100
32/32 ━━━━━━━━ 0s 2ms/step - accuracy: 0.8685 - loss: 0.3136 - val_accuracy: 0.8356 - val_loss: 0.3268
Epoch 62/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8613 - loss: 0.3130 - val_accuracy: 0.8447 - val_loss: 0.3223
Epoch 63/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8747 - loss: 0.3144 - val_accuracy: 0.8447 - val_loss: 0.3275
Epoch 64/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8896 - loss: 0.2875 - val_accuracy: 0.8447 - val_loss: 0.3209
Epoch 65/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8777 - loss: 0.3118 - val_accuracy: 0.8447 - val_loss: 0.3195
Epoch 66/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8805 - loss: 0.2971 - val_accuracy: 0.8447 - val_loss: 0.3176
Epoch 67/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8694 - loss: 0.3155 - val_accuracy: 0.8447 - val_loss: 0.3173
Epoch 68/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8715 - loss: 0.3077 - val_accuracy: 0.8493 - val_loss: 0.3166
Epoch 69/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8868 - loss: 0.2911 - val_accuracy: 0.8539 - val_loss: 0.3118
Epoch 70/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8814 - loss: 0.2833 - val_accuracy: 0.8539 - val_loss: 0.3105
Epoch 71/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8889 - loss: 0.2809 - val_accuracy: 0.8447 - val_loss: 0.3118
Epoch 72/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8855 - loss: 0.2890 - val_accuracy: 0.8447 - val_loss: 0.3092
Epoch 73/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.9001 - loss: 0.2898 - val_accuracy: 0.8493 - val_loss: 0.3101
Epoch 74/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8796 - loss: 0.2959 - val_accuracy: 0.8493 - val_loss: 0.3059
Epoch 75/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8709 - loss: 0.3060 - val_accuracy: 0.8493 - val_loss: 0.3078
```

```
Epoch 76/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8844 - loss: 0.2848 - val_accuracy: 0.8539 - val_loss: 0.3054
Epoch 77/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8646 - loss: 0.3243 - val_accuracy: 0.8493 - val_loss: 0.3080
Epoch 78/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8804 - loss: 0.2954 - val_accuracy: 0.8630 - val_loss: 0.3003
Epoch 79/100
32/32 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8961 - loss: 0.2601 - val_accuracy: 0.8493 - val_loss: 0.3015
Epoch 80/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8980 - loss: 0.2728 - val_accuracy: 0.8630 - val_loss: 0.2992
Epoch 81/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8947 - loss: 0.2767 - val_accuracy: 0.8539 - val_loss: 0.3053
Epoch 82/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8936 - loss: 0.2726 - val_accuracy: 0.8584 - val_loss: 0.2995
Epoch 83/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8901 - loss: 0.2725 - val_accuracy: 0.8539 - val_loss: 0.3047
Epoch 84/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8875 - loss: 0.2778 - val_accuracy: 0.8539 - val_loss: 0.3026
Epoch 85/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8868 - loss: 0.2873 - val_accuracy: 0.8630 - val_loss: 0.2995
Epoch 86/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8883 - loss: 0.2886 - val_accuracy: 0.8630 - val_loss: 0.2929
Epoch 87/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8762 - loss: 0.2993 - val_accuracy: 0.8630 - val_loss: 0.2936
Epoch 88/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8800 - loss: 0.3166 - val_accuracy: 0.8630 - val_loss: 0.2912
Epoch 89/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8869 - loss: 0.2747 - val_accuracy: 0.8721 - val_loss: 0.2946
Epoch 90/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8982 - loss: 0.2635 - val_accuracy: 0.8721 - val_loss: 0.2930
Epoch 91/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9069 - loss: 0.2605 - val_accuracy: 0.8676 - val_loss: 0.2912
Epoch 92/100
32/32 ━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8948 - loss: 0.2490 - val_accuracy: 0.8721 - val_loss: 0.2932
Epoch 93/100
32/32 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8971 - loss: 0.2925 - val_accuracy: 0.8721 - val_loss: 0.2918
Epoch 94/100
32/32 ━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8799 - loss: 0.2950 - val_accuracy:
```

```
acy: 0.8721 - val_loss: 0.2878
Epoch 95/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8951 - loss: 0.2639 - val_accur
acy: 0.8721 - val_loss: 0.2874
Epoch 96/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8966 - loss: 0.2668 - val_accur
acy: 0.8721 - val_loss: 0.2883
Epoch 97/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8940 - loss: 0.2754 - val_accur
acy: 0.8721 - val_loss: 0.2896
Epoch 98/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8802 - loss: 0.2805 - val_accur
acy: 0.8721 - val_loss: 0.2868
Epoch 99/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8973 - loss: 0.2620 - val_accur
acy: 0.8721 - val_loss: 0.2880
Epoch 100/100
32/32 ━━━━━━━━ 0s 1ms/step - accuracy: 0.8970 - loss: 0.2766 - val_accur
acy: 0.8721 - val_loss: 0.2840
Epoch 1/100
32/32 ━━━━━━ 2s 24ms/step - accuracy: 0.6487 - loss: 0.5887 - val_accur
acy: 0.8493 - val_loss: 0.3468
Epoch 2/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8631 - loss: 0.3518 - val_accur
acy: 0.8311 - val_loss: 0.3759
Epoch 3/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8664 - loss: 0.3354 - val_accur
acy: 0.8630 - val_loss: 0.2762
Epoch 4/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.9013 - loss: 0.2637 - val_accur
acy: 0.8858 - val_loss: 0.2783
Epoch 5/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8839 - loss: 0.2714 - val_accur
acy: 0.8858 - val_loss: 0.2542
Epoch 6/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8622 - loss: 0.2857 - val_accur
acy: 0.8630 - val_loss: 0.3087
Epoch 7/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.9026 - loss: 0.2998 - val_accur
acy: 0.8767 - val_loss: 0.2391
Epoch 8/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.9037 - loss: 0.2382 - val_accur
acy: 0.8995 - val_loss: 0.2406
Epoch 9/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8909 - loss: 0.2351 - val_accur
acy: 0.9132 - val_loss: 0.2146
Epoch 10/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.9034 - loss: 0.2552 - val_accur
acy: 0.8813 - val_loss: 0.3104
Epoch 11/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.9048 - loss: 0.2695 - val_accur
acy: 0.8995 - val_loss: 0.2408
Epoch 12/100
32/32 ━━━━━━ 1s 18ms/step - accuracy: 0.8976 - loss: 0.2359 - val_accur
acy: 0.8493 - val_loss: 0.3667
Epoch 13/100
```

```
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.8947 - loss: 0.2544 - val_accuracy: 0.8767 - val_loss: 0.2711
Epoch 14/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9288 - loss: 0.1847 - val_accuracy: 0.9132 - val_loss: 0.2168
Epoch 15/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9110 - loss: 0.2096 - val_accuracy: 0.8630 - val_loss: 0.2715
Epoch 16/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9198 - loss: 0.1992 - val_accuracy: 0.9087 - val_loss: 0.2570
Epoch 17/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9124 - loss: 0.2095 - val_accuracy: 0.8904 - val_loss: 0.2566
Epoch 18/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9080 - loss: 0.2021 - val_accuracy: 0.8630 - val_loss: 0.3002
Epoch 19/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9260 - loss: 0.1849 - val_accuracy: 0.9132 - val_loss: 0.2359
Epoch 20/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9154 - loss: 0.1936 - val_accuracy: 0.9132 - val_loss: 0.2176
Epoch 21/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9232 - loss: 0.1760 - val_accuracy: 0.8950 - val_loss: 0.2445
Epoch 22/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9100 - loss: 0.1916 - val_accuracy: 0.8721 - val_loss: 0.3106
Epoch 23/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9170 - loss: 0.2192 - val_accuracy: 0.9132 - val_loss: 0.2336
Epoch 24/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9056 - loss: 0.2169 - val_accuracy: 0.8904 - val_loss: 0.2939
Epoch 25/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9262 - loss: 0.1646 - val_accuracy: 0.8676 - val_loss: 0.2855
Epoch 26/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9335 - loss: 0.1668 - val_accuracy: 0.8584 - val_loss: 0.2864
Epoch 27/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9073 - loss: 0.2132 - val_accuracy: 0.9041 - val_loss: 0.2248
Epoch 28/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9261 - loss: 0.1728 - val_accuracy: 0.8721 - val_loss: 0.2689
Epoch 29/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9379 - loss: 0.1584 - val_accuracy: 0.8767 - val_loss: 0.3580
Epoch 30/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9207 - loss: 0.1882 - val_accuracy: 0.8493 - val_loss: 0.3438
Epoch 31/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9133 - loss: 0.1862 - val_accuracy: 0.8904 - val_loss: 0.2935
```

