Captcha Recognition

Import Libaries

[149]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as img
import tensorflow as tf
import keras as k
import os
import cv2
import pickle
from keras.utils import to_categorical
from PIL import Image
from keras.preprocessing.image import img_to_array, ImageDataGenerator
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Activation, MaxPooling2D, Flatten, Conv2D, Dropout,␣
↪Dense
from keras.callbacks import EarlyStopping
import warnings
warnings.filterwarnings('ignore')
```

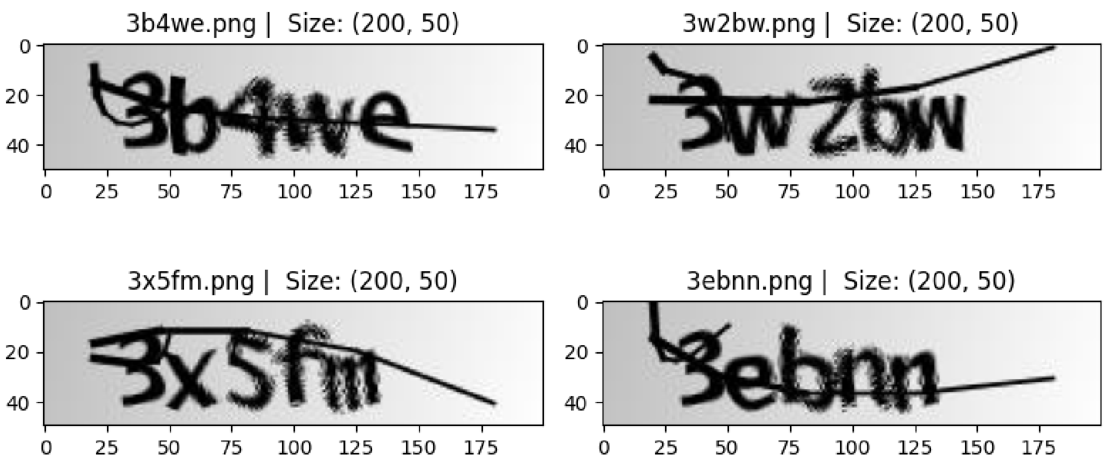
Loading and Preprocessing Data

[150]:
```
# Directory containing the images
directory = '/content/drive/MyDrive/CAPTCHA/samples'
```

[151]:
```
sample = []

for filename in os.listdir(directory):
    image_path = os.path.join(directory, filename)
    img = Image.open(image_path)
    sample.append((filename, img))
```

[152]:
```
sample_images = sample[:4]

fig, axes = plt.subplots(2, 2, figsize=(8, 4))
```

```python
for i in range(len(sample_images)):
    row = i // 2
    col = i % 2
    filename, img = sample_images[i]
    axes[row, col].imshow(img)
    axes[row, col].set_title(f'{filename} |  Size: {img.size}')
    axes[row, col].axis()

plt.tight_layout()
plt.show()
```

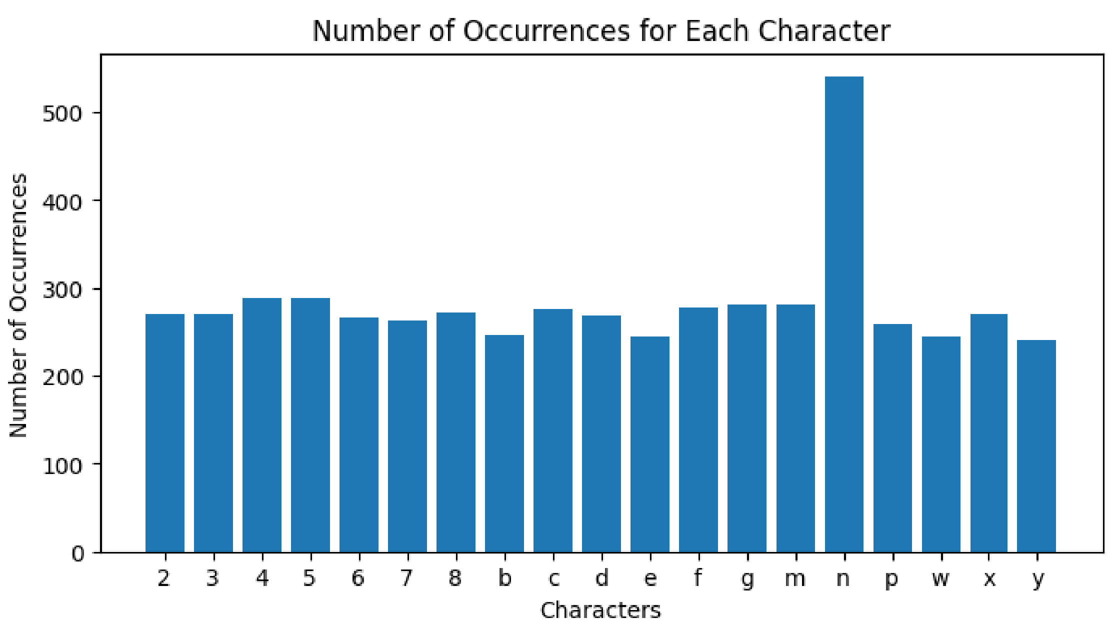


[153]:
```python
unique_characters = set()
max_length = 0
total_samples = 0

for filename in os.listdir(directory):
    # Extract characters from the filename
    characters = os.path.splitext(filename)[0]
    # Update unique characters
    unique_characters.update(set(characters))
    # Update maximum length
    max_length = max(max_length, len(characters))
    # Update total samples
    total_samples += 1

characters_present = sorted(list(unique_characters))

print("Number of unique characters in the whole dataset:", len(unique_characters))
print("Maximum length of any captcha:", max_length)
print("Characters present:", characters_present)
```

```python
print("Total number of samples in the dataset:", total_samples)
```

```
Number of unique characters in the whole dataset: 19
Maximum length of any captcha: 5
Characters present: ['2', '3', '4', '5', '6', '7', '8', 'b', 'c', 'd', 'e', 'f',
'g', 'm', 'n', 'p', 'w', 'x', 'y']
Total number of samples in the dataset: 1070
```

[154]:
```python
character_counts = {}

for filename in os.listdir(directory):
    # Extract characters from the filename
    characters = os.path.splitext(filename)[0]
    # Update character counts
    for char in characters:
        character_counts[char] = character_counts.get(char, 0) + 1

characters_present = sorted(character_counts.keys())
counts = [character_counts[char] for char in characters_present]

plt.figure(figsize=(8, 4))
plt.bar(characters_present, counts)
plt.xlabel('Characters')
plt.ylabel('Number of Occurrences')
plt.title('Number of Occurrences for Each Character')
plt.show()
```



Preprocessing Data

[155]:
```
def load_data(data_dir):
    images = []
    labels = []
    for img_name in os.listdir(data_dir):
        img_path = os.path.join(data_dir, img_name)
        label = os.path.splitext(img_name)[0]

        image = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)  # Read image in␣
↪grayscale
        image = cv2.adaptiveThreshold(image, 255, cv2.
↪ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 145, 0)
        kernel_close = np.ones((5, 5), np.uint8)
        image = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel_close)
        kernel_dilate = np.ones((2, 2), np.uint8)
        image = cv2.dilate(image, kernel_dilate, iterations=1)
        image = cv2.GaussianBlur(image, (5, 5), 0)

        # Split the image into segments
        segments = [image[10:50, 30:50], image[10:50, 50:70],
                    image[10:50, 70:90], image[10:50, 90:110], image[10:50, 110:
↪130]]

        for segment, letter in zip(segments, label):
          images.append(segment)
          labels.append(letter)

    return np.array(images), np.array(labels)
```

[156]:
```
images, labels = load_data(directory)
```

[157]:
```
images=images.astype('float32')
images/=255
```

[158]:
```
labels_le = LabelEncoder().fit_transform(labels)
labels_ohe = OneHotEncoder(sparse = False).fit_transform(labels_le.
↪reshape(len(labels_le),1))
```

[159]:
```
X_train, X_test, y_train, y_test = train_test_split(images, labels_ohe,␣
↪test_size = 0.2, random_state = 42)
```

[160]:
```
row, col = images.shape[1],images.shape[2]
categories = labels_ohe.shape[1]
info = {labels_le[i] : labels[i] for i in range(len(labels))}
```

Training the Model

[161]:
```
model = Sequential()

model.add(Conv2D(filters=16, kernel_size=(3,3), padding='same',␣
 ↪input_shape=(row, col, 1)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(filters=16, kernel_size=(3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dropout(0.4))
model.add(Dense(1500))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(categories))
model.add(Activation("softmax"))

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

[162]:
```
model.summary()
```

```
Model: "sequential_6"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_24 (Conv2D)          (None, 40, 20, 16)        160

 activation_18 (Activation)  (None, 40, 20, 16)        0

 max_pooling2d_24 (MaxPooli  (None, 20, 10, 16)        0
 ng2D)

 conv2d_25 (Conv2D)          (None, 20, 10, 16)        2320

 activation_19 (Activation)  (None, 20, 10, 16)        0
```

```
 max_pooling2d_25 (MaxPooli  (None, 10, 5, 16)         0
 ng2D)

 conv2d_26 (Conv2D)          (None, 10, 5, 32)         4640

 activation_20 (Activation)  (None, 10, 5, 32)         0

 max_pooling2d_26 (MaxPooli  (None, 5, 2, 32)          0
 ng2D)

 conv2d_27 (Conv2D)          (None, 5, 2, 32)          9248

 activation_21 (Activation)  (None, 5, 2, 32)          0

 max_pooling2d_27 (MaxPooli  (None, 2, 1, 32)          0
 ng2D)

 flatten_6 (Flatten)         (None, 64)                0

 dropout_12 (Dropout)        (None, 64)                0

 dense_12 (Dense)            (None, 1500)              97500

 activation_22 (Activation)  (None, 1500)              0

 dropout_13 (Dropout)        (None, 1500)              0

 dense_13 (Dense)            (None, 19)                28519

 activation_23 (Activation)  (None, 19)                0

=================================================================
Total params: 142387 (556.20 KB)
Trainable params: 142387 (556.20 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[163]:

```
batch_size = 150
epochs = 200

history = model.fit(X_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
                    validation_data=(X_test, y_test),
                    shuffle=True)
```

```
Epoch 1/200
```

```
29/29 [==============================] - 6s 150ms/step - loss: 2.9157 -
accuracy: 0.1009 - val_loss: 2.8505 - val_accuracy: 0.0935
Epoch 2/200
29/29 [==============================] - 2s 83ms/step - loss: 2.4981 - accuracy:
0.2243 - val_loss: 1.9184 - val_accuracy: 0.4477
Epoch 3/200
29/29 [==============================] - 2s 77ms/step - loss: 1.7756 - accuracy:
0.4769 - val_loss: 1.3503 - val_accuracy: 0.6308
Epoch 4/200
29/29 [==============================] - 2s 81ms/step - loss: 1.4178 - accuracy:
0.5871 - val_loss: 1.1295 - val_accuracy: 0.6710
Epoch 5/200
29/29 [==============================] - 2s 82ms/step - loss: 1.2439 - accuracy:
0.6357 - val_loss: 0.9850 - val_accuracy: 0.7168
Epoch 6/200
29/29 [==============================] - 4s 131ms/step - loss: 1.1345 -
accuracy: 0.6804 - val_loss: 0.8832 - val_accuracy: 0.7514
Epoch 7/200
29/29 [==============================] - 3s 110ms/step - loss: 1.0300 -
accuracy: 0.7072 - val_loss: 0.8165 - val_accuracy: 0.7523
Epoch 8/200
29/29 [==============================] - 2s 76ms/step - loss: 0.9441 - accuracy:
0.7273 - val_loss: 0.7942 - val_accuracy: 0.7692
Epoch 9/200
29/29 [==============================] - 2s 83ms/step - loss: 0.8823 - accuracy:
0.7449 - val_loss: 0.7092 - val_accuracy: 0.7869
Epoch 10/200
29/29 [==============================] - 2s 76ms/step - loss: 0.8450 - accuracy:
0.7528 - val_loss: 0.6774 - val_accuracy: 0.7944
Epoch 11/200
29/29 [==============================] - 3s 90ms/step - loss: 0.7932 - accuracy:
0.7785 - val_loss: 0.6366 - val_accuracy: 0.8178
Epoch 12/200
29/29 [==============================] - 4s 138ms/step - loss: 0.7485 -
accuracy: 0.7792 - val_loss: 0.6026 - val_accuracy: 0.8224
Epoch 13/200
29/29 [==============================] - 2s 83ms/step - loss: 0.7059 - accuracy:
0.7991 - val_loss: 0.6002 - val_accuracy: 0.8215
Epoch 14/200
29/29 [==============================] - 2s 77ms/step - loss: 0.7090 - accuracy:
0.7867 - val_loss: 0.5964 - val_accuracy: 0.8215
Epoch 15/200
29/29 [==============================] - 2s 79ms/step - loss: 0.6701 - accuracy:
0.8065 - val_loss: 0.5793 - val_accuracy: 0.8355
Epoch 16/200
29/29 [==============================] - 2s 83ms/step - loss: 0.6395 - accuracy:
0.8096 - val_loss: 0.5635 - val_accuracy: 0.8411
Epoch 17/200
```