```
Epoch 32/100
32/32 1s 19ms/step - accuracy: 0.9199 - loss: 0.1609 - val_accuracy: 0.8584 - val_loss: 0.3220
Epoch 33/100
32/32 1s 18ms/step - accuracy: 0.9303 - loss: 0.1671 - val_accuracy: 0.8813 - val_loss: 0.2697
Epoch 34/100
32/32 1s 18ms/step - accuracy: 0.9320 - loss: 0.1586 - val_accuracy: 0.8721 - val_loss: 0.2890
Epoch 35/100
32/32 1s 18ms/step - accuracy: 0.9369 - loss: 0.1585 - val_accuracy: 0.8630 - val_loss: 0.3069
Epoch 36/100
32/32 1s 18ms/step - accuracy: 0.9235 - loss: 0.1979 - val_accuracy: 0.8630 - val_loss: 0.3079
Epoch 37/100
32/32 1s 18ms/step - accuracy: 0.9184 - loss: 0.1706 - val_accuracy: 0.8721 - val_loss: 0.3350
Epoch 38/100
32/32 1s 18ms/step - accuracy: 0.9163 - loss: 0.1879 - val_accuracy: 0.8858 - val_loss: 0.3110
Epoch 39/100
32/32 1s 18ms/step - accuracy: 0.9363 - loss: 0.1820 - val_accuracy: 0.8676 - val_loss: 0.3781
Epoch 40/100
32/32 1s 18ms/step - accuracy: 0.9380 - loss: 0.1762 - val_accuracy: 0.8995 - val_loss: 0.3072
Epoch 41/100
32/32 1s 18ms/step - accuracy: 0.9262 - loss: 0.1766 - val_accuracy: 0.8904 - val_loss: 0.3008
Epoch 42/100
32/32 1s 18ms/step - accuracy: 0.9333 - loss: 0.1310 - val_accuracy: 0.8813 - val_loss: 0.3092
Epoch 43/100
32/32 1s 18ms/step - accuracy: 0.9320 - loss: 0.1370 - val_accuracy: 0.8584 - val_loss: 0.3855
Epoch 44/100
32/32 1s 18ms/step - accuracy: 0.9348 - loss: 0.1421 - val_accuracy: 0.8721 - val_loss: 0.3183
Epoch 45/100
32/32 1s 18ms/step - accuracy: 0.9220 - loss: 0.1530 - val_accuracy: 0.8858 - val_loss: 0.3475
Epoch 46/100
32/32 1s 18ms/step - accuracy: 0.9332 - loss: 0.1490 - val_accuracy: 0.8721 - val_loss: 0.3247
Epoch 47/100
32/32 1s 18ms/step - accuracy: 0.9269 - loss: 0.1564 - val_accuracy: 0.8767 - val_loss: 0.3683
Epoch 48/100
32/32 1s 18ms/step - accuracy: 0.9418 - loss: 0.1327 - val_accuracy: 0.8995 - val_loss: 0.3631
Epoch 49/100
32/32 1s 18ms/step - accuracy: 0.9269 - loss: 0.1852 - val_accuracy: 0.8447 - val_loss: 0.5032
Epoch 50/100
32/32 1s 18ms/step - accuracy: 0.9338 - loss: 0.1641 - val_accuracy:
```

```
racy: 0.8904 - val_loss: 0.3046
Epoch 51/100
32/32 1s 18ms/step - accuracy: 0.9254 - loss: 0.1606 - val_accu
racy: 0.8767 - val_loss: 0.3442
Epoch 52/100
32/32 1s 17ms/step - accuracy: 0.9290 - loss: 0.1570 - val_accu
racy: 0.8813 - val_loss: 0.2898
Epoch 53/100
32/32 1s 18ms/step - accuracy: 0.9413 - loss: 0.1354 - val_accu
racy: 0.8767 - val_loss: 0.3306
Epoch 54/100
32/32 1s 18ms/step - accuracy: 0.9441 - loss: 0.1287 - val_accu
racy: 0.8721 - val_loss: 0.3545
Epoch 55/100
32/32 1s 18ms/step - accuracy: 0.9457 - loss: 0.1408 - val_accu
racy: 0.8858 - val_loss: 0.3479
Epoch 56/100
32/32 1s 18ms/step - accuracy: 0.9477 - loss: 0.1300 - val_accu
racy: 0.8995 - val_loss: 0.3959
Epoch 59/100
32/32 1s 18ms/step - accuracy: 0.8957 - loss: 0.2239 - val_accu
racy: 0.8584 - val_loss: 0.3447
Epoch 60/100
32/32 1s 18ms/step - accuracy: 0.9238 - loss: 0.1818 - val_accu
racy: 0.8858 - val_loss: 0.3175
Epoch 61/100
32/32 1s 18ms/step - accuracy: 0.9281 - loss: 0.1787 - val_accu
racy: 0.9041 - val_loss: 0.3069
Epoch 62/100
32/32 1s 18ms/step - accuracy: 0.9279 - loss: 0.1451 - val_accu
racy: 0.8584 - val_loss: 0.4885
Epoch 63/100
32/32 1s 18ms/step - accuracy: 0.9477 - loss: 0.1178 - val_accu
racy: 0.8813 - val_loss: 0.3667
Epoch 64/100
32/32 1s 18ms/step - accuracy: 0.9320 - loss: 0.1319 - val_accu
racy: 0.8904 - val_loss: 0.3711
Epoch 65/100
32/32 1s 17ms/step - accuracy: 0.9430 - loss: 0.1250 - val_accu
racy: 0.8813 - val_loss: 0.3446
Epoch 66/100
32/32 1s 17ms/step - accuracy: 0.9491 - loss: 0.1287 - val_accu
racy: 0.8539 - val_loss: 0.4204
Epoch 67/100
32/32 1s 18ms/step - accuracy: 0.9420 - loss: 0.1379 - val_accu
racy: 0.8721 - val_loss: 0.3441
Epoch 68/100
32/32 1s 18ms/step - accuracy: 0.9316 - loss: 0.1483 - val_accu
racy: 0.8813 - val_loss: 0.3748
Epoch 69/100
```

```
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9392 - loss: 0.1368 - val_accuracy: 0.8721 - val_loss: 0.4223
Epoch 70/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9479 - loss: 0.1211 - val_accuracy: 0.8630 - val_loss: 0.4069
Epoch 71/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9540 - loss: 0.1052 - val_accuracy: 0.8584 - val_loss: 0.3290
Epoch 72/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9228 - loss: 0.1463 - val_accuracy: 0.8447 - val_loss: 0.4405
Epoch 73/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9499 - loss: 0.1182 - val_accuracy: 0.8584 - val_loss: 0.4340
Epoch 74/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9441 - loss: 0.1662 - val_accuracy: 0.8676 - val_loss: 0.3588
Epoch 75/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9478 - loss: 0.1502 - val_accuracy: 0.8950 - val_loss: 0.2995
Epoch 76/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9162 - loss: 0.1727 - val_accuracy: 0.8630 - val_loss: 0.3439
Epoch 77/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9292 - loss: 0.1446 - val_accuracy: 0.8630 - val_loss: 0.3603
Epoch 78/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9401 - loss: 0.1281 - val_accuracy: 0.8630 - val_loss: 0.5059
Epoch 79/100
32/32 ━━━━━━━━━━ 1s 17ms/step - accuracy: 0.9432 - loss: 0.1320 - val_accuracy: 0.8402 - val_loss: 0.5601
Epoch 80/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9532 - loss: 0.1205 - val_accuracy: 0.8539 - val_loss: 0.4366
Epoch 81/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9398 - loss: 0.1231 - val_accuracy: 0.8721 - val_loss: 0.4694
Epoch 82/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9408 - loss: 0.1275 - val_accuracy: 0.8493 - val_loss: 0.4039
Epoch 83/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9537 - loss: 0.1065 - val_accuracy: 0.8676 - val_loss: 0.4626
Epoch 84/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9542 - loss: 0.1063 - val_accuracy: 0.8676 - val_loss: 0.4934
Epoch 85/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9585 - loss: 0.0953 - val_accuracy: 0.8539 - val_loss: 0.5664
Epoch 86/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9598 - loss: 0.0952 - val_accuracy: 0.8630 - val_loss: 0.5230
Epoch 87/100
32/32 ━━━━━━━━━━ 1s 18ms/step - accuracy: 0.9446 - loss: 0.1044 - val_accuracy: 0.8721 - val_loss: 0.5248
```

```
Epoch 88/100
32/32 1s 18ms/step - accuracy: 0.9597 - loss: 0.1012 - val_accuracy: 0.8721 - val_loss: 0.5397
Epoch 89/100
32/32 1s 18ms/step - accuracy: 0.9482 - loss: 0.1263 - val_accuracy: 0.8539 - val_loss: 0.3623
Epoch 90/100
32/32 1s 18ms/step - accuracy: 0.9520 - loss: 0.1207 - val_accuracy: 0.8767 - val_loss: 0.4466
Epoch 91/100
32/32 1s 18ms/step - accuracy: 0.9601 - loss: 0.0938 - val_accuracy: 0.8584 - val_loss: 0.4859
Epoch 92/100
32/32 1s 17ms/step - accuracy: 0.9330 - loss: 0.1790 - val_accuracy: 0.8904 - val_loss: 0.3387
Epoch 93/100
32/32 1s 18ms/step - accuracy: 0.9315 - loss: 0.1372 - val_accuracy: 0.8858 - val_loss: 0.3318
Epoch 94/100
32/32 1s 18ms/step - accuracy: 0.9599 - loss: 0.1077 - val_accuracy: 0.8493 - val_loss: 0.4314
Epoch 95/100
32/32 1s 18ms/step - accuracy: 0.9519 - loss: 0.1202 - val_accuracy: 0.8721 - val_loss: 0.4524
Epoch 96/100
32/32 1s 18ms/step - accuracy: 0.9706 - loss: 0.0861 - val_accuracy: 0.8402 - val_loss: 0.4897
Epoch 97/100
32/32 1s 17ms/step - accuracy: 0.9483 - loss: 0.1123 - val_accuracy: 0.8950 - val_loss: 0.4225
Epoch 98/100
32/32 1s 18ms/step - accuracy: 0.9539 - loss: 0.1102 - val_accuracy: 0.8995 - val_loss: 0.6053
Epoch 99/100
32/32 1s 17ms/step - accuracy: 0.9522 - loss: 0.0973 - val_accuracy: 0.8676 - val_loss: 0.4083
Epoch 100/100
32/32 1s 18ms/step - accuracy: 0.9433 - loss: 0.1322 - val_accuracy: 0.8676 - val_loss: 0.5580
Epoch 1/100
32/32 2s 27ms/step - accuracy: 0.5303 - loss: 21.1171 - val_accuracy: 0.8037 - val_loss: 3.8875
Epoch 2/100
32/32 1s 23ms/step - accuracy: 0.8460 - loss: 2.4729 - val_accuracy: 0.8311 - val_loss: 0.6622
Epoch 3/100
32/32 1s 23ms/step - accuracy: 0.8738 - loss: 0.6002 - val_accuracy: 0.8447 - val_loss: 0.5161
Epoch 4/100
32/32 1s 23ms/step - accuracy: 0.8730 - loss: 0.4976 - val_accuracy: 0.8676 - val_loss: 0.4855
Epoch 5/100
32/32 1s 23ms/step - accuracy: 0.8686 - loss: 0.4924 - val_accuracy: 0.8539 - val_loss: 0.4990
Epoch 6/100
32/32 1s 23ms/step - accuracy: 0.8891 - loss: 0.4693 - val_accuracy:
```

```
racy: 0.8539 - val_loss: 0.4644
Epoch 7/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8836 - loss: 0.4669 - val_accu
racy: 0.8813 - val_loss: 0.4589
Epoch 8/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8726 - loss: 0.4749 - val_accu
racy: 0.8584 - val_loss: 0.4779
Epoch 9/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8837 - loss: 0.4593 - val_accu
racy: 0.8584 - val_loss: 0.4794
Epoch 10/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8744 - loss: 0.4682 - val_accu
racy: 0.8630 - val_loss: 0.4675
Epoch 11/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8936 - loss: 0.4363 - val_accu
racy: 0.8904 - val_loss: 0.4507
Epoch 12/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8736 - loss: 0.4392 - val_accu
racy: 0.8539 - val_loss: 0.4800
Epoch 13/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8940 - loss: 0.4448 - val_accu
racy: 0.8904 - val_loss: 0.4464
Epoch 14/100
32/32 ━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8709 - loss: 0.4556 - val_accu
racy: 0.8767 - val_loss: 0.4605
Epoch 15/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8926 - loss: 0.4346 - val_accu
racy: 0.8311 - val_loss: 0.5147
Epoch 16/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8847 - loss: 0.4494 - val_accu
racy: 0.8767 - val_loss: 0.4411
Epoch 17/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8898 - loss: 0.4488 - val_accu
racy: 0.8630 - val_loss: 0.4753
Epoch 18/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8814 - loss: 0.4615 - val_accu
racy: 0.8813 - val_loss: 0.4353
Epoch 19/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8872 - loss: 0.4428 - val_accu
racy: 0.8813 - val_loss: 0.4353
Epoch 20/100
32/32 ━━━━━━━━━━ 1s 22ms/step - accuracy: 0.9053 - loss: 0.4087 - val_accu
racy: 0.8082 - val_loss: 0.5460
Epoch 21/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8690 - loss: 0.4942 - val_accu
racy: 0.8402 - val_loss: 0.5007
Epoch 22/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8934 - loss: 0.4345 - val_accu
racy: 0.8904 - val_loss: 0.4405
Epoch 23/100
32/32 ━━━━━━━━━━ 1s 22ms/step - accuracy: 0.8998 - loss: 0.4084 - val_accu
racy: 0.8767 - val_loss: 0.4269
Epoch 24/100
32/32 ━━━━━━━━━━ 1s 22ms/step - accuracy: 0.8810 - loss: 0.4392 - val_accu
racy: 0.8767 - val_loss: 0.4266
Epoch 25/100
```

```
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8855 - loss: 0.4461 - val_accuracy: 0.8174 - val_loss: 0.5269
Epoch 26/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8879 - loss: 0.4456 - val_accuracy: 0.8584 - val_loss: 0.4742
Epoch 27/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8828 - loss: 0.4585 - val_accuracy: 0.8630 - val_loss: 0.4587
Epoch 28/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8885 - loss: 0.4332 - val_accuracy: 0.8858 - val_loss: 0.4294
Epoch 29/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8875 - loss: 0.4215 - val_accuracy: 0.8858 - val_loss: 0.4224
Epoch 30/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8865 - loss: 0.4450 - val_accuracy: 0.8904 - val_loss: 0.4394
Epoch 31/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8737 - loss: 0.4649 - val_accuracy: 0.8904 - val_loss: 0.4353
Epoch 32/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8620 - loss: 0.4649 - val_accuracy: 0.8858 - val_loss: 0.4238
Epoch 33/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8991 - loss: 0.4092 - val_accuracy: 0.8676 - val_loss: 0.4670
Epoch 34/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8811 - loss: 0.4420 - val_accuracy: 0.8858 - val_loss: 0.4205
Epoch 35/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8856 - loss: 0.4464 - val_accuracy: 0.8858 - val_loss: 0.4438
Epoch 36/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8845 - loss: 0.4268 - val_accuracy: 0.8995 - val_loss: 0.4314
Epoch 37/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8943 - loss: 0.4236 - val_accuracy: 0.8813 - val_loss: 0.4248
Epoch 38/100
32/32 ━━━━━━━━━━ 1s 22ms/step - accuracy: 0.8985 - loss: 0.4310 - val_accuracy: 0.8493 - val_loss: 0.4893
Epoch 39/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.9031 - loss: 0.4271 - val_accuracy: 0.8904 - val_loss: 0.4243
Epoch 40/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8778 - loss: 0.4294 - val_accuracy: 0.8858 - val_loss: 0.4204
Epoch 41/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8787 - loss: 0.4340 - val_accuracy: 0.8858 - val_loss: 0.4221
Epoch 42/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8705 - loss: 0.4732 - val_accuracy: 0.8858 - val_loss: 0.4215
Epoch 43/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8800 - loss: 0.4295 - val_accuracy: 0.8995 - val_loss: 0.4207
```

```
Epoch 44/100
32/32 1s 23ms/step - accuracy: 0.8740 - loss: 0.4691 - val_accuracy: 0.8813 - val_loss: 0.4239
Epoch 45/100
32/32 1s 22ms/step - accuracy: 0.8974 - loss: 0.4275 - val_accuracy: 0.8402 - val_loss: 0.4884
Epoch 46/100
32/32 1s 23ms/step - accuracy: 0.8757 - loss: 0.4426 - val_accuracy: 0.8767 - val_loss: 0.4339
Epoch 47/100
32/32 1s 23ms/step - accuracy: 0.8866 - loss: 0.4392 - val_accuracy: 0.8950 - val_loss: 0.4345
Epoch 48/100
32/32 1s 23ms/step - accuracy: 0.8898 - loss: 0.4393 - val_accuracy: 0.8813 - val_loss: 0.4345
Epoch 49/100
32/32 1s 23ms/step - accuracy: 0.8843 - loss: 0.4534 - val_accuracy: 0.8858 - val_loss: 0.4318
Epoch 50/100
32/32 1s 23ms/step - accuracy: 0.8745 - loss: 0.4581 - val_accuracy: 0.8767 - val_loss: 0.4275
Epoch 51/100
32/32 1s 23ms/step - accuracy: 0.9017 - loss: 0.4264 - val_accuracy: 0.8904 - val_loss: 0.4207
Epoch 52/100
32/32 1s 23ms/step - accuracy: 0.8548 - loss: 0.4620 - val_accuracy: 0.8813 - val_loss: 0.4424
Epoch 53/100
32/32 1s 23ms/step - accuracy: 0.8833 - loss: 0.4358 - val_accuracy: 0.8950 - val_loss: 0.4373
Epoch 54/100
32/32 1s 23ms/step - accuracy: 0.8731 - loss: 0.4599 - val_accuracy: 0.8767 - val_loss: 0.4303
Epoch 55/100
32/32 1s 23ms/step - accuracy: 0.8817 - loss: 0.4473 - val_accuracy: 0.8402 - val_loss: 0.5140
Epoch 56/100
32/32 1s 23ms/step - accuracy: 0.8787 - loss: 0.4565 - val_accuracy: 0.8904 - val_loss: 0.4311
Epoch 57/100
32/32 1s 23ms/step - accuracy: 0.8811 - loss: 0.4407 - val_accuracy: 0.8813 - val_loss: 0.4239
Epoch 58/100
32/32 1s 23ms/step - accuracy: 0.8762 - loss: 0.4394 - val_accuracy: 0.8813 - val_loss: 0.4507
Epoch 59/100
32/32 1s 23ms/step - accuracy: 0.8813 - loss: 0.4433 - val_accuracy: 0.8995 - val_loss: 0.4306
Epoch 60/100
32/32 1s 23ms/step - accuracy: 0.8765 - loss: 0.4585 - val_accuracy: 0.8767 - val_loss: 0.4506
Epoch 61/100
32/32 1s 23ms/step - accuracy: 0.8854 - loss: 0.4418 - val_accuracy: 0.8813 - val_loss: 0.4221
Epoch 62/100
32/32 1s 23ms/step - accuracy: 0.8892 - loss: 0.4224 - val_accuracy:
```