```
29/29 [==============================] - 3s 109ms/step - loss: 0.6174 -
accuracy: 0.8147 - val_loss: 0.5546 - val_accuracy: 0.8364
Epoch 18/200
29/29 [==============================] - 4s 130ms/step - loss: 0.6110 -
accuracy: 0.8192 - val_loss: 0.5267 - val_accuracy: 0.8495
Epoch 19/200
29/29 [==============================] - 2s 76ms/step - loss: 0.5714 - accuracy:
0.8273 - val_loss: 0.5171 - val_accuracy: 0.8467
Epoch 20/200
29/29 [==============================] - 2s 82ms/step - loss: 0.5685 - accuracy:
0.8315 - val_loss: 0.5217 - val_accuracy: 0.8477
Epoch 21/200
29/29 [==============================] - 2s 77ms/step - loss: 0.5585 - accuracy:
0.8280 - val_loss: 0.5123 - val_accuracy: 0.8486
Epoch 22/200
29/29 [==============================] - 2s 83ms/step - loss: 0.5330 - accuracy:
0.8350 - val_loss: 0.5236 - val_accuracy: 0.8533
Epoch 23/200
29/29 [==============================] - 4s 125ms/step - loss: 0.5363 -
accuracy: 0.8357 - val_loss: 0.5094 - val_accuracy: 0.8570
Epoch 24/200
29/29 [==============================] - 3s 112ms/step - loss: 0.5052 -
accuracy: 0.8435 - val_loss: 0.4971 - val_accuracy: 0.8626
Epoch 25/200
29/29 [==============================] - 2s 81ms/step - loss: 0.4823 - accuracy:
0.8500 - val_loss: 0.4710 - val_accuracy: 0.8626
Epoch 26/200
29/29 [==============================] - 2s 76ms/step - loss: 0.4716 - accuracy:
0.8521 - val_loss: 0.4810 - val_accuracy: 0.8607
Epoch 27/200
29/29 [==============================] - 2s 78ms/step - loss: 0.4806 - accuracy:
0.8470 - val_loss: 0.4812 - val_accuracy: 0.8710
Epoch 28/200
29/29 [==============================] - 2s 77ms/step - loss: 0.4813 - accuracy:
0.8491 - val_loss: 0.5058 - val_accuracy: 0.8617
Epoch 29/200
29/29 [==============================] - 4s 135ms/step - loss: 0.4719 -
accuracy: 0.8509 - val_loss: 0.4823 - val_accuracy: 0.8664
Epoch 30/200
29/29 [==============================] - 3s 97ms/step - loss: 0.4459 - accuracy:
0.8565 - val_loss: 0.4743 - val_accuracy: 0.8617
Epoch 31/200
29/29 [==============================] - 2s 77ms/step - loss: 0.4496 - accuracy:
0.8556 - val_loss: 0.4952 - val_accuracy: 0.8692
Epoch 32/200
29/29 [==============================] - 2s 77ms/step - loss: 0.4420 - accuracy:
0.8572 - val_loss: 0.4874 - val_accuracy: 0.8673
Epoch 33/200
```

```
29/29 [==============================] - 2s 77ms/step - loss: 0.4241 - accuracy:
0.8626 - val_loss: 0.4861 - val_accuracy: 0.8636
Epoch 34/200
29/29 [==============================] - 3s 99ms/step - loss: 0.4149 - accuracy:
0.8664 - val_loss: 0.5280 - val_accuracy: 0.8673
Epoch 35/200
29/29 [==============================] - 4s 139ms/step - loss: 0.4317 -
accuracy: 0.8551 - val_loss: 0.4878 - val_accuracy: 0.8589
Epoch 36/200
29/29 [==============================] - 2s 78ms/step - loss: 0.4007 - accuracy:
0.8706 - val_loss: 0.4760 - val_accuracy: 0.8701
Epoch 37/200
29/29 [==============================] - 2s 82ms/step - loss: 0.4037 - accuracy:
0.8664 - val_loss: 0.4793 - val_accuracy: 0.8673
Epoch 38/200
29/29 [==============================] - 2s 79ms/step - loss: 0.3899 - accuracy:
0.8717 - val_loss: 0.4723 - val_accuracy: 0.8720
Epoch 39/200
29/29 [==============================] - 2s 83ms/step - loss: 0.3950 - accuracy:
0.8701 - val_loss: 0.4980 - val_accuracy: 0.8673
Epoch 40/200
29/29 [==============================] - 3s 120ms/step - loss: 0.3959 -
accuracy: 0.8701 - val_loss: 0.4768 - val_accuracy: 0.8682
Epoch 41/200
29/29 [==============================] - 3s 112ms/step - loss: 0.3881 -
accuracy: 0.8745 - val_loss: 0.4916 - val_accuracy: 0.8682
Epoch 42/200
29/29 [==============================] - 2s 82ms/step - loss: 0.3789 - accuracy:
0.8762 - val_loss: 0.5065 - val_accuracy: 0.8664
Epoch 43/200
29/29 [==============================] - 2s 84ms/step - loss: 0.3683 - accuracy:
0.8780 - val_loss: 0.5140 - val_accuracy: 0.8673
Epoch 44/200
29/29 [==============================] - 2s 78ms/step - loss: 0.3535 - accuracy:
0.8853 - val_loss: 0.4890 - val_accuracy: 0.8729
Epoch 45/200
29/29 [==============================] - 2s 84ms/step - loss: 0.3468 - accuracy:
0.8804 - val_loss: 0.4750 - val_accuracy: 0.8692
Epoch 46/200
29/29 [==============================] - 4s 140ms/step - loss: 0.3347 -
accuracy: 0.8813 - val_loss: 0.4930 - val_accuracy: 0.8701
Epoch 47/200
29/29 [==============================] - 3s 85ms/step - loss: 0.3625 - accuracy:
0.8769 - val_loss: 0.4972 - val_accuracy: 0.8645
Epoch 48/200
29/29 [==============================] - 2s 77ms/step - loss: 0.3511 - accuracy:
0.8778 - val_loss: 0.5039 - val_accuracy: 0.8738
Epoch 49/200
```