```
racy: 0.8767 - val_loss: 0.4232
Epoch 63/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8771 - loss: 0.4534 - val_accu
racy: 0.8950 - val_loss: 0.4390
Epoch 64/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8897 - loss: 0.4290 - val_accu
racy: 0.8950 - val_loss: 0.4197
Epoch 65/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8952 - loss: 0.4104 - val_accu
racy: 0.8813 - val_loss: 0.4528
Epoch 66/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8867 - loss: 0.4489 - val_accu
racy: 0.8904 - val_loss: 0.4183
Epoch 67/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8881 - loss: 0.4372 - val_accu
racy: 0.8904 - val_loss: 0.4232
Epoch 68/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8896 - loss: 0.4300 - val_accu
racy: 0.8813 - val_loss: 0.4210
Epoch 69/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8859 - loss: 0.4442 - val_accu
racy: 0.8904 - val_loss: 0.4222
Epoch 70/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8655 - loss: 0.4481 - val_accu
racy: 0.8858 - val_loss: 0.4169
Epoch 71/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8912 - loss: 0.4300 - val_accu
racy: 0.8630 - val_loss: 0.4593
Epoch 72/100
32/32 ━━━━━━━━ 1s 24ms/step - accuracy: 0.8866 - loss: 0.4198 - val_accu
racy: 0.8904 - val_loss: 0.4250
Epoch 73/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.9002 - loss: 0.4241 - val_accu
racy: 0.8904 - val_loss: 0.4215
Epoch 74/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8847 - loss: 0.4258 - val_accu
racy: 0.8858 - val_loss: 0.4253
Epoch 75/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8953 - loss: 0.4127 - val_accu
racy: 0.8813 - val_loss: 0.4452
Epoch 76/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8842 - loss: 0.4315 - val_accu
racy: 0.8813 - val_loss: 0.4296
Epoch 77/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.9137 - loss: 0.3998 - val_accu
racy: 0.8904 - val_loss: 0.4296
Epoch 78/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8949 - loss: 0.4279 - val_accu
racy: 0.8858 - val_loss: 0.4159
Epoch 79/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8979 - loss: 0.4149 - val_accu
racy: 0.8447 - val_loss: 0.4830
Epoch 80/100
32/32 ━━━━━━━━ 1s 23ms/step - accuracy: 0.8802 - loss: 0.4589 - val_accu
racy: 0.8950 - val_loss: 0.4362
Epoch 81/100
```

```
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8856 - loss: 0.4203 - val_accuracy: 0.7808 - val_loss: 0.6546
Epoch 82/100
32/32 ━━━━━━━━━━ 1s 22ms/step - accuracy: 0.8363 - loss: 0.5398 - val_accuracy: 0.8767 - val_loss: 0.4586
Epoch 83/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.9056 - loss: 0.3993 - val_accuracy: 0.8813 - val_loss: 0.4164
Epoch 84/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8936 - loss: 0.4197 - val_accuracy: 0.8721 - val_loss: 0.4537
Epoch 85/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.9007 - loss: 0.4225 - val_accuracy: 0.8904 - val_loss: 0.4207
Epoch 86/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8734 - loss: 0.4292 - val_accuracy: 0.8721 - val_loss: 0.4488
Epoch 87/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8653 - loss: 0.4610 - val_accuracy: 0.8995 - val_loss: 0.4297
Epoch 88/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8776 - loss: 0.4519 - val_accuracy: 0.8813 - val_loss: 0.4595
Epoch 89/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8969 - loss: 0.4175 - val_accuracy: 0.8676 - val_loss: 0.4542
Epoch 90/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8872 - loss: 0.4591 - val_accuracy: 0.8813 - val_loss: 0.4239
Epoch 91/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8632 - loss: 0.4594 - val_accuracy: 0.8858 - val_loss: 0.4192
Epoch 92/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8885 - loss: 0.4500 - val_accuracy: 0.8858 - val_loss: 0.4348
Epoch 93/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.9073 - loss: 0.4040 - val_accuracy: 0.8813 - val_loss: 0.4459
Epoch 94/100
32/32 ━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8912 - loss: 0.4414 - val_accuracy: 0.8813 - val_loss: 0.4356
Epoch 95/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8727 - loss: 0.4610 - val_accuracy: 0.8676 - val_loss: 0.4783
Epoch 96/100
32/32 ━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8668 - loss: 0.4812 - val_accuracy: 0.8858 - val_loss: 0.4275
Epoch 97/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8773 - loss: 0.4351 - val_accuracy: 0.8813 - val_loss: 0.4256
Epoch 98/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8825 - loss: 0.4217 - val_accuracy: 0.8676 - val_loss: 0.4538
Epoch 99/100
32/32 ━━━━━━━━━━ 1s 23ms/step - accuracy: 0.8836 - loss: 0.4319 - val_accuracy: 0.7991 - val_loss: 0.5566
```

```
Epoch 100/100
32/32 [=====] 1s 23ms/step - accuracy: 0.8978 - loss: 0.4271 - val_accuracy: 0.8858 - val_loss: 0.4188
```

```
In [13]: from keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

print('x_train shape:', x_train.shape)
print('y_train shape:', y_train.shape)

print(x_train[0])
img = plt.imshow(x_train[0])
print('The label is:', y_train[0])
img = plt.imshow(x_train[1])
print('The label is:', y_train[1])

y_train_one_hot = keras.utils.to_categorical(y_train, 10)
y_test_one_hot = keras.utils.to_categorical(y_test, 10)
print('The one hot label is:', y_train_one_hot[1])

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train = x_train / 255
x_test = x_test / 255
x_train[0]

model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(32,32,
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(10, activation='softmax'))
model.summary()

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

hist = model.fit(x_train, y_train_one_hot,
```

```
batch_size=32, epochs=20,  
validation_split=0.2)
```

```
x_train shape: (50000, 32, 32, 3)
y_train shape: (50000, 1)
[[[ 59  62  63]
 [ 43  46  45]
 [ 50  48  43]
 ...
 [158 132 108]
 [152 125 102]
 [148 124 103]]]

[[ 16  20  20]
 [  0   0   0]
 [ 18   8   0]
 ...
 [123  88  55]
 [119  83  50]
 [122  87  57]]]

[[ 25  24  21]
 [ 16    7   0]
 [ 49  27   8]
 ...
 [118  84  50]
 [120  84  50]
 [109  73  42]]]

...
[[208 170  96]
 [201 153  34]
 [198 161  26]
 ...
 [160 133  70]
 [ 56  31   7]
 [ 53  34  20]]]

[[180 139  96]
 [173 123  42]
 [186 144  30]
 ...
 [184 148  94]
 [ 97  62  34]
 [ 83  53  34]]]

[[177 144 116]
 [168 129  94]
 [179 142  87]
 ...
 [216 184 140]
 [151 118  84]
 [123  92  72]]]