```
29/29 [==============================] - 2s 77ms/step - loss: 0.3502 - accuracy:
0.8797 - val_loss: 0.4672 - val_accuracy: 0.8776
Epoch 50/200
29/29 [==============================] - 2s 78ms/step - loss: 0.3402 - accuracy:
0.8839 - val_loss: 0.5086 - val_accuracy: 0.8692
Epoch 51/200
29/29 [==============================] - 3s 100ms/step - loss: 0.3368 -
accuracy: 0.8815 - val_loss: 0.5172 - val_accuracy: 0.8682
Epoch 52/200
29/29 [==============================] - 4s 142ms/step - loss: 0.3223 -
accuracy: 0.8893 - val_loss: 0.4881 - val_accuracy: 0.8748
Epoch 53/200
29/29 [==============================] - 2s 76ms/step - loss: 0.3125 - accuracy:
0.8916 - val_loss: 0.4994 - val_accuracy: 0.8701
Epoch 54/200
29/29 [==============================] - 2s 79ms/step - loss: 0.3155 - accuracy:
0.8928 - val_loss: 0.4950 - val_accuracy: 0.8692
Epoch 55/200
29/29 [==============================] - 2s 82ms/step - loss: 0.2991 - accuracy:
0.8986 - val_loss: 0.5201 - val_accuracy: 0.8682
Epoch 56/200
29/29 [==============================] - 2s 79ms/step - loss: 0.3081 - accuracy:
0.8963 - val_loss: 0.5063 - val_accuracy: 0.8729
Epoch 57/200
29/29 [==============================] - 3s 122ms/step - loss: 0.3007 -
accuracy: 0.8951 - val_loss: 0.4988 - val_accuracy: 0.8748
Epoch 58/200
29/29 [==============================] - 3s 117ms/step - loss: 0.3108 -
accuracy: 0.8916 - val_loss: 0.5278 - val_accuracy: 0.8748
Epoch 59/200
29/29 [==============================] - 2s 79ms/step - loss: 0.3048 - accuracy:
0.8930 - val_loss: 0.5010 - val_accuracy: 0.8757
Epoch 60/200
29/29 [==============================] - 2s 84ms/step - loss: 0.3000 - accuracy:
0.8923 - val_loss: 0.5176 - val_accuracy: 0.8785
Epoch 61/200
29/29 [==============================] - 2s 84ms/step - loss: 0.2987 - accuracy:
0.8958 - val_loss: 0.5127 - val_accuracy: 0.8794
Epoch 62/200
29/29 [==============================] - 2s 85ms/step - loss: 0.3025 - accuracy:
0.8942 - val_loss: 0.5046 - val_accuracy: 0.8776
Epoch 63/200
29/29 [==============================] - 4s 137ms/step - loss: 0.2935 -
accuracy: 0.8958 - val_loss: 0.5473 - val_accuracy: 0.8692
Epoch 64/200
29/29 [==============================] - 3s 92ms/step - loss: 0.3031 - accuracy:
0.8986 - val_loss: 0.5168 - val_accuracy: 0.8701
Epoch 65/200
```

```
29/29 [==============================] - 2s 79ms/step - loss: 0.3088 - accuracy:
0.8911 - val_loss: 0.5113 - val_accuracy: 0.8748
Epoch 66/200
29/29 [==============================] - 2s 76ms/step - loss: 0.2809 - accuracy:
0.9014 - val_loss: 0.5114 - val_accuracy: 0.8766
Epoch 67/200
29/29 [==============================] - 2s 85ms/step - loss: 0.2766 - accuracy:
0.9033 - val_loss: 0.5454 - val_accuracy: 0.8720
Epoch 68/200
29/29 [==============================] - 3s 109ms/step - loss: 0.2850 -
accuracy: 0.9007 - val_loss: 0.4976 - val_accuracy: 0.8794
Epoch 69/200
29/29 [==============================] - 4s 132ms/step - loss: 0.2683 -
accuracy: 0.9035 - val_loss: 0.5081 - val_accuracy: 0.8794
Epoch 70/200
29/29 [==============================] - 2s 82ms/step - loss: 0.2695 - accuracy:
0.9096 - val_loss: 0.5190 - val_accuracy: 0.8710
Epoch 71/200
29/29 [==============================] - 2s 84ms/step - loss: 0.2795 - accuracy:
0.9033 - val_loss: 0.5214 - val_accuracy: 0.8710
Epoch 72/200
29/29 [==============================] - 2s 87ms/step - loss: 0.2629 - accuracy:
0.9079 - val_loss: 0.5050 - val_accuracy: 0.8729
Epoch 73/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2647 - accuracy:
0.9091 - val_loss: 0.5151 - val_accuracy: 0.8860
Epoch 74/200
29/29 [==============================] - 4s 134ms/step - loss: 0.2699 -
accuracy: 0.9035 - val_loss: 0.5510 - val_accuracy: 0.8776
Epoch 75/200
29/29 [==============================] - 3s 92ms/step - loss: 0.2772 - accuracy:
0.9068 - val_loss: 0.5307 - val_accuracy: 0.8766
Epoch 76/200
29/29 [==============================] - 2s 78ms/step - loss: 0.2712 - accuracy:
0.9051 - val_loss: 0.5259 - val_accuracy: 0.8757
Epoch 77/200
29/29 [==============================] - 2s 84ms/step - loss: 0.2456 - accuracy:
0.9105 - val_loss: 0.5387 - val_accuracy: 0.8794
Epoch 78/200
29/29 [==============================] - 2s 77ms/step - loss: 0.2508 - accuracy:
0.9084 - val_loss: 0.5531 - val_accuracy: 0.8785
Epoch 79/200
29/29 [==============================] - 3s 100ms/step - loss: 0.2378 -
accuracy: 0.9164 - val_loss: 0.5456 - val_accuracy: 0.8776
Epoch 80/200
29/29 [==============================] - 4s 141ms/step - loss: 0.2475 -
accuracy: 0.9112 - val_loss: 0.5471 - val_accuracy: 0.8813
Epoch 81/200
```

```
29/29 [==============================] - 2s 83ms/step - loss: 0.2747 - accuracy:
0.9009 - val_loss: 0.5597 - val_accuracy: 0.8813
Epoch 82/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2484 - accuracy:
0.9121 - val_loss: 0.5664 - val_accuracy: 0.8813
Epoch 83/200
29/29 [==============================] - 2s 82ms/step - loss: 0.2480 - accuracy:
0.9140 - val_loss: 0.5723 - val_accuracy: 0.8776
Epoch 84/200
29/29 [==============================] - 2s 79ms/step - loss: 0.2391 - accuracy:
0.9171 - val_loss: 0.5603 - val_accuracy: 0.8804
Epoch 85/200
29/29 [==============================] - 4s 127ms/step - loss: 0.2464 -
accuracy: 0.9119 - val_loss: 0.5511 - val_accuracy: 0.8804
Epoch 86/200
29/29 [==============================] - 3s 110ms/step - loss: 0.2447 -
accuracy: 0.9161 - val_loss: 0.5771 - val_accuracy: 0.8794
Epoch 87/200
29/29 [==============================] - 2s 82ms/step - loss: 0.2378 - accuracy:
0.9126 - val_loss: 0.5743 - val_accuracy: 0.8804
Epoch 88/200
29/29 [==============================] - 2s 77ms/step - loss: 0.2478 - accuracy:
0.9114 - val_loss: 0.5412 - val_accuracy: 0.8804
Epoch 89/200
29/29 [==============================] - 2s 85ms/step - loss: 0.2515 - accuracy:
0.9129 - val_loss: 0.5584 - val_accuracy: 0.8710
Epoch 90/200
29/29 [==============================] - 2s 86ms/step - loss: 0.2381 - accuracy:
0.9143 - val_loss: 0.5375 - val_accuracy: 0.8804
Epoch 91/200
29/29 [==============================] - 4s 137ms/step - loss: 0.2441 -
accuracy: 0.9114 - val_loss: 0.5590 - val_accuracy: 0.8785
Epoch 92/200
29/29 [==============================] - 3s 89ms/step - loss: 0.2297 - accuracy:
0.9140 - val_loss: 0.5852 - val_accuracy: 0.8757
Epoch 93/200
29/29 [==============================] - 2s 77ms/step - loss: 0.2262 - accuracy:
0.9185 - val_loss: 0.5775 - val_accuracy: 0.8804
Epoch 94/200
29/29 [==============================] - 2s 78ms/step - loss: 0.2292 - accuracy:
0.9145 - val_loss: 0.5640 - val_accuracy: 0.8776
Epoch 95/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2371 - accuracy:
0.9124 - val_loss: 0.5600 - val_accuracy: 0.8776
Epoch 96/200
29/29 [==============================] - 3s 112ms/step - loss: 0.2434 -
accuracy: 0.9112 - val_loss: 0.5483 - val_accuracy: 0.8785
Epoch 97/200
```

```
29/29 [==============================] - 4s 128ms/step - loss: 0.2261 -
accuracy: 0.9187 - val_loss: 0.6144 - val_accuracy: 0.8804
Epoch 98/200
29/29 [==============================] - 2s 82ms/step - loss: 0.2271 - accuracy:
0.9187 - val_loss: 0.5562 - val_accuracy: 0.8804
Epoch 99/200
29/29 [==============================] - 2s 78ms/step - loss: 0.2019 - accuracy:
0.9276 - val_loss: 0.5885 - val_accuracy: 0.8850
Epoch 100/200
29/29 [==============================] - 2s 85ms/step - loss: 0.2164 - accuracy:
0.9220 - val_loss: 0.5881 - val_accuracy: 0.8766
Epoch 101/200
29/29 [==============================] - 2s 80ms/step - loss: 0.2113 - accuracy:
0.9264 - val_loss: 0.5932 - val_accuracy: 0.8748
Epoch 102/200
29/29 [==============================] - 4s 138ms/step - loss: 0.2037 -
accuracy: 0.9255 - val_loss: 0.6009 - val_accuracy: 0.8841
Epoch 103/200
29/29 [==============================] - 3s 104ms/step - loss: 0.2147 -
accuracy: 0.9222 - val_loss: 0.5962 - val_accuracy: 0.8813
Epoch 104/200
29/29 [==============================] - 2s 78ms/step - loss: 0.2091 - accuracy:
0.9245 - val_loss: 0.6055 - val_accuracy: 0.8813
Epoch 105/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2098 - accuracy:
0.9229 - val_loss: 0.5675 - val_accuracy: 0.8794
Epoch 106/200
29/29 [==============================] - 2s 78ms/step - loss: 0.2352 - accuracy:
0.9159 - val_loss: 0.6186 - val_accuracy: 0.8804
Epoch 107/200
29/29 [==============================] - 3s 92ms/step - loss: 0.2279 - accuracy:
0.9187 - val_loss: 0.5702 - val_accuracy: 0.8804
Epoch 108/200
29/29 [==============================] - 4s 140ms/step - loss: 0.2247 -
accuracy: 0.9175 - val_loss: 0.5752 - val_accuracy: 0.8785
Epoch 109/200
29/29 [==============================] - 2s 81ms/step - loss: 0.2264 - accuracy:
0.9136 - val_loss: 0.6045 - val_accuracy: 0.8822
Epoch 110/200
29/29 [==============================] - 2s 77ms/step - loss: 0.2044 - accuracy:
0.9243 - val_loss: 0.6087 - val_accuracy: 0.8738
Epoch 111/200
29/29 [==============================] - 2s 80ms/step - loss: 0.1987 - accuracy:
0.9255 - val_loss: 0.5911 - val_accuracy: 0.8766
Epoch 112/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1966 - accuracy:
0.9292 - val_loss: 0.6235 - val_accuracy: 0.8850
Epoch 113/200
```

```
29/29 [==============================] - 3s 109ms/step - loss: 0.2004 -
accuracy: 0.9294 - val_loss: 0.6485 - val_accuracy: 0.8832
Epoch 114/200
29/29 [==============================] - 4s 124ms/step - loss: 0.1983 -
accuracy: 0.9264 - val_loss: 0.6055 - val_accuracy: 0.8822
Epoch 115/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2034 - accuracy:
0.9259 - val_loss: 0.5973 - val_accuracy: 0.8813
Epoch 116/200
29/29 [==============================] - 3s 87ms/step - loss: 0.2077 - accuracy:
0.9231 - val_loss: 0.6366 - val_accuracy: 0.8785
Epoch 117/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1990 - accuracy:
0.9280 - val_loss: 0.5812 - val_accuracy: 0.8813
Epoch 118/200
29/29 [==============================] - 2s 77ms/step - loss: 0.2069 - accuracy:
0.9215 - val_loss: 0.6163 - val_accuracy: 0.8785
Epoch 119/200
29/29 [==============================] - 4s 131ms/step - loss: 0.2124 -
accuracy: 0.9217 - val_loss: 0.5937 - val_accuracy: 0.8804
Epoch 120/200
29/29 [==============================] - 3s 100ms/step - loss: 0.2057 -
accuracy: 0.9262 - val_loss: 0.6263 - val_accuracy: 0.8766
Epoch 121/200
29/29 [==============================] - 2s 84ms/step - loss: 0.1949 - accuracy:
0.9290 - val_loss: 0.5900 - val_accuracy: 0.8860
Epoch 122/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1927 - accuracy:
0.9285 - val_loss: 0.5645 - val_accuracy: 0.8850
Epoch 123/200
29/29 [==============================] - 2s 83ms/step - loss: 0.2020 - accuracy:
0.9271 - val_loss: 0.6094 - val_accuracy: 0.8822
Epoch 124/200
29/29 [==============================] - 3s 104ms/step - loss: 0.2017 -
accuracy: 0.9280 - val_loss: 0.6184 - val_accuracy: 0.8776
Epoch 125/200
29/29 [==============================] - 4s 139ms/step - loss: 0.1894 -
accuracy: 0.9332 - val_loss: 0.5911 - val_accuracy: 0.8879
Epoch 126/200
29/29 [==============================] - 2s 83ms/step - loss: 0.1952 - accuracy:
0.9315 - val_loss: 0.5970 - val_accuracy: 0.8757
Epoch 127/200
29/29 [==============================] - 2s 85ms/step - loss: 0.2119 - accuracy:
0.9229 - val_loss: 0.5665 - val_accuracy: 0.8804
Epoch 128/200
29/29 [==============================] - 2s 79ms/step - loss: 0.2048 - accuracy:
0.9231 - val_loss: 0.6208 - val_accuracy: 0.8813
Epoch 129/200
```