The label is: [6]
The label is: [9]
The one hot label is: [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
Model: "sequential_5"
```

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 32, 32, 32)	896
conv2d_9 (Conv2D)	(None, 32, 32, 32)	9,248
max_pooling2d_4 (MaxPooling2D)	(None, 16, 16, 32)	0
dropout_10 (Dropout)	(None, 16, 16, 32)	0
conv2d_10 (Conv2D)	(None, 16, 16, 64)	18,496
conv2d_11 (Conv2D)	(None, 16, 16, 64)	36,928
max_pooling2d_5 (MaxPooling2D)	(None, 8, 8, 64)	0
dropout_11 (Dropout)	(None, 8, 8, 64)	0
flatten_2 (Flatten)	(None, 4096)	0
dense_17 (Dense)	(None, 512)	2,097,664
dropout_12 (Dropout)	(None, 512)	0
dense_18 (Dense)	(None, 10)	5,130

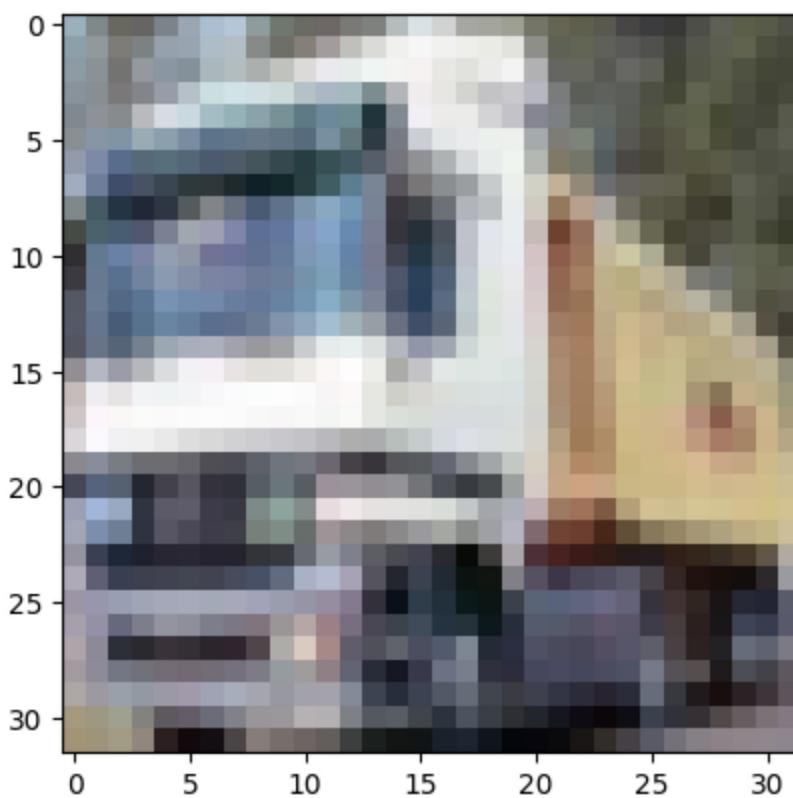
Total params: 2,168,362 (8.27 MB)

Trainable params: 2,168,362 (8.27 MB)

Non-trainable params: 0 (0.00 B)

Epoch 1/20
1250/1250 38s 29ms/step - accuracy: 0.3413 - loss: 1.7856 - val_accuracy: 0.5859 - val_loss: 1.1627
Epoch 2/20
1250/1250 35s 28ms/step - accuracy: 0.5831 - loss: 1.1767 - val_accuracy: 0.6552 - val_loss: 0.9593
Epoch 3/20
1250/1250 33s 27ms/step - accuracy: 0.6573 - loss: 0.9717 - val_accuracy: 0.7015 - val_loss: 0.8524
Epoch 4/20
1250/1250 33s 27ms/step - accuracy: 0.6939 - loss: 0.8613 - val_accuracy: 0.7144 - val_loss: 0.8119
Epoch 5/20
1250/1250 33s 27ms/step - accuracy: 0.7198 - loss: 0.7872 - val_accuracy: 0.7173 - val_loss: 0.8127
Epoch 6/20
1250/1250 33s 27ms/step - accuracy: 0.7464 - loss: 0.7169 - val_accuracy: 0.7503 - val_loss: 0.7270
Epoch 7/20
1250/1250 33s 27ms/step - accuracy: 0.7648 - loss: 0.6675 - val_accuracy: 0.7488 - val_loss: 0.7362
Epoch 8/20
1250/1250 34s 27ms/step - accuracy: 0.7771 - loss: 0.6302 - val_accuracy: 0.7590 - val_loss: 0.7002
Epoch 9/20
1250/1250 33s 26ms/step - accuracy: 0.7879 - loss: 0.5944 - val_accuracy: 0.7594 - val_loss: 0.6959
Epoch 10/20
1250/1250 33s 26ms/step - accuracy: 0.8031 - loss: 0.5548 - val_accuracy: 0.7764 - val_loss: 0.6781
Epoch 11/20
1250/1250 33s 26ms/step - accuracy: 0.8167 - loss: 0.5188 - val_accuracy: 0.7647 - val_loss: 0.6942
Epoch 12/20
1250/1250 33s 26ms/step - accuracy: 0.8213 - loss: 0.5027 - val_accuracy: 0.7705 - val_loss: 0.6914
Epoch 13/20
1250/1250 33s 26ms/step - accuracy: 0.8282 - loss: 0.4895 - val_accuracy: 0.7771 - val_loss: 0.6706
Epoch 14/20
1250/1250 33s 26ms/step - accuracy: 0.8381 - loss: 0.4610 - val_accuracy: 0.7657 - val_loss: 0.7138
Epoch 15/20
1250/1250 33s 27ms/step - accuracy: 0.8437 - loss: 0.4464 - val_accuracy: 0.7704 - val_loss: 0.7323
Epoch 16/20
1250/1250 33s 26ms/step - accuracy: 0.8439 - loss: 0.4467 - val_accuracy: 0.7628 - val_loss: 0.7130
Epoch 17/20
1250/1250 33s 27ms/step - accuracy: 0.8553 - loss: 0.4089 - val_accuracy: 0.7776 - val_loss: 0.6990
Epoch 18/20
1250/1250 33s 27ms/step - accuracy: 0.8585 - loss: 0.3999 - val_accuracy: 0.7806 - val_loss: 0.6784
Epoch 19/20
1250/1250 34s 27ms/step - accuracy: 0.8605 - loss: 0.3983 - val

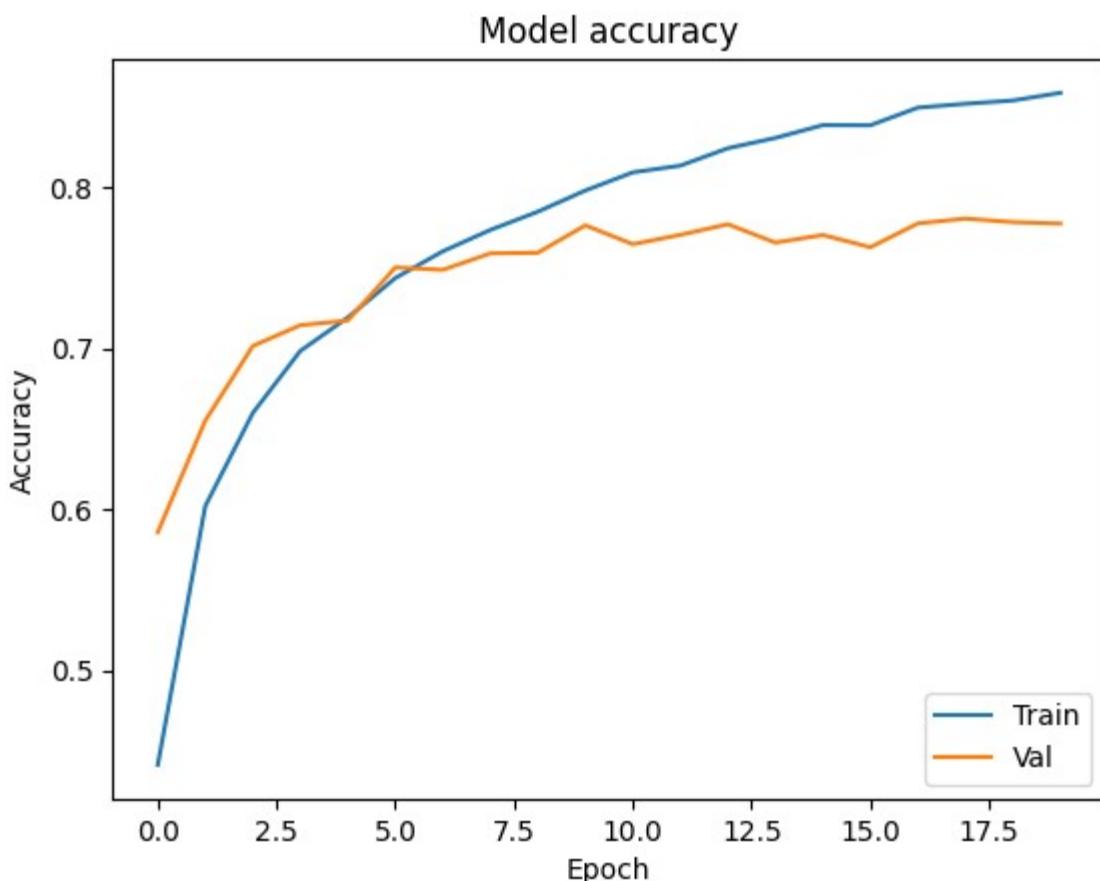
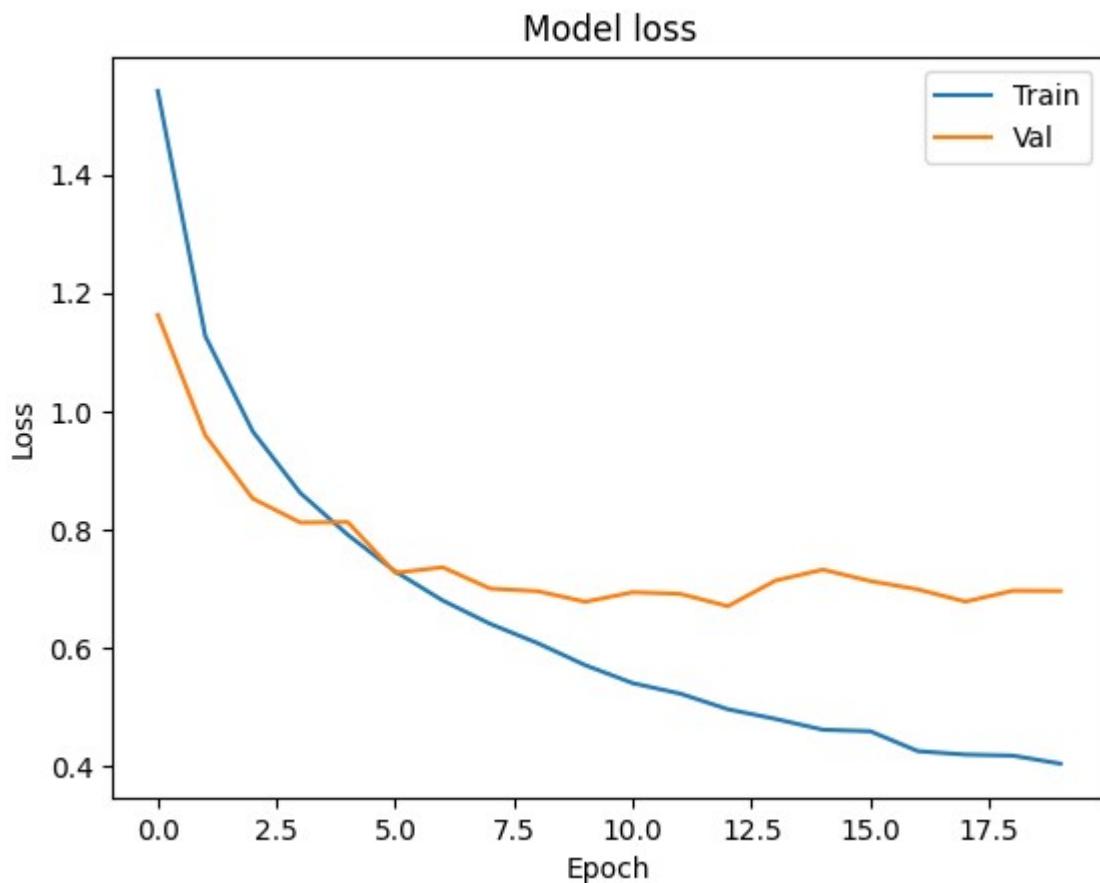
```
_accuracy: 0.7784 - val_loss: 0.6966
Epoch 20/20
1250/1250 ━━━━━━━━ 33s 26ms/step - accuracy: 0.8620 - loss: 0.3919 - val
_accuracy: 0.7775 - val_loss: 0.6963
```