```
29/29 [==============================] - 2s 77ms/step - loss: 0.1950 - accuracy:
0.9278 - val_loss: 0.5850 - val_accuracy: 0.8822
Epoch 130/200
29/29 [==============================] - 4s 136ms/step - loss: 0.1919 -
accuracy: 0.9290 - val_loss: 0.5845 - val_accuracy: 0.8841
Epoch 131/200
29/29 [==============================] - 3s 106ms/step - loss: 0.1998 -
accuracy: 0.9269 - val_loss: 0.6086 - val_accuracy: 0.8879
Epoch 132/200
29/29 [==============================] - 2s 83ms/step - loss: 0.1832 - accuracy:
0.9364 - val_loss: 0.5795 - val_accuracy: 0.8822
Epoch 133/200
29/29 [==============================] - 2s 83ms/step - loss: 0.1948 - accuracy:
0.9273 - val_loss: 0.5850 - val_accuracy: 0.8766
Epoch 134/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1851 - accuracy:
0.9332 - val_loss: 0.6112 - val_accuracy: 0.8860
Epoch 135/200
29/29 [==============================] - 3s 96ms/step - loss: 0.1766 - accuracy:
0.9346 - val_loss: 0.6588 - val_accuracy: 0.8813
Epoch 136/200
29/29 [==============================] - 4s 137ms/step - loss: 0.1797 -
accuracy: 0.9334 - val_loss: 0.5940 - val_accuracy: 0.8766
Epoch 137/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1960 - accuracy:
0.9304 - val_loss: 0.6072 - val_accuracy: 0.8841
Epoch 138/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1902 - accuracy:
0.9322 - val_loss: 0.6227 - val_accuracy: 0.8785
Epoch 139/200
29/29 [==============================] - 2s 84ms/step - loss: 0.1811 - accuracy:
0.9357 - val_loss: 0.6252 - val_accuracy: 0.8832
Epoch 140/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1756 - accuracy:
0.9325 - val_loss: 0.6492 - val_accuracy: 0.8813
Epoch 141/200
29/29 [==============================] - 3s 115ms/step - loss: 0.1708 -
accuracy: 0.9339 - val_loss: 0.6425 - val_accuracy: 0.8822
Epoch 142/200
29/29 [==============================] - 3s 116ms/step - loss: 0.1679 -
accuracy: 0.9393 - val_loss: 0.6235 - val_accuracy: 0.8860
Epoch 143/200
29/29 [==============================] - 2s 77ms/step - loss: 0.1751 - accuracy:
0.9341 - val_loss: 0.6972 - val_accuracy: 0.8785
Epoch 144/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1836 - accuracy:
0.9325 - val_loss: 0.6603 - val_accuracy: 0.8832
Epoch 145/200
```

```
29/29 [==============================] - 2s 77ms/step - loss: 0.1977 - accuracy:
0.9287 - val_loss: 0.6428 - val_accuracy: 0.8841
Epoch 146/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1840 - accuracy:
0.9311 - val_loss: 0.6151 - val_accuracy: 0.8841
Epoch 147/200
29/29 [==============================] - 4s 141ms/step - loss: 0.1805 -
accuracy: 0.9327 - val_loss: 0.6540 - val_accuracy: 0.8794
Epoch 148/200
29/29 [==============================] - 3s 108ms/step - loss: 0.1755 -
accuracy: 0.9362 - val_loss: 0.6245 - val_accuracy: 0.8850
Epoch 149/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1841 - accuracy:
0.9308 - val_loss: 0.6744 - val_accuracy: 0.8841
Epoch 150/200
29/29 [==============================] - 2s 80ms/step - loss: 0.1750 - accuracy:
0.9357 - val_loss: 0.6542 - val_accuracy: 0.8738
Epoch 151/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1690 - accuracy:
0.9397 - val_loss: 0.6439 - val_accuracy: 0.8841
Epoch 152/200
29/29 [==============================] - 3s 99ms/step - loss: 0.1641 - accuracy:
0.9397 - val_loss: 0.6341 - val_accuracy: 0.8860
Epoch 153/200
29/29 [==============================] - 4s 143ms/step - loss: 0.1582 -
accuracy: 0.9432 - val_loss: 0.6370 - val_accuracy: 0.8832
Epoch 154/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1724 - accuracy:
0.9346 - val_loss: 0.6453 - val_accuracy: 0.8832
Epoch 155/200
29/29 [==============================] - 2s 77ms/step - loss: 0.1839 - accuracy:
0.9322 - val_loss: 0.6625 - val_accuracy: 0.8897
Epoch 156/200
29/29 [==============================] - 3s 87ms/step - loss: 0.1677 - accuracy:
0.9423 - val_loss: 0.6181 - val_accuracy: 0.8879
Epoch 157/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1571 - accuracy:
0.9397 - val_loss: 0.6251 - val_accuracy: 0.8888
Epoch 158/200
29/29 [==============================] - 4s 130ms/step - loss: 0.1836 -
accuracy: 0.9334 - val_loss: 0.6511 - val_accuracy: 0.8850
Epoch 159/200
29/29 [==============================] - 3s 113ms/step - loss: 0.1747 -
accuracy: 0.9376 - val_loss: 0.6085 - val_accuracy: 0.8776
Epoch 160/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1653 - accuracy:
0.9425 - val_loss: 0.6405 - val_accuracy: 0.8850
Epoch 161/200
```

```
29/29 [==============================] - 2s 80ms/step - loss: 0.1778 - accuracy:
0.9329 - val_loss: 0.6659 - val_accuracy: 0.8813
Epoch 162/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1761 - accuracy:
0.9325 - val_loss: 0.6526 - val_accuracy: 0.8841
Epoch 163/200
29/29 [==============================] - 2s 82ms/step - loss: 0.1674 - accuracy:
0.9400 - val_loss: 0.6723 - val_accuracy: 0.8832
Epoch 164/200
29/29 [==============================] - 4s 134ms/step - loss: 0.1738 -
accuracy: 0.9374 - val_loss: 0.6085 - val_accuracy: 0.8860
Epoch 165/200
29/29 [==============================] - 3s 100ms/step - loss: 0.1412 -
accuracy: 0.9477 - val_loss: 0.6507 - val_accuracy: 0.8869
Epoch 166/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1696 - accuracy:
0.9381 - val_loss: 0.7099 - val_accuracy: 0.8794
Epoch 167/200
29/29 [==============================] - 2s 83ms/step - loss: 0.1703 - accuracy:
0.9381 - val_loss: 0.6719 - val_accuracy: 0.8841
Epoch 168/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1642 - accuracy:
0.9379 - val_loss: 0.6712 - val_accuracy: 0.8785
Epoch 169/200
29/29 [==============================] - 3s 102ms/step - loss: 0.1541 -
accuracy: 0.9449 - val_loss: 0.6363 - val_accuracy: 0.8729
Epoch 170/200
29/29 [==============================] - 4s 140ms/step - loss: 0.1691 -
accuracy: 0.9400 - val_loss: 0.6576 - val_accuracy: 0.8776
Epoch 171/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1530 - accuracy:
0.9416 - val_loss: 0.6653 - val_accuracy: 0.8776
Epoch 172/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1582 - accuracy:
0.9456 - val_loss: 0.7024 - val_accuracy: 0.8720
Epoch 173/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1594 - accuracy:
0.9404 - val_loss: 0.6891 - val_accuracy: 0.8757
Epoch 174/200
29/29 [==============================] - 2s 80ms/step - loss: 0.1802 - accuracy:
0.9350 - val_loss: 0.6257 - val_accuracy: 0.8757
Epoch 175/200
29/29 [==============================] - 3s 117ms/step - loss: 0.1590 -
accuracy: 0.9411 - val_loss: 0.6326 - val_accuracy: 0.8785
Epoch 176/200
29/29 [==============================] - 3s 120ms/step - loss: 0.1699 -
accuracy: 0.9379 - val_loss: 0.6652 - val_accuracy: 0.8869
Epoch 177/200
```