```
In [14]: plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()

plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='lower right')
plt.show()

model.evaluate(x_test, y_test_one_hot)[1]
model.save('my_cifar10_model.h5')
```



313/313 ━━━━━━ 2s 6ms/step - accuracy: 0.7708 - loss: 0.7214

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

```
In [5]: my_image = plt.imread("cat.jpg")

my_image_resized = resize(my_image, (32,32,3))

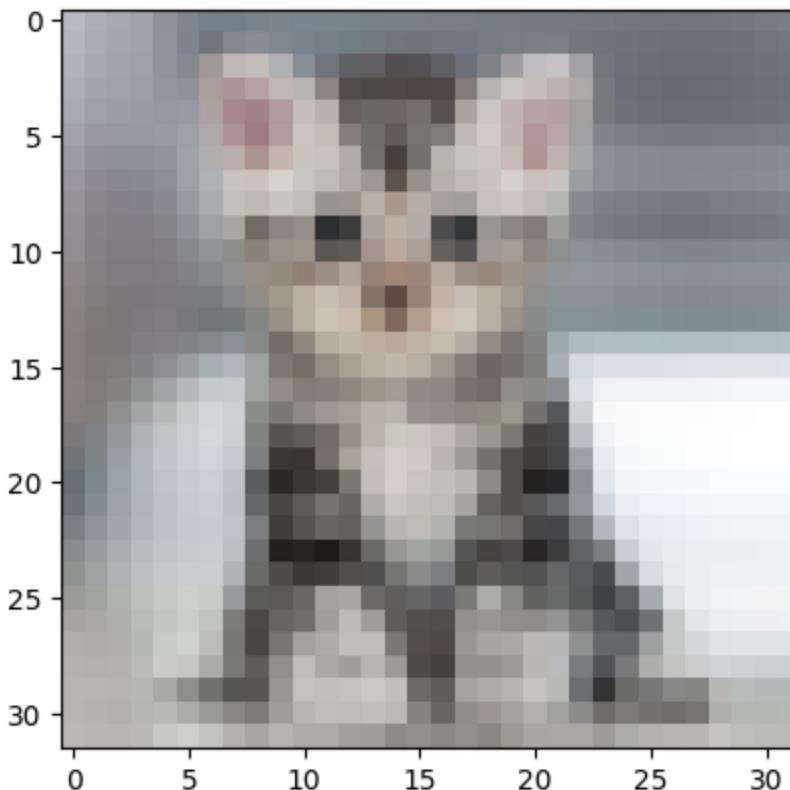
img = plt.imshow(my_image_resized)

probabilities = model.predict(np.array([my_image_resized, ]))

probabilities

number_to_class = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
index = np.argsort(probabilities[0,:])
print("Most likely class:", number_to_class[index[9]], "-- Probability:", probabilities[0][9])
print("Second most likely class:", number_to_class[index[8]], "-- Probability:", probabilities[0][8])
print("Third most likely class:", number_to_class[index[7]], "-- Probability:", probabilities[0][7])
print("Fourth most likely class:", number_to_class[index[6]], "-- Probability:", probabilities[0][6])
print("Fifth most likely class:", number_to_class[index[5]], "-- Probability:", probabilities[0][5])
```

1/1 ————— 0s 61ms/step
Most likely class: cat -- Probability: 0.4468107
Second most likely class: dog -- Probability: 0.3195176
Third most likely class: horse -- Probability: 0.11163055
Fourth most likely class: deer -- Probability: 0.09686223
Fifth most likely class: bird -- Probability: 0.020489702



```
In [10]: plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
```

```
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()

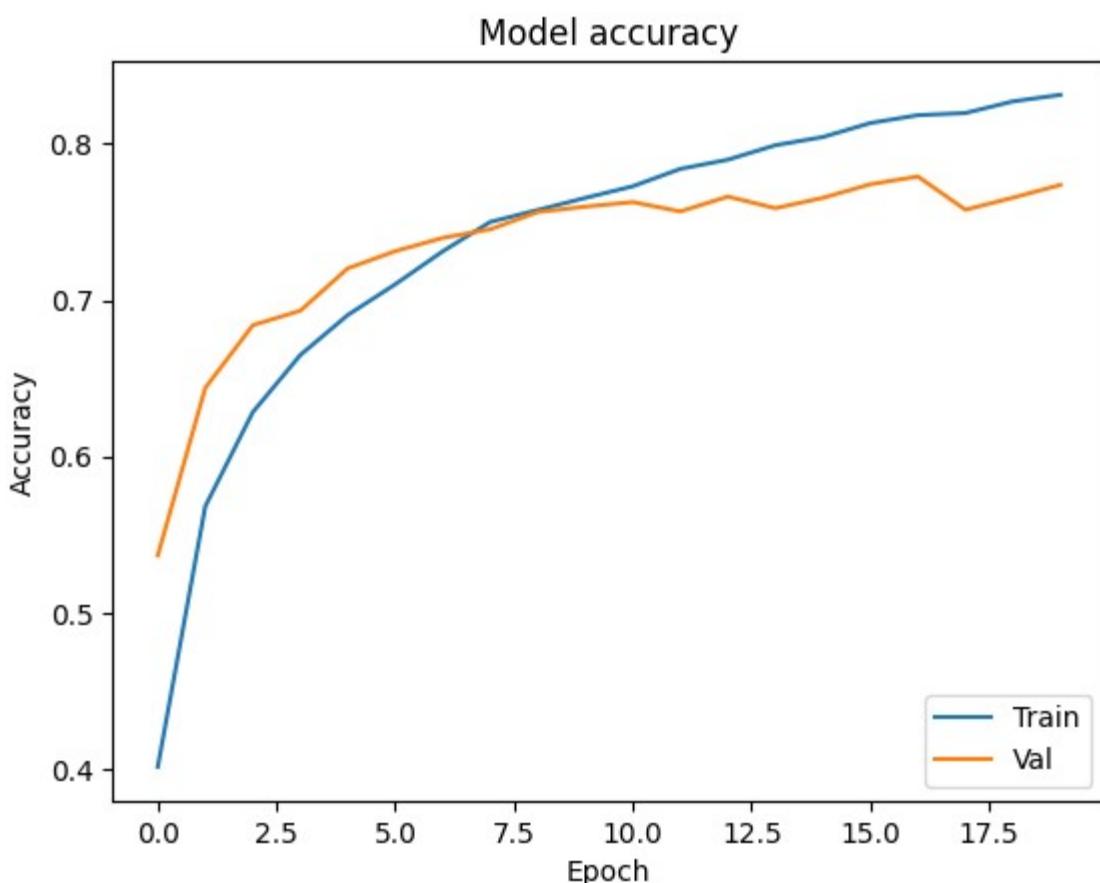
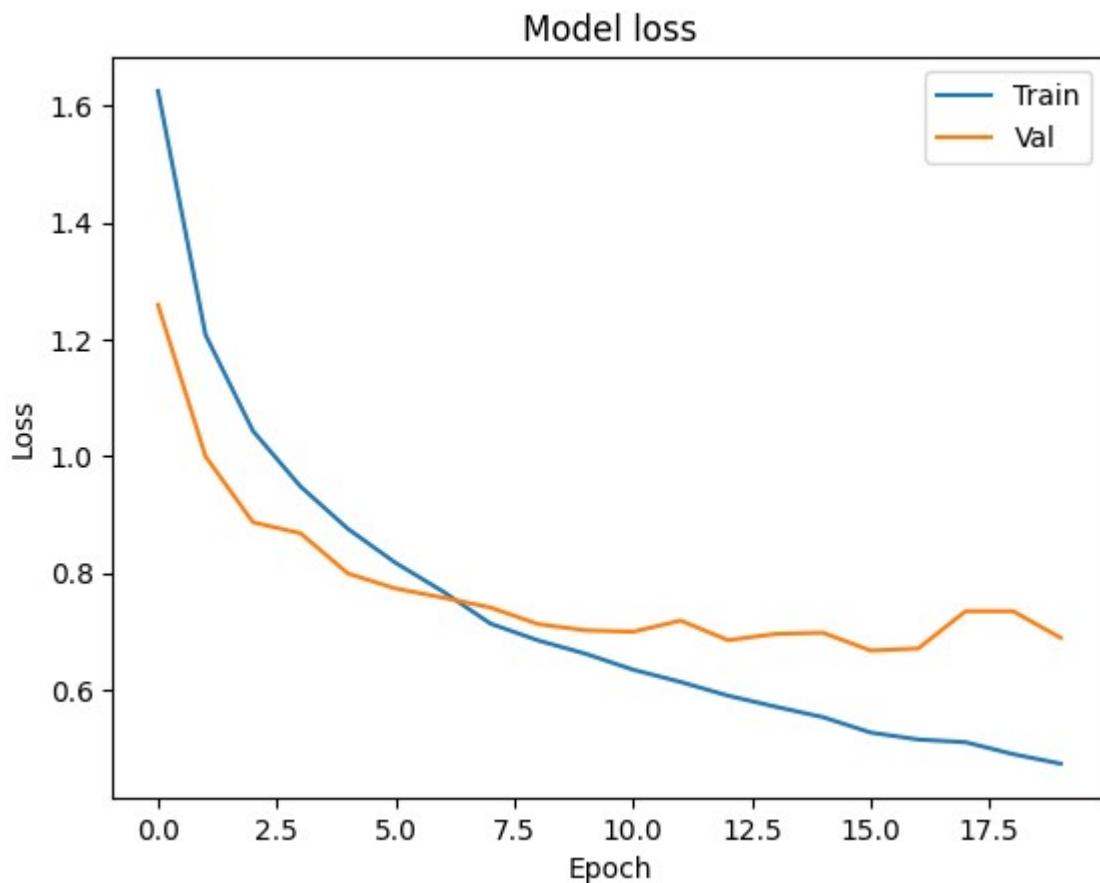
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='lower right')
plt.show()

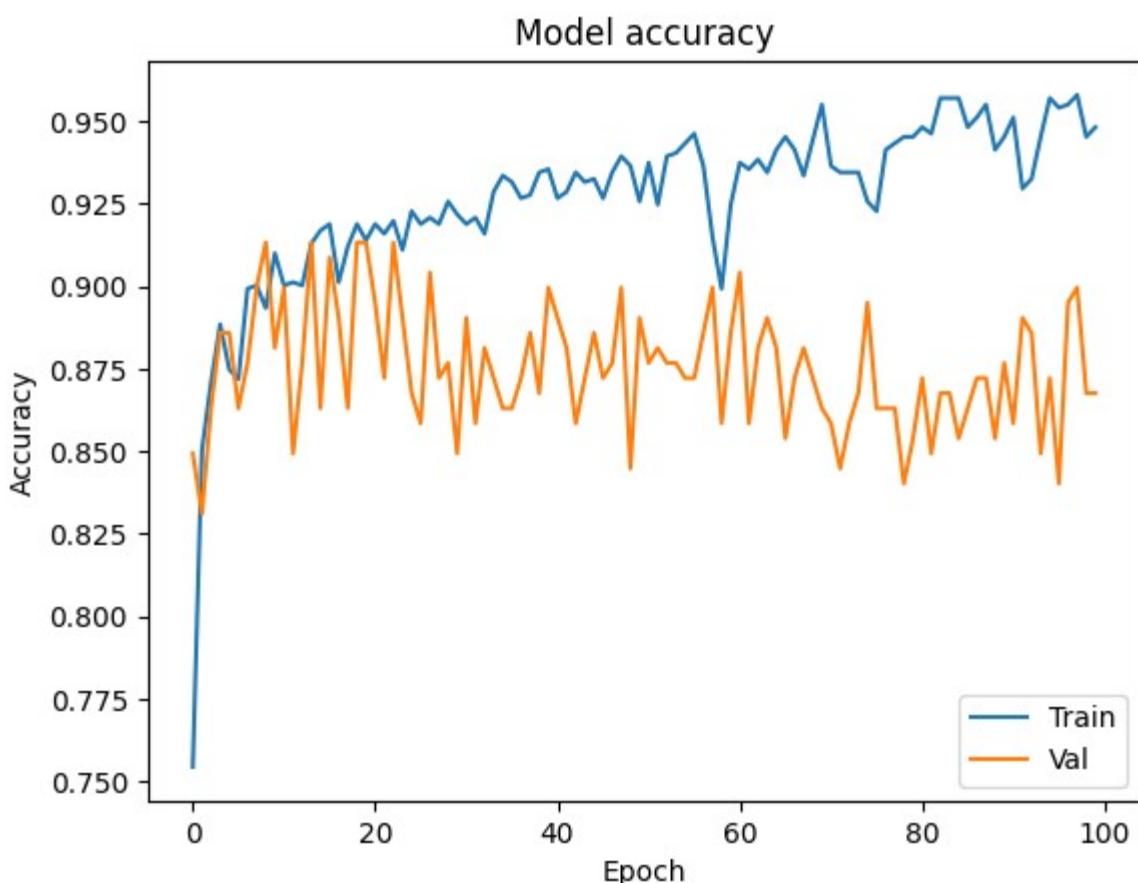
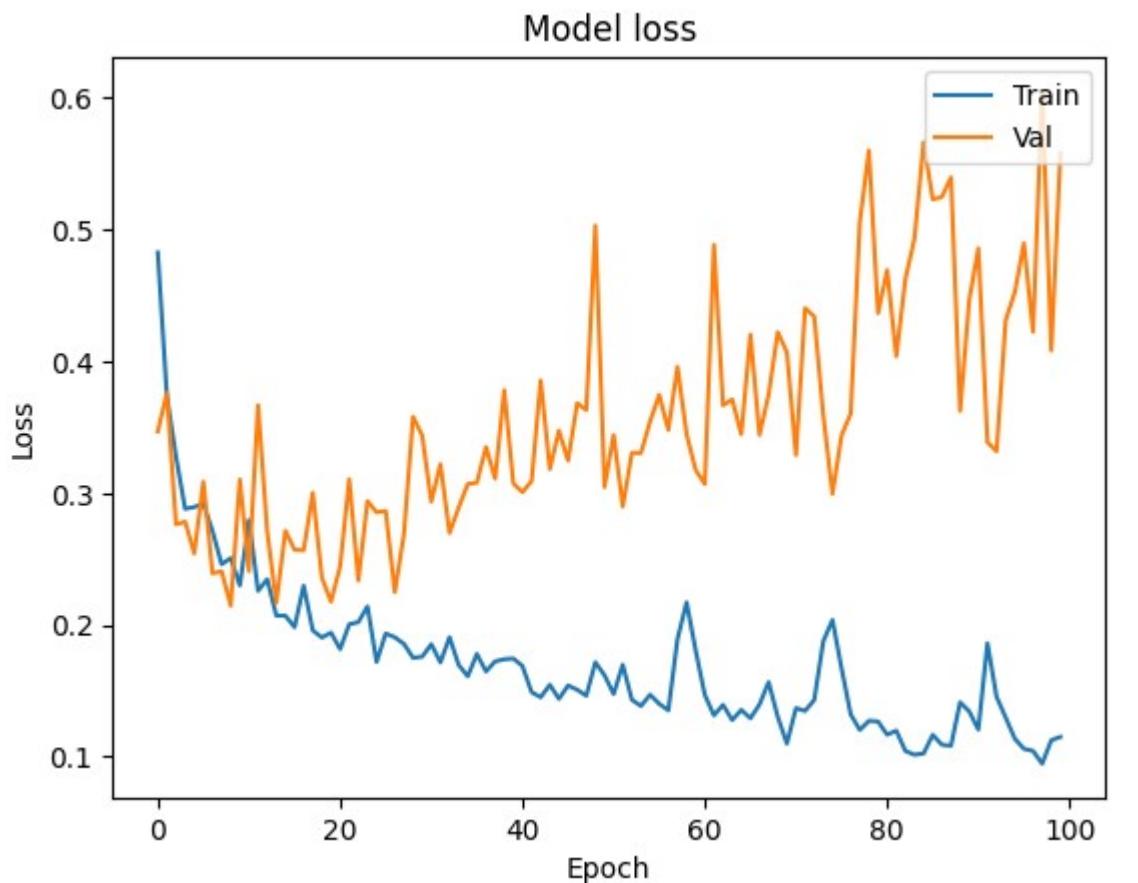
plt.plot(hist_2.history['loss'])
plt.plot(hist_2.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()