```
29/29 [==============================] - 2s 77ms/step - loss: 0.1633 - accuracy:
0.9402 - val_loss: 0.6451 - val_accuracy: 0.8888
Epoch 178/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1658 - accuracy:
0.9418 - val_loss: 0.6547 - val_accuracy: 0.8785
Epoch 179/200
29/29 [==============================] - 2s 80ms/step - loss: 0.1623 - accuracy:
0.9409 - val_loss: 0.6818 - val_accuracy: 0.8850
Epoch 180/200
29/29 [==============================] - 2s 86ms/step - loss: 0.1408 - accuracy:
0.9442 - val_loss: 0.6535 - val_accuracy: 0.8860
Epoch 181/200
29/29 [==============================] - 4s 142ms/step - loss: 0.1497 -
accuracy: 0.9449 - val_loss: 0.6990 - val_accuracy: 0.8832
Epoch 182/200
29/29 [==============================] - 3s 100ms/step - loss: 0.1514 -
accuracy: 0.9423 - val_loss: 0.6823 - val_accuracy: 0.8832
Epoch 183/200
29/29 [==============================] - 2s 86ms/step - loss: 0.1748 - accuracy:
0.9341 - val_loss: 0.6894 - val_accuracy: 0.8832
Epoch 184/200
29/29 [==============================] - 2s 79ms/step - loss: 0.1426 - accuracy:
0.9465 - val_loss: 0.7009 - val_accuracy: 0.8813
Epoch 185/200
29/29 [==============================] - 2s 83ms/step - loss: 0.1561 - accuracy:
0.9411 - val_loss: 0.7164 - val_accuracy: 0.8766
Epoch 186/200
29/29 [==============================] - 3s 104ms/step - loss: 0.1589 -
accuracy: 0.9411 - val_loss: 0.6466 - val_accuracy: 0.8850
Epoch 187/200
29/29 [==============================] - 4s 134ms/step - loss: 0.1497 -
accuracy: 0.9423 - val_loss: 0.6888 - val_accuracy: 0.8841
Epoch 188/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1568 - accuracy:
0.9425 - val_loss: 0.6533 - val_accuracy: 0.8804
Epoch 189/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1466 - accuracy:
0.9451 - val_loss: 0.7104 - val_accuracy: 0.8776
Epoch 190/200
29/29 [==============================] - 3s 109ms/step - loss: 0.1386 -
accuracy: 0.9479 - val_loss: 0.6428 - val_accuracy: 0.8822
Epoch 191/200
29/29 [==============================] - 5s 181ms/step - loss: 0.1529 -
accuracy: 0.9437 - val_loss: 0.6663 - val_accuracy: 0.8841
Epoch 192/200
29/29 [==============================] - 4s 147ms/step - loss: 0.1499 -
accuracy: 0.9437 - val_loss: 0.6739 - val_accuracy: 0.8794
Epoch 193/200
```

```
29/29 [==============================] - 3s 117ms/step - loss: 0.1556 -
accuracy: 0.9393 - val_loss: 0.6530 - val_accuracy: 0.8860
Epoch 194/200
29/29 [==============================] - 3s 105ms/step - loss: 0.1561 -
accuracy: 0.9418 - val_loss: 0.6767 - val_accuracy: 0.8766
Epoch 195/200
29/29 [==============================] - 3s 101ms/step - loss: 0.1460 -
accuracy: 0.9430 - val_loss: 0.6709 - val_accuracy: 0.8794
Epoch 196/200
29/29 [==============================] - 4s 145ms/step - loss: 0.1295 -
accuracy: 0.9502 - val_loss: 0.6718 - val_accuracy: 0.8879
Epoch 197/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1546 - accuracy:
0.9411 - val_loss: 0.6356 - val_accuracy: 0.8897
Epoch 198/200
29/29 [==============================] - 2s 80ms/step - loss: 0.1318 - accuracy:
0.9521 - val_loss: 0.6903 - val_accuracy: 0.8879
Epoch 199/200
29/29 [==============================] - 2s 78ms/step - loss: 0.1480 - accuracy:
0.9465 - val_loss: 0.6562 - val_accuracy: 0.8888
Epoch 200/200
29/29 [==============================] - 2s 86ms/step - loss: 0.1460 - accuracy:
0.9470 - val_loss: 0.7013 - val_accuracy: 0.8832
```

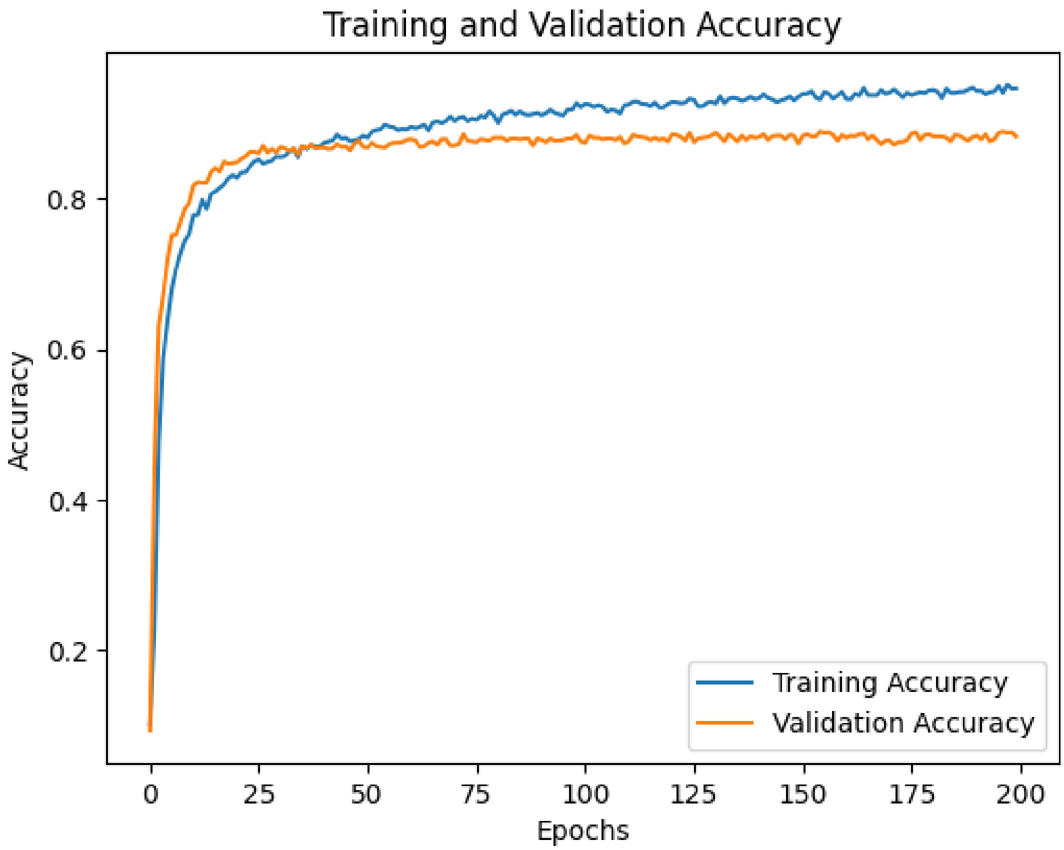
Evaluating the Model performance

[164]:
```python
scores = model.evaluate(X_test, y_test, verbose=1)
print('Test loss:', scores[0])
print('Test accuracy:', scores[1])
```

```
34/34 [==============================] - 0s 7ms/step - loss: 0.7013 - accuracy:
0.8832
Test loss: 0.701292872428894
Test accuracy: 0.8831775784492493
```

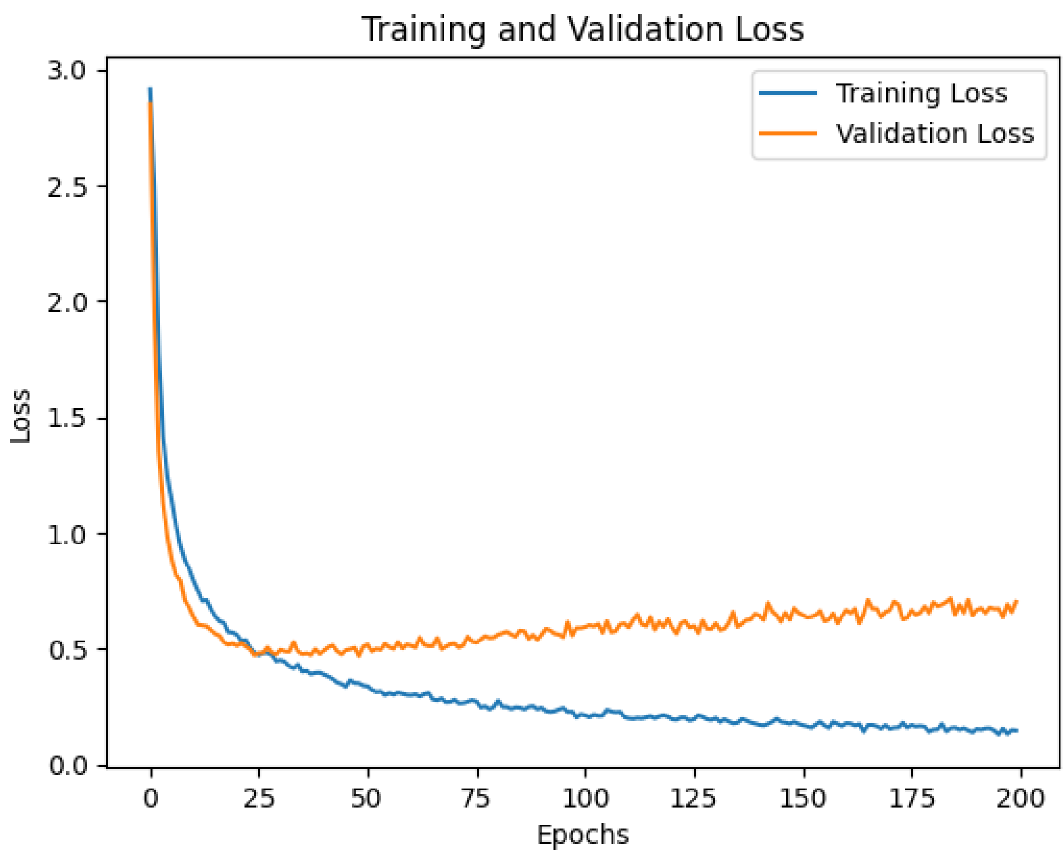
Plot for training and validation accuracy

[165]:
```python
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



Plot for training and validation loss

[166]:
```
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

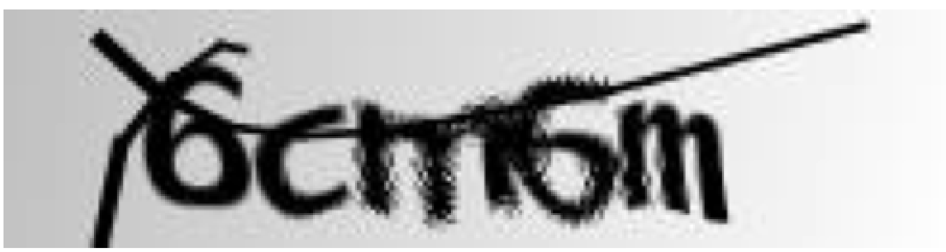


Predicting with new image

[167]:
```
def predict_captcha(img_path) :

    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

    plt.imshow(img, cmap='gray')
    plt.axis('off')
    plt.show()

    image = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.
 ↪THRESH_BINARY, 145, 0)
    image = cv2.adaptiveThreshold(image, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,␣
 ↪cv2.THRESH_BINARY, 145, 0)
    kernel = np.ones((5,5),np.uint8)
    image = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)
    kernel = np.ones((2,2),np.uint8)
    image = cv2.dilate(image, kernel, iterations = 1)
    image = cv2.GaussianBlur(image, (5,5), 0)
```

```python
    x = [image[10:50, 30:50], image[10:50, 50:70], image[10:50, 70:90],
                  image[10:50, 90:110], image[10:50, 110:130]]

    X_pred = []
    for i in range(5) :
        X_pred.append(img_to_array(Image.fromarray(x[i])))

    X_pred = np.array(X_pred)
    X_pred/= 255.0

    y_pred = model.predict(X_pred)
    y_pred = np.argmax(y_pred, axis = 1)

    print('Prediction: ', end='')
    for res in y_pred :
        print(info[res], end='')

    print('\nActual:    ', img_path[len(img_path)-9:len(img_path)-4])
```

[168]:
```python
predict_captcha('/content/drive/MyDrive/CAPTCHA/samples/6cm6m.png')
```



```
1/1 [==============================] - 0s 156ms/step
Prediction: 6cn6m
Actual:     6cm6m
```

Save the model

[169]:
```python
model.save('cnn_model.h5')

with open('cnn_model.pkl', 'wb') as f:
    pickle.dump(model, f)

print("Model saved successfully!")
```

```
Model saved successfully!
```