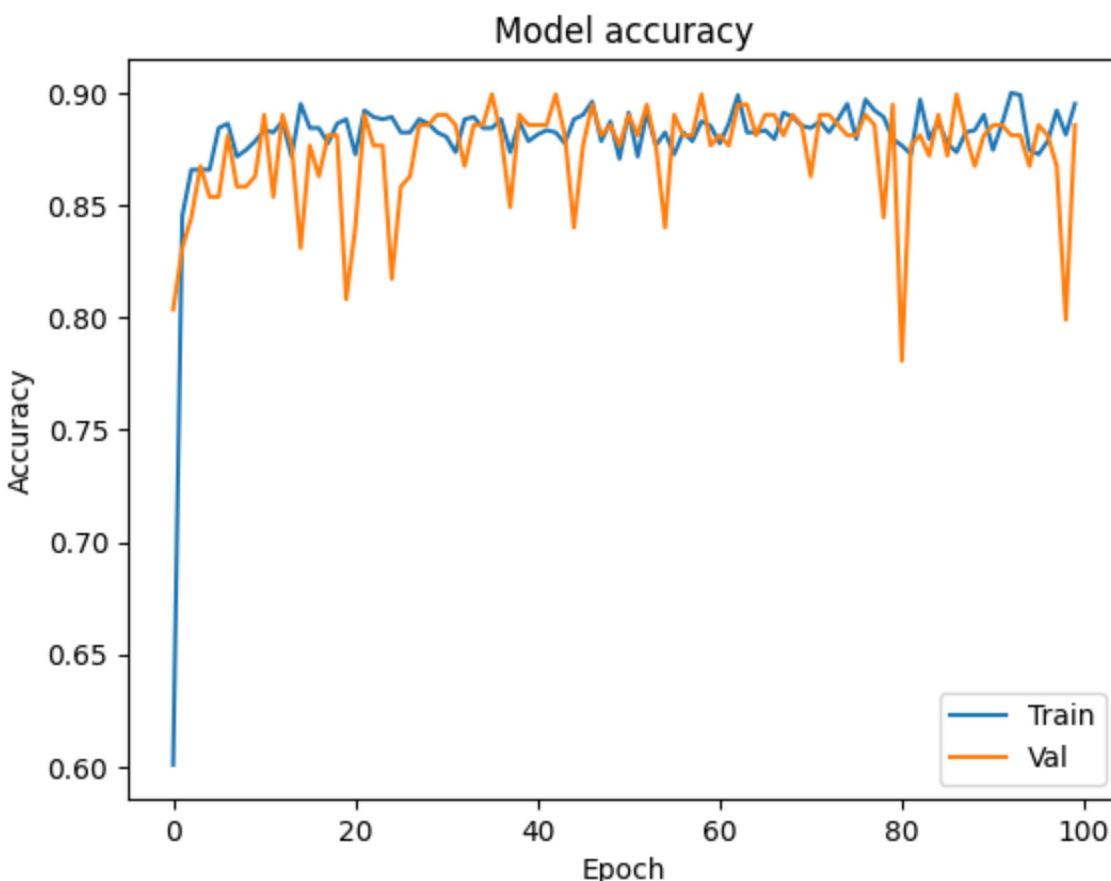
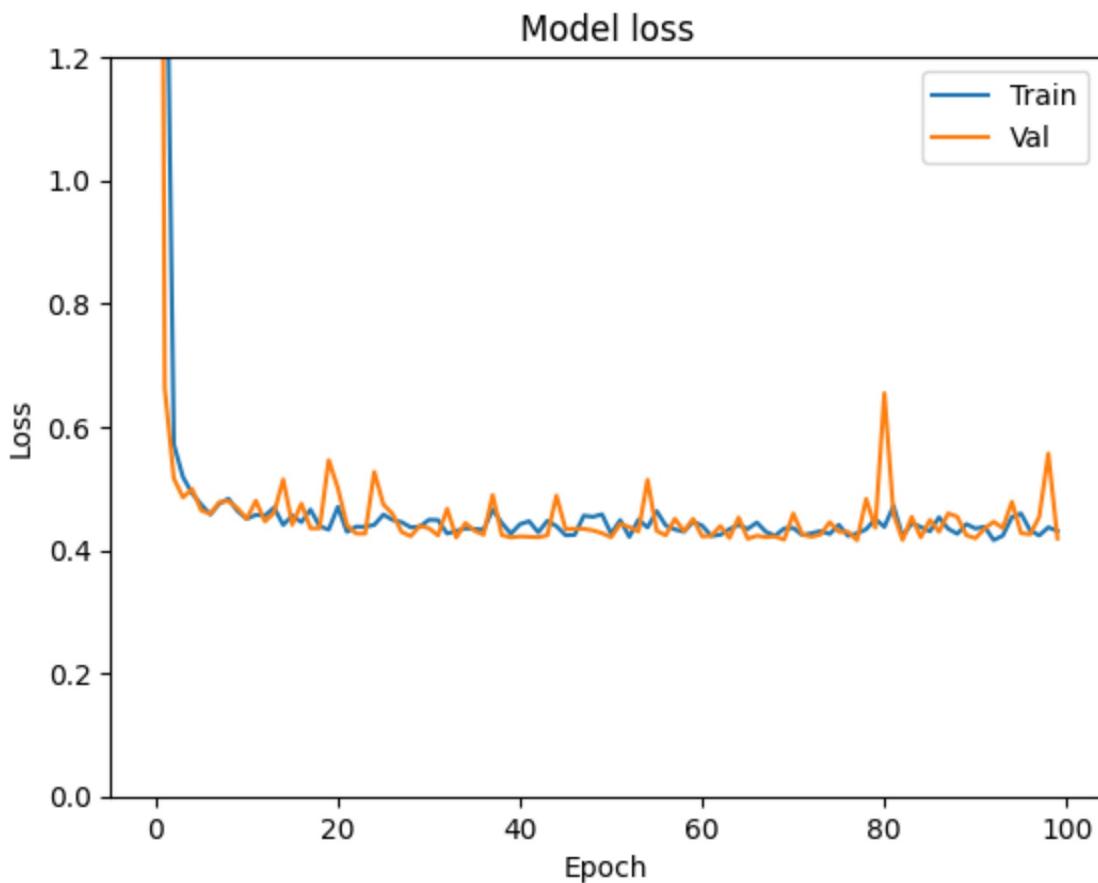
plt.plot(hist_2.history['accuracy'])
plt.plot(hist_2.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='lower right')
plt.show()

plt.plot(hist_3.history['loss'])
plt.plot(hist_3.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.ylim(top=1.2, bottom=0)
plt.show()

plt.plot(hist_3.history['accuracy'])
plt.plot(hist_3.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='lower right')
plt.show()
```







In []:

In []: