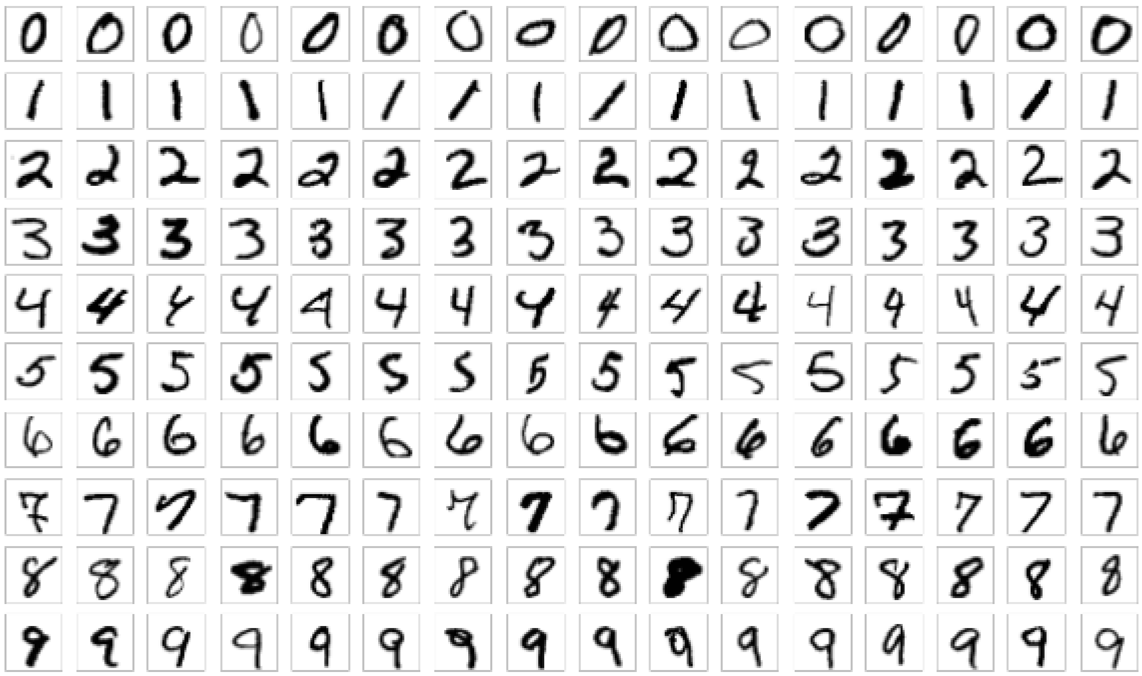
Handwritten Digit Classifier Using the MNIST Dataset

CAP 6619 - Deep Learning | Project 1 | Summer 2022 - Dr. Marques | Matthew Acs



This project will create and evaluate a handwritten digit classifier based on the MNIST dataset

References and Sources

The following links contain useful resources that provide information relating to this project. Some of the sources provide background information while others answer questions specifically relating to the implementation of the classifier.

- https://www.tensorflow.org/datasets/catalog/mnist

  Background information on the MNIST dataset

- https://en.wikipedia.org/wiki/MNIST_database

  Background information on the MNIST dataset

- https://keras.io/examples/vision/mnist_convnet/

  An example CNN classifier for the MNIST dataset

- https://github.com/the-deep-learners/deep-learning-illustrated/blob/master/notebooks/shallow_net_in_keras.ipynb

  An example shallow classifier for the MNIST dataset

- https://stackoverflow.com/questions/68836551/keras-attributeerror-sequential-object-has-no-attribute-predict-classes

  How to get an array of the predictions from a Keras classifier

---

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html#sklearn.metrics.C

  Scikit-learn documentation to create a confusion matrix

- https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html

  Scikit-learn documentation example of a confusion matrix for a digit classifier

Setup

In [1]:

```
from tensorflow import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras.optimizers import SGD
from sklearn import metrics
from tensorflow.keras import layers
from matplotlib import pyplot as plt
import numpy as np
```

Load and prepare the data

In [2]:

```
# Model / data parameters
num_classes = 10
input_shape = (28, 28, 1)

# the data, split between train and validation sets
(X_train, y_train), (X_valid, y_valid) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.
npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

In [3]:

```
X_train.shape
```

Out[3]:
```
(60000, 28, 28)
```

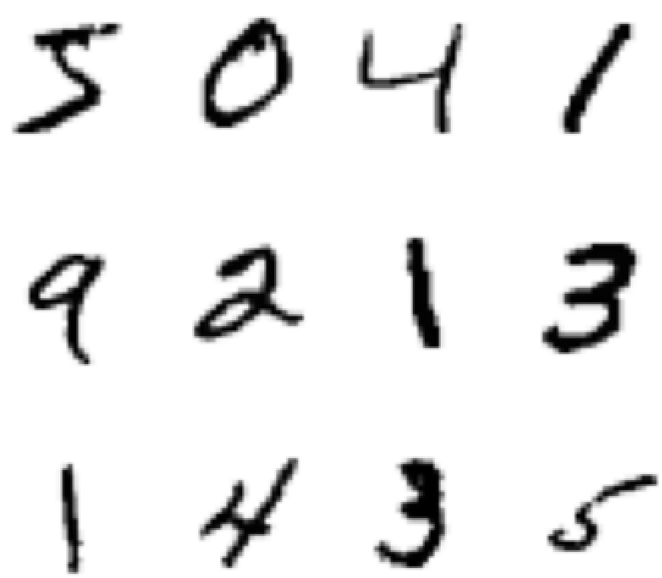
In [4]:
```
y_train.shape
```

Out[4]:
```
(60000,)
```

In [5]:
```
y_train[0:12]
```

Out[5]:
```
array([5, 0, 4, 1, 9, 2, 1, 3, 1, 4, 3, 5], dtype=uint8)
```

In [6]:
```
plt.figure(figsize=(5,5))
for k in range(12):
    plt.subplot(3, 4, k+1)
    plt.imshow(X_train[k], cmap='Greys')
    plt.axis('off')
plt.tight_layout()
plt.show()
```



In [7]:
```
X_valid.shape
```

Out[7]:
```
(10000, 28, 28)
```

In [8]:
```
y_valid.shape
```

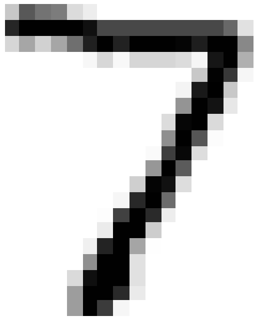
Out[8]:
```
(10000,)
```

In [9]:
```
y_valid[0]
```

Out[9]:
```
7
```

In [10]:
```
plt.imshow(X_valid[0], cmap='Greys')
```

```
plt.axis('off')
plt.show()
```



In [11]:

```
# Reshape (flatten) images
X_train_reshaped = X_train.reshape(60000, 784).astype('float32')
X_valid_reshaped = X_valid.reshape(10000, 784).astype('float32')

# Scale images to the [0, 1] range
X_train_scaled_reshaped = X_train_reshaped / 255
X_valid_scaled_reshaped = X_valid_reshaped / 255

# Renaming for conciseness
X_training = X_train_scaled_reshaped
X_validation = X_valid_scaled_reshaped

print("X_training shape (after reshaping + scaling):", X_training.shape)
print(X_training.shape[0], "train samples")
print("X_validation shape (after reshaping + scaling):", X_validation.shape)
print(X_validation.shape[0], "validation samples")
```

```
X_training shape (after reshaping + scaling): (60000, 784)
60000 train samples
X_validation shape (after reshaping + scaling): (10000, 784)
10000 validation samples
```

In [12]:

```
# convert class vectors to binary class matrices
y_training = keras.utils.to_categorical(y_train, num_classes)
y_validation = keras.utils.to_categorical(y_valid, num_classes)
```

In [13]:

```
print(y_valid[0])
print(y_validation[0])
```

```
7
[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
```

Question 1

Explain the meaning and contents of X_training, y_training, X_validation, and y_validation

X_training is a tensor that contains the data for training the model while y_training is a tensor that contains the labels for the training data. X_validation is a tensor that contains the data for validating the model while y_validation is a tensor that contains the labels for the validation data. All four tensors are rank-2 tensors because they are arrays of arrays. The labels are represented in an array such that there are ten elements in each array and a 1 is present at the index that corresponds to the label. For example, the label 2 is represented by [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]. The data is represented in an array that was created by flattening a 28 by 28 pixel grid into an array of 784 elements. Each element in the array represents a greyscale pixel with a value in the range of 0 to 1 with 1 being black and 0 being white.

As an example, X_training [0] would give you an array of 784 numbers in the range 0-1 that represents the first digit in the dataset. y_training [0] would give you an array of 10 numbers with a 1 in the index that represents the label of the first digit in the dataset. X_training [1] and y_training [1] would give you the data and label for the second digit in the dataset and so on.

PART 1 - Shallow neural network architecture

In [14]:

```
model = Sequential()
model.add(Dense(64, activation='sigmoid', input_shape=(784,)))
model.add(Dense(10, activation='softmax'))
```

In [15]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                50240

 dense_1 (Dense)             (None, 10)                650

=================================================================
Total params: 50,890
Trainable params: 50,890
Non-trainable params: 0
_________________________________________________________________
```

Question 2

Explain the meaning of the results you get when you run model.summary()

When you run model.summary(), the output shows the structure of the neural network's architecture. It displays the Keras model type along with the layers and the information associated with each layer. The summary shows us that the model type is sequential and that there are two layers in the model. The first layer has an output shape of 64 and contains 50,240 parameters. This output shape refers to the fact that the output of the first layer will be a rank-2 tensor with the innermost array having 64 elements in it. The 50,240 parameters are the weights and biases that the model updates in order to train the model. There are 64 outputs and 784 inputs and there is one weight for each connection from the input to the layer, so there are 784 *64 = 50,176 weights. Each output also has a bias so there are 50,176 + 64 = 50,240 total trainable parameters. The second layer has an output shape of 10 and contains 650 parameters. This means that the output of the layer will be a rank-2 tensor with the innermost array containing 10 elements. This is the output layer of the model, so the 10 elements refer to the probability that the input is in one of the 10 classes. It contains 650 parameters because 64 inputs are each connected to 10 outputs with a weight for each connection, thus totaling 64* 10 = 640 weights. Adding the 10 biases for each output makes 640 + 10 = 650 trainable parameters for the second layer. Thus, adding all of the parameters you get 50,890. The bottom section then shows us that there are a total of 50,890 total parameters and trainable parameters.

In [16]:

```
(64*784)
```

Out[16]:

```
50176
```

In [17]:

```
(64*784)+64
```

Out[17]:

```
50240
```

In [18]:

```
(10*64)+10
```

Out[18]:

```
650
```

Configure model

The following two configurations will configure the model using different loss functions and optimizers. The second configuration with a loss function of mean_squared_error had about a 90% accuracy while the first configuration with a loss function of poisson had an accuracy of about 82%. After experimentation, it can be seen that the choice of loss function and optimizer has a significant impact on the performance of the model. The configuration is currently set to the original with a loss function of mean_squared_error.

In [19]:

```
x = 0

if x == 1:
  model.compile(
      loss='poisson',
      optimizer='adadelta',
      metrics=['accuracy']
  )
else:
  model.compile(
      loss='mean_squared_error',
      optimizer=SGD(learning_rate=0.01),
      metrics=['accuracy']
  )
```

Train!

In [20]:
```
batch_size=128
epochs=500

history = model.fit(
  X_training, # training data
  y_training, # training targets
  epochs=epochs,
  batch_size=batch_size,
  verbose=1,
  validation_data=(X_validation, y_validation)
)
```

```
Epoch 1/500
469/469 [==============================] - 4s 4ms/step - loss: 0.0936 - accuracy: 0.0971
- val_loss: 0.0927 - val_accuracy: 0.0924
Epoch 2/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0921 - accuracy: 0.0870
- val_loss: 0.0916 - val_accuracy: 0.0809
Epoch 3/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0912 - accuracy: 0.0864
- val_loss: 0.0909 - val_accuracy: 0.0854
Epoch 4/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0906 - accuracy: 0.0939
- val_loss: 0.0903 - val_accuracy: 0.0920
Epoch 5/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0901 - accuracy: 0.1046
- val_loss: 0.0899 - val_accuracy: 0.1051
Epoch 6/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0897 - accuracy: 0.1178
- val_loss: 0.0895 - val_accuracy: 0.1195
Epoch 7/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0893 - accuracy: 0.1402
- val_loss: 0.0891 - val_accuracy: 0.1527
Epoch 8/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0890 - accuracy: 0.1825
- val_loss: 0.0888 - val_accuracy: 0.2047
Epoch 9/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0887 - accuracy: 0.2304
- val_loss: 0.0885 - val_accuracy: 0.2533
Epoch 10/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0884 - accuracy: 0.2587
- val_loss: 0.0882 - val_accuracy: 0.2765
Epoch 11/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0881 - accuracy: 0.2734
- val_loss: 0.0880 - val_accuracy: 0.2867
Epoch 12/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0879 - accuracy: 0.2830
- val_loss: 0.0877 - val_accuracy: 0.2926
Epoch 13/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0876 - accuracy: 0.2912
- val_loss: 0.0874 - val_accuracy: 0.3009
Epoch 14/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0873 - accuracy: 0.3012
- val_loss: 0.0871 - val_accuracy: 0.3163
Epoch 15/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0870 - accuracy: 0.3158
- val_loss: 0.0868 - val_accuracy: 0.3313
```

```
Epoch 16/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0867 - accuracy: 0.3324
- val_loss: 0.0865 - val_accuracy: 0.3510
Epoch 17/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0865 - accuracy: 0.3491
- val_loss: 0.0862 - val_accuracy: 0.3671
Epoch 18/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0862 - accuracy: 0.3647
- val_loss: 0.0859 - val_accuracy: 0.3786
Epoch 19/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0859 - accuracy: 0.3791
- val_loss: 0.0856 - val_accuracy: 0.3893
Epoch 20/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0856 - accuracy: 0.3903
- val_loss: 0.0853 - val_accuracy: 0.3996
Epoch 21/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0852 - accuracy: 0.4028
- val_loss: 0.0850 - val_accuracy: 0.4079
Epoch 22/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0849 - accuracy: 0.4098
- val_loss: 0.0847 - val_accuracy: 0.4156
Epoch 23/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0846 - accuracy: 0.4168
- val_loss: 0.0843 - val_accuracy: 0.4245
Epoch 24/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0843 - accuracy: 0.4227
- val_loss: 0.0840 - val_accuracy: 0.4330
Epoch 25/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0839 - accuracy: 0.4302
- val_loss: 0.0836 - val_accuracy: 0.4396
Epoch 26/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0836 - accuracy: 0.4347
- val_loss: 0.0833 - val_accuracy: 0.4468
Epoch 27/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0832 - accuracy: 0.4412
- val_loss: 0.0829 - val_accuracy: 0.4542
Epoch 28/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0828 - accuracy: 0.4480
- val_loss: 0.0825 - val_accuracy: 0.4592
Epoch 29/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0824 - accuracy: 0.4525
- val_loss: 0.0821 - val_accuracy: 0.4634
Epoch 30/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0821 - accuracy: 0.4567
- val_loss: 0.0817 - val_accuracy: 0.4681
Epoch 31/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0817 - accuracy: 0.4629
- val_loss: 0.0813 - val_accuracy: 0.4742
Epoch 32/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0813 - accuracy: 0.4692
- val_loss: 0.0809 - val_accuracy: 0.4794
Epoch 33/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0808 - accuracy: 0.4747
- val_loss: 0.0805 - val_accuracy: 0.4847
Epoch 34/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0804 - accuracy: 0.4790
- val_loss: 0.0800 - val_accuracy: 0.4903
Epoch 35/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0800 - accuracy: 0.4855
- val_loss: 0.0796 - val_accuracy: 0.4961
```

```
Epoch 36/500
469/469 [==============================] - 2s 5ms/step - loss: 0.0796 - accuracy: 0.4903
- val_loss: 0.0791 - val_accuracy: 0.5009
Epoch 37/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0791 - accuracy: 0.4958
- val_loss: 0.0787 - val_accuracy: 0.5065
Epoch 38/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0787 - accuracy: 0.5018
- val_loss: 0.0782 - val_accuracy: 0.5127
Epoch 39/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0782 - accuracy: 0.5083
- val_loss: 0.0778 - val_accuracy: 0.5166
Epoch 40/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0778 - accuracy: 0.5128
- val_loss: 0.0773 - val_accuracy: 0.5228
Epoch 41/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0773 - accuracy: 0.5188
- val_loss: 0.0768 - val_accuracy: 0.5280
Epoch 42/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0768 - accuracy: 0.5239
- val_loss: 0.0763 - val_accuracy: 0.5325
Epoch 43/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0763 - accuracy: 0.5284
- val_loss: 0.0759 - val_accuracy: 0.5381
Epoch 44/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0759 - accuracy: 0.5342
- val_loss: 0.0754 - val_accuracy: 0.5425
Epoch 45/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0754 - accuracy: 0.5383
- val_loss: 0.0749 - val_accuracy: 0.5474
Epoch 46/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0749 - accuracy: 0.5436
- val_loss: 0.0744 - val_accuracy: 0.5529
Epoch 47/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0744 - accuracy: 0.5487
- val_loss: 0.0739 - val_accuracy: 0.5570
Epoch 48/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0739 - accuracy: 0.5534
- val_loss: 0.0734 - val_accuracy: 0.5608
Epoch 49/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0734 - accuracy: 0.5582
- val_loss: 0.0729 - val_accuracy: 0.5650
Epoch 50/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0729 - accuracy: 0.5621
- val_loss: 0.0724 - val_accuracy: 0.5692
Epoch 51/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0724 - accuracy: 0.5667
- val_loss: 0.0719 - val_accuracy: 0.5722
Epoch 52/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0719 - accuracy: 0.5704
- val_loss: 0.0714 - val_accuracy: 0.5766
Epoch 53/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0714 - accuracy: 0.5743
- val_loss: 0.0708 - val_accuracy: 0.5800
Epoch 54/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0709 - accuracy: 0.5783
- val_loss: 0.0703 - val_accuracy: 0.5834
Epoch 55/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0704 - accuracy: 0.5826
- val_loss: 0.0698 - val_accuracy: 0.5867
```

```
Epoch 56/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0699 - accuracy: 0.5859
- val_loss: 0.0693 - val_accuracy: 0.5896
Epoch 57/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0694 - accuracy: 0.5891
- val_loss: 0.0688 - val_accuracy: 0.5922
Epoch 58/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0689 - accuracy: 0.5921
- val_loss: 0.0683 - val_accuracy: 0.5959
Epoch 59/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0684 - accuracy: 0.5948
- val_loss: 0.0678 - val_accuracy: 0.5998
Epoch 60/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0679 - accuracy: 0.5982
- val_loss: 0.0672 - val_accuracy: 0.6026
Epoch 61/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0674 - accuracy: 0.6008
- val_loss: 0.0667 - val_accuracy: 0.6048
Epoch 62/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0668 - accuracy: 0.6032
- val_loss: 0.0662 - val_accuracy: 0.6061
Epoch 63/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0663 - accuracy: 0.6059
- val_loss: 0.0657 - val_accuracy: 0.6084
Epoch 64/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0658 - accuracy: 0.6082
- val_loss: 0.0652 - val_accuracy: 0.6117
Epoch 65/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0653 - accuracy: 0.6113
- val_loss: 0.0647 - val_accuracy: 0.6144
Epoch 66/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0648 - accuracy: 0.6136
- val_loss: 0.0642 - val_accuracy: 0.6159
Epoch 67/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0643 - accuracy: 0.6160
- val_loss: 0.0637 - val_accuracy: 0.6186
Epoch 68/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0638 - accuracy: 0.6183
- val_loss: 0.0632 - val_accuracy: 0.6209
Epoch 69/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0633 - accuracy: 0.6210
- val_loss: 0.0627 - val_accuracy: 0.6225
Epoch 70/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0629 - accuracy: 0.6228
- val_loss: 0.0622 - val_accuracy: 0.6244
Epoch 71/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0624 - accuracy: 0.6251
- val_loss: 0.0617 - val_accuracy: 0.6277
Epoch 72/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0619 - accuracy: 0.6277
- val_loss: 0.0612 - val_accuracy: 0.6302
Epoch 73/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0614 - accuracy: 0.6298
- val_loss: 0.0607 - val_accuracy: 0.6319
Epoch 74/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0609 - accuracy: 0.6324
- val_loss: 0.0603 - val_accuracy: 0.6345
Epoch 75/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0605 - accuracy: 0.6346
- val_loss: 0.0598 - val_accuracy: 0.6362
```

```
Epoch 76/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0600 - accuracy: 0.6369
- val_loss: 0.0593 - val_accuracy: 0.6388
Epoch 77/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0595 - accuracy: 0.6395
- val_loss: 0.0588 - val_accuracy: 0.6410
Epoch 78/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0591 - accuracy: 0.6416
- val_loss: 0.0584 - val_accuracy: 0.6436
Epoch 79/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0586 - accuracy: 0.6440
- val_loss: 0.0579 - val_accuracy: 0.6452
Epoch 80/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0582 - accuracy: 0.6464
- val_loss: 0.0575 - val_accuracy: 0.6475
Epoch 81/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0577 - accuracy: 0.6484
- val_loss: 0.0570 - val_accuracy: 0.6498
Epoch 82/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0573 - accuracy: 0.6506
- val_loss: 0.0566 - val_accuracy: 0.6522
Epoch 83/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0568 - accuracy: 0.6528
- val_loss: 0.0561 - val_accuracy: 0.6557
Epoch 84/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0564 - accuracy: 0.6550
- val_loss: 0.0557 - val_accuracy: 0.6578
Epoch 85/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0560 - accuracy: 0.6576
- val_loss: 0.0553 - val_accuracy: 0.6597
Epoch 86/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0555 - accuracy: 0.6594
- val_loss: 0.0548 - val_accuracy: 0.6610
Epoch 87/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0551 - accuracy: 0.6622
- val_loss: 0.0544 - val_accuracy: 0.6640
Epoch 88/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0547 - accuracy: 0.6647
- val_loss: 0.0540 - val_accuracy: 0.6661
Epoch 89/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0543 - accuracy: 0.6674
- val_loss: 0.0536 - val_accuracy: 0.6686
Epoch 90/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0539 - accuracy: 0.6699
- val_loss: 0.0532 - val_accuracy: 0.6718
Epoch 91/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0535 - accuracy: 0.6728
- val_loss: 0.0528 - val_accuracy: 0.6749
Epoch 92/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0531 - accuracy: 0.6750
- val_loss: 0.0524 - val_accuracy: 0.6780
Epoch 93/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0527 - accuracy: 0.6775
- val_loss: 0.0520 - val_accuracy: 0.6810
Epoch 94/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0523 - accuracy: 0.6806
- val_loss: 0.0516 - val_accuracy: 0.6844
Epoch 95/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0519 - accuracy: 0.6830
- val_loss: 0.0512 - val_accuracy: 0.6870
```

```
Epoch 96/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0516 - accuracy: 0.6857
- val_loss: 0.0508 - val_accuracy: 0.6897
Epoch 97/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0512 - accuracy: 0.6882
- val_loss: 0.0505 - val_accuracy: 0.6928
Epoch 98/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0508 - accuracy: 0.6903
- val_loss: 0.0501 - val_accuracy: 0.6953
Epoch 99/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0505 - accuracy: 0.6929
- val_loss: 0.0497 - val_accuracy: 0.6982
Epoch 100/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0501 - accuracy: 0.6957
- val_loss: 0.0494 - val_accuracy: 0.7017
Epoch 101/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0497 - accuracy: 0.6982
- val_loss: 0.0490 - val_accuracy: 0.7049
Epoch 102/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0494 - accuracy: 0.7009
- val_loss: 0.0487 - val_accuracy: 0.7090
Epoch 103/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0490 - accuracy: 0.7039
- val_loss: 0.0483 - val_accuracy: 0.7124
Epoch 104/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0487 - accuracy: 0.7064
- val_loss: 0.0480 - val_accuracy: 0.7159
Epoch 105/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0484 - accuracy: 0.7094
- val_loss: 0.0476 - val_accuracy: 0.7191
Epoch 106/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0480 - accuracy: 0.7124
- val_loss: 0.0473 - val_accuracy: 0.7217
Epoch 107/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0477 - accuracy: 0.7148
- val_loss: 0.0470 - val_accuracy: 0.7245
Epoch 108/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0474 - accuracy: 0.7171
- val_loss: 0.0466 - val_accuracy: 0.7269
Epoch 109/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0470 - accuracy: 0.7196
- val_loss: 0.0463 - val_accuracy: 0.7299
Epoch 110/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0467 - accuracy: 0.7222
- val_loss: 0.0460 - val_accuracy: 0.7327
Epoch 111/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0464 - accuracy: 0.7247
- val_loss: 0.0457 - val_accuracy: 0.7349
Epoch 112/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0461 - accuracy: 0.7272
- val_loss: 0.0454 - val_accuracy: 0.7372
Epoch 113/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0458 - accuracy: 0.7298
- val_loss: 0.0450 - val_accuracy: 0.7405
Epoch 114/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0455 - accuracy: 0.7326
- val_loss: 0.0447 - val_accuracy: 0.7427
Epoch 115/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0452 - accuracy: 0.7355
- val_loss: 0.0444 - val_accuracy: 0.7448
```

```
Epoch 116/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0449 - accuracy: 0.7380
- val_loss: 0.0441 - val_accuracy: 0.7481
Epoch 117/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0446 - accuracy: 0.7407
- val_loss: 0.0438 - val_accuracy: 0.7519
Epoch 118/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0443 - accuracy: 0.7433
- val_loss: 0.0435 - val_accuracy: 0.7545
Epoch 119/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0440 - accuracy: 0.7469
- val_loss: 0.0433 - val_accuracy: 0.7580
Epoch 120/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0437 - accuracy: 0.7496
- val_loss: 0.0430 - val_accuracy: 0.7609
Epoch 121/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0434 - accuracy: 0.7528
- val_loss: 0.0427 - val_accuracy: 0.7636
Epoch 122/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0432 - accuracy: 0.7554
- val_loss: 0.0424 - val_accuracy: 0.7660
Epoch 123/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0429 - accuracy: 0.7580
- val_loss: 0.0421 - val_accuracy: 0.7688
Epoch 124/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0426 - accuracy: 0.7609
- val_loss: 0.0418 - val_accuracy: 0.7713
Epoch 125/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0423 - accuracy: 0.7640
- val_loss: 0.0416 - val_accuracy: 0.7744
Epoch 126/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0421 - accuracy: 0.7668
- val_loss: 0.0413 - val_accuracy: 0.7775
Epoch 127/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0418 - accuracy: 0.7701
- val_loss: 0.0410 - val_accuracy: 0.7802
Epoch 128/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0416 - accuracy: 0.7724
- val_loss: 0.0408 - val_accuracy: 0.7827
Epoch 129/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0413 - accuracy: 0.7753
- val_loss: 0.0405 - val_accuracy: 0.7857
Epoch 130/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0410 - accuracy: 0.7785
- val_loss: 0.0403 - val_accuracy: 0.7886
Epoch 131/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0408 - accuracy: 0.7810
- val_loss: 0.0400 - val_accuracy: 0.7920
Epoch 132/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0405 - accuracy: 0.7833
- val_loss: 0.0398 - val_accuracy: 0.7949
Epoch 133/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0403 - accuracy: 0.7855
- val_loss: 0.0395 - val_accuracy: 0.7981
Epoch 134/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0401 - accuracy: 0.7894
- val_loss: 0.0393 - val_accuracy: 0.8008
Epoch 135/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0398 - accuracy: 0.7917
- val_loss: 0.0390 - val_accuracy: 0.8034
```

```
Epoch 136/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0396 - accuracy: 0.7941
- val_loss: 0.0388 - val_accuracy: 0.8059
Epoch 137/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0393 - accuracy: 0.7961
- val_loss: 0.0386 - val_accuracy: 0.8090
Epoch 138/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0391 - accuracy: 0.7985
- val_loss: 0.0383 - val_accuracy: 0.8108
Epoch 139/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0389 - accuracy: 0.8013
- val_loss: 0.0381 - val_accuracy: 0.8133
Epoch 140/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0387 - accuracy: 0.8032
- val_loss: 0.0379 - val_accuracy: 0.8157
Epoch 141/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0384 - accuracy: 0.8052
- val_loss: 0.0376 - val_accuracy: 0.8166
Epoch 142/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0382 - accuracy: 0.8078
- val_loss: 0.0374 - val_accuracy: 0.8192
Epoch 143/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0380 - accuracy: 0.8094
- val_loss: 0.0372 - val_accuracy: 0.8213
Epoch 144/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0378 - accuracy: 0.8115
- val_loss: 0.0370 - val_accuracy: 0.8229
Epoch 145/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0376 - accuracy: 0.8136
- val_loss: 0.0368 - val_accuracy: 0.8244
Epoch 146/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0374 - accuracy: 0.8154
- val_loss: 0.0365 - val_accuracy: 0.8254
Epoch 147/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0371 - accuracy: 0.8167
- val_loss: 0.0363 - val_accuracy: 0.8271
Epoch 148/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0369 - accuracy: 0.8183
- val_loss: 0.0361 - val_accuracy: 0.8286
Epoch 149/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0367 - accuracy: 0.8200
- val_loss: 0.0359 - val_accuracy: 0.8297
Epoch 150/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0365 - accuracy: 0.8210
- val_loss: 0.0357 - val_accuracy: 0.8317
Epoch 151/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0363 - accuracy: 0.8231
- val_loss: 0.0355 - val_accuracy: 0.8330
Epoch 152/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0361 - accuracy: 0.8242
- val_loss: 0.0353 - val_accuracy: 0.8344
Epoch 153/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0359 - accuracy: 0.8252
- val_loss: 0.0351 - val_accuracy: 0.8351
Epoch 154/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0357 - accuracy: 0.8263
- val_loss: 0.0349 - val_accuracy: 0.8356
Epoch 155/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0356 - accuracy: 0.8278
- val_loss: 0.0347 - val_accuracy: 0.8362
```

```
Epoch 156/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0354 - accuracy: 0.8288
- val_loss: 0.0345 - val_accuracy: 0.8372
Epoch 157/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0352 - accuracy: 0.8300
- val_loss: 0.0343 - val_accuracy: 0.8379
Epoch 158/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0350 - accuracy: 0.8308
- val_loss: 0.0342 - val_accuracy: 0.8390
Epoch 159/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0348 - accuracy: 0.8316
- val_loss: 0.0340 - val_accuracy: 0.8404
Epoch 160/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0346 - accuracy: 0.8327
- val_loss: 0.0338 - val_accuracy: 0.8408
Epoch 161/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0345 - accuracy: 0.8340
- val_loss: 0.0336 - val_accuracy: 0.8419
Epoch 162/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0343 - accuracy: 0.8353
- val_loss: 0.0334 - val_accuracy: 0.8425
Epoch 163/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0341 - accuracy: 0.8363
- val_loss: 0.0333 - val_accuracy: 0.8438
Epoch 164/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0339 - accuracy: 0.8372
- val_loss: 0.0331 - val_accuracy: 0.8443
Epoch 165/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0338 - accuracy: 0.8380
- val_loss: 0.0329 - val_accuracy: 0.8448
Epoch 166/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0336 - accuracy: 0.8390
- val_loss: 0.0327 - val_accuracy: 0.8461
Epoch 167/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0334 - accuracy: 0.8399
- val_loss: 0.0326 - val_accuracy: 0.8466
Epoch 168/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0333 - accuracy: 0.8406
- val_loss: 0.0324 - val_accuracy: 0.8475
Epoch 169/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0331 - accuracy: 0.8414
- val_loss: 0.0322 - val_accuracy: 0.8477
Epoch 170/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0329 - accuracy: 0.8422
- val_loss: 0.0321 - val_accuracy: 0.8485
Epoch 171/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0328 - accuracy: 0.8430
- val_loss: 0.0319 - val_accuracy: 0.8492
Epoch 172/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0326 - accuracy: 0.8434
- val_loss: 0.0318 - val_accuracy: 0.8505
Epoch 173/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0325 - accuracy: 0.8441
- val_loss: 0.0316 - val_accuracy: 0.8510
Epoch 174/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0323 - accuracy: 0.8446
- val_loss: 0.0315 - val_accuracy: 0.8521
Epoch 175/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0322 - accuracy: 0.8456
- val_loss: 0.0313 - val_accuracy: 0.8529
```

```
Epoch 176/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0320 - accuracy: 0.8463
- val_loss: 0.0312 - val_accuracy: 0.8536
Epoch 177/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0319 - accuracy: 0.8468
- val_loss: 0.0310 - val_accuracy: 0.8541
Epoch 178/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0317 - accuracy: 0.8475
- val_loss: 0.0309 - val_accuracy: 0.8547
Epoch 179/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0316 - accuracy: 0.8480
- val_loss: 0.0307 - val_accuracy: 0.8554
Epoch 180/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0314 - accuracy: 0.8490
- val_loss: 0.0306 - val_accuracy: 0.8561
Epoch 181/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0313 - accuracy: 0.8493
- val_loss: 0.0304 - val_accuracy: 0.8574
Epoch 182/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0312 - accuracy: 0.8498
- val_loss: 0.0303 - val_accuracy: 0.8579
Epoch 183/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0310 - accuracy: 0.8505
- val_loss: 0.0301 - val_accuracy: 0.8588
Epoch 184/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0309 - accuracy: 0.8509
- val_loss: 0.0300 - val_accuracy: 0.8594
Epoch 185/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0307 - accuracy: 0.8513
- val_loss: 0.0299 - val_accuracy: 0.8598
Epoch 186/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0306 - accuracy: 0.8517
- val_loss: 0.0297 - val_accuracy: 0.8601
Epoch 187/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0305 - accuracy: 0.8522
- val_loss: 0.0296 - val_accuracy: 0.8605
Epoch 188/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0304 - accuracy: 0.8526
- val_loss: 0.0295 - val_accuracy: 0.8613
Epoch 189/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0302 - accuracy: 0.8532
- val_loss: 0.0293 - val_accuracy: 0.8619
Epoch 190/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0301 - accuracy: 0.8536
- val_loss: 0.0292 - val_accuracy: 0.8623
Epoch 191/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0300 - accuracy: 0.8540
- val_loss: 0.0291 - val_accuracy: 0.8628
Epoch 192/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0298 - accuracy: 0.8544
- val_loss: 0.0290 - val_accuracy: 0.8631
Epoch 193/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0297 - accuracy: 0.8549
- val_loss: 0.0288 - val_accuracy: 0.8634
Epoch 194/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0296 - accuracy: 0.8554
- val_loss: 0.0287 - val_accuracy: 0.8639
Epoch 195/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0295 - accuracy: 0.8557
- val_loss: 0.0286 - val_accuracy: 0.8642
```

```
Epoch 196/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0294 - accuracy: 0.8564
- val_loss: 0.0285 - val_accuracy: 0.8646
Epoch 197/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0292 - accuracy: 0.8565
- val_loss: 0.0284 - val_accuracy: 0.8651
Epoch 198/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0291 - accuracy: 0.8570
- val_loss: 0.0282 - val_accuracy: 0.8655
Epoch 199/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0290 - accuracy: 0.8574
- val_loss: 0.0281 - val_accuracy: 0.8661
Epoch 200/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0289 - accuracy: 0.8578
- val_loss: 0.0280 - val_accuracy: 0.8667
Epoch 201/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0288 - accuracy: 0.8582
- val_loss: 0.0279 - val_accuracy: 0.8670
Epoch 202/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0287 - accuracy: 0.8585
- val_loss: 0.0278 - val_accuracy: 0.8675
Epoch 203/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0286 - accuracy: 0.8589
- val_loss: 0.0277 - val_accuracy: 0.8681
Epoch 204/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0285 - accuracy: 0.8591
- val_loss: 0.0276 - val_accuracy: 0.8684
Epoch 205/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0284 - accuracy: 0.8594
- val_loss: 0.0274 - val_accuracy: 0.8688
Epoch 206/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0282 - accuracy: 0.8598
- val_loss: 0.0273 - val_accuracy: 0.8695
Epoch 207/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0281 - accuracy: 0.8602
- val_loss: 0.0272 - val_accuracy: 0.8696
Epoch 208/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0280 - accuracy: 0.8605
- val_loss: 0.0271 - val_accuracy: 0.8702
Epoch 209/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0279 - accuracy: 0.8608
- val_loss: 0.0270 - val_accuracy: 0.8702
Epoch 210/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0278 - accuracy: 0.8612
- val_loss: 0.0269 - val_accuracy: 0.8703
Epoch 211/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0277 - accuracy: 0.8616
- val_loss: 0.0268 - val_accuracy: 0.8705
Epoch 212/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0276 - accuracy: 0.8619
- val_loss: 0.0267 - val_accuracy: 0.8708
Epoch 213/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0275 - accuracy: 0.8622
- val_loss: 0.0266 - val_accuracy: 0.8710
Epoch 214/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0274 - accuracy: 0.8626
- val_loss: 0.0265 - val_accuracy: 0.8716
Epoch 215/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0273 - accuracy: 0.8627
- val_loss: 0.0264 - val_accuracy: 0.8719
```

```
Epoch 216/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0272 - accuracy: 0.8630
- val_loss: 0.0263 - val_accuracy: 0.8720
Epoch 217/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0271 - accuracy: 0.8632
- val_loss: 0.0262 - val_accuracy: 0.8721
Epoch 218/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0271 - accuracy: 0.8637
- val_loss: 0.0261 - val_accuracy: 0.8724
Epoch 219/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0270 - accuracy: 0.8640
- val_loss: 0.0260 - val_accuracy: 0.8729
Epoch 220/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0269 - accuracy: 0.8641
- val_loss: 0.0260 - val_accuracy: 0.8729
Epoch 221/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0268 - accuracy: 0.8645
- val_loss: 0.0259 - val_accuracy: 0.8729
Epoch 222/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0267 - accuracy: 0.8648
- val_loss: 0.0258 - val_accuracy: 0.8731
Epoch 223/500
469/469 [==============================] - 1s 3ms/step - loss: 0.0266 - accuracy: 0.8651
- val_loss: 0.0257 - val_accuracy: 0.8735
Epoch 224/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0265 - accuracy: 0.8653
- val_loss: 0.0256 - val_accuracy: 0.8737
Epoch 225/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0264 - accuracy: 0.8657
- val_loss: 0.0255 - val_accuracy: 0.8738
Epoch 226/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0263 - accuracy: 0.8659
- val_loss: 0.0254 - val_accuracy: 0.8739
Epoch 227/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0263 - accuracy: 0.8663
- val_loss: 0.0253 - val_accuracy: 0.8741
Epoch 228/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0262 - accuracy: 0.8665
- val_loss: 0.0253 - val_accuracy: 0.8741
Epoch 229/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0261 - accuracy: 0.8668
- val_loss: 0.0252 - val_accuracy: 0.8741
Epoch 230/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0260 - accuracy: 0.8670
- val_loss: 0.0251 - val_accuracy: 0.8743
Epoch 231/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0259 - accuracy: 0.8672
- val_loss: 0.0250 - val_accuracy: 0.8744
Epoch 232/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0258 - accuracy: 0.8675
- val_loss: 0.0249 - val_accuracy: 0.8749
Epoch 233/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0258 - accuracy: 0.8678
- val_loss: 0.0248 - val_accuracy: 0.8751
Epoch 234/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0257 - accuracy: 0.8679
- val_loss: 0.0248 - val_accuracy: 0.8755
Epoch 235/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0256 - accuracy: 0.8680
- val_loss: 0.0247 - val_accuracy: 0.8755
```

```
Epoch 236/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0255 - accuracy: 0.8685
- val_loss: 0.0246 - val_accuracy: 0.8757
Epoch 237/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0255 - accuracy: 0.8686
- val_loss: 0.0245 - val_accuracy: 0.8760
Epoch 238/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0254 - accuracy: 0.8691
- val_loss: 0.0245 - val_accuracy: 0.8761
Epoch 239/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0253 - accuracy: 0.8692
- val_loss: 0.0244 - val_accuracy: 0.8764
Epoch 240/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0252 - accuracy: 0.8694
- val_loss: 0.0243 - val_accuracy: 0.8767
Epoch 241/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0252 - accuracy: 0.8696
- val_loss: 0.0242 - val_accuracy: 0.8770
Epoch 242/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0251 - accuracy: 0.8696
- val_loss: 0.0242 - val_accuracy: 0.8774
Epoch 243/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0250 - accuracy: 0.8698
- val_loss: 0.0241 - val_accuracy: 0.8774
Epoch 244/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0249 - accuracy: 0.8702
- val_loss: 0.0240 - val_accuracy: 0.8777
Epoch 245/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0249 - accuracy: 0.8704
- val_loss: 0.0239 - val_accuracy: 0.8779
Epoch 246/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0248 - accuracy: 0.8706
- val_loss: 0.0239 - val_accuracy: 0.8781
Epoch 247/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0247 - accuracy: 0.8708
- val_loss: 0.0238 - val_accuracy: 0.8782
Epoch 248/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0247 - accuracy: 0.8710
- val_loss: 0.0237 - val_accuracy: 0.8782
Epoch 249/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0246 - accuracy: 0.8712
- val_loss: 0.0237 - val_accuracy: 0.8785
Epoch 250/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0245 - accuracy: 0.8714
- val_loss: 0.0236 - val_accuracy: 0.8784
Epoch 251/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0245 - accuracy: 0.8717
- val_loss: 0.0235 - val_accuracy: 0.8787
Epoch 252/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0244 - accuracy: 0.8719
- val_loss: 0.0235 - val_accuracy: 0.8789
Epoch 253/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0243 - accuracy: 0.8722
- val_loss: 0.0234 - val_accuracy: 0.8794
Epoch 254/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0243 - accuracy: 0.8724
- val_loss: 0.0233 - val_accuracy: 0.8795
Epoch 255/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0242 - accuracy: 0.8725
- val_loss: 0.0233 - val_accuracy: 0.8795
```

```
Epoch 256/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0241 - accuracy: 0.8726
- val_loss: 0.0232 - val_accuracy: 0.8796
Epoch 257/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0241 - accuracy: 0.8729
- val_loss: 0.0231 - val_accuracy: 0.8799
Epoch 258/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0240 - accuracy: 0.8730
- val_loss: 0.0231 - val_accuracy: 0.8800
Epoch 259/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0239 - accuracy: 0.8735
- val_loss: 0.0230 - val_accuracy: 0.8803
Epoch 260/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0239 - accuracy: 0.8737
- val_loss: 0.0230 - val_accuracy: 0.8805
Epoch 261/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0238 - accuracy: 0.8740
- val_loss: 0.0229 - val_accuracy: 0.8809
Epoch 262/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0238 - accuracy: 0.8741
- val_loss: 0.0228 - val_accuracy: 0.8813
Epoch 263/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0237 - accuracy: 0.8743
- val_loss: 0.0228 - val_accuracy: 0.8816
Epoch 264/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0236 - accuracy: 0.8744
- val_loss: 0.0227 - val_accuracy: 0.8817
Epoch 265/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0236 - accuracy: 0.8744
- val_loss: 0.0227 - val_accuracy: 0.8822
Epoch 266/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0235 - accuracy: 0.8747
- val_loss: 0.0226 - val_accuracy: 0.8824
Epoch 267/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0235 - accuracy: 0.8750
- val_loss: 0.0225 - val_accuracy: 0.8825
Epoch 268/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0234 - accuracy: 0.8754
- val_loss: 0.0225 - val_accuracy: 0.8828
Epoch 269/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0234 - accuracy: 0.8756
- val_loss: 0.0224 - val_accuracy: 0.8829
Epoch 270/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0233 - accuracy: 0.8759
- val_loss: 0.0224 - val_accuracy: 0.8831
Epoch 271/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0232 - accuracy: 0.8762
- val_loss: 0.0223 - val_accuracy: 0.8832
Epoch 272/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0232 - accuracy: 0.8763
- val_loss: 0.0223 - val_accuracy: 0.8834
Epoch 273/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0231 - accuracy: 0.8765
- val_loss: 0.0222 - val_accuracy: 0.8838
Epoch 274/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0231 - accuracy: 0.8766
- val_loss: 0.0221 - val_accuracy: 0.8839
Epoch 275/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0230 - accuracy: 0.8768
- val_loss: 0.0221 - val_accuracy: 0.8838
```

```
Epoch 276/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0230 - accuracy: 0.8770
- val_loss: 0.0220 - val_accuracy: 0.8840
Epoch 277/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0229 - accuracy: 0.8771
- val_loss: 0.0220 - val_accuracy: 0.8839
Epoch 278/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0229 - accuracy: 0.8773
- val_loss: 0.0219 - val_accuracy: 0.8840
Epoch 279/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0228 - accuracy: 0.8775
- val_loss: 0.0219 - val_accuracy: 0.8839
Epoch 280/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0228 - accuracy: 0.8777
- val_loss: 0.0218 - val_accuracy: 0.8842
Epoch 281/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0227 - accuracy: 0.8779
- val_loss: 0.0218 - val_accuracy: 0.8842
Epoch 282/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0227 - accuracy: 0.8781
- val_loss: 0.0217 - val_accuracy: 0.8846
Epoch 283/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0226 - accuracy: 0.8784
- val_loss: 0.0217 - val_accuracy: 0.8848
Epoch 284/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0226 - accuracy: 0.8785
- val_loss: 0.0216 - val_accuracy: 0.8848
Epoch 285/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0225 - accuracy: 0.8787
- val_loss: 0.0216 - val_accuracy: 0.8852
Epoch 286/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0225 - accuracy: 0.8788
- val_loss: 0.0215 - val_accuracy: 0.8854
Epoch 287/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0224 - accuracy: 0.8789
- val_loss: 0.0215 - val_accuracy: 0.8855
Epoch 288/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0224 - accuracy: 0.8790
- val_loss: 0.0214 - val_accuracy: 0.8858
Epoch 289/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0223 - accuracy: 0.8792
- val_loss: 0.0214 - val_accuracy: 0.8859
Epoch 290/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0223 - accuracy: 0.8794
- val_loss: 0.0213 - val_accuracy: 0.8860
Epoch 291/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0222 - accuracy: 0.8796
- val_loss: 0.0213 - val_accuracy: 0.8863
Epoch 292/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0222 - accuracy: 0.8799
- val_loss: 0.0213 - val_accuracy: 0.8867
Epoch 293/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0221 - accuracy: 0.8800
- val_loss: 0.0212 - val_accuracy: 0.8870
Epoch 294/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0221 - accuracy: 0.8801
- val_loss: 0.0212 - val_accuracy: 0.8871
Epoch 295/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0220 - accuracy: 0.8804
- val_loss: 0.0211 - val_accuracy: 0.8873
```

```
Epoch 296/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0220 - accuracy: 0.8805
- val_loss: 0.0211 - val_accuracy: 0.8876
Epoch 297/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0220 - accuracy: 0.8806
- val_loss: 0.0210 - val_accuracy: 0.8876
Epoch 298/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0219 - accuracy: 0.8807
- val_loss: 0.0210 - val_accuracy: 0.8878
Epoch 299/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0219 - accuracy: 0.8810
- val_loss: 0.0209 - val_accuracy: 0.8880
Epoch 300/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0218 - accuracy: 0.8811
- val_loss: 0.0209 - val_accuracy: 0.8882
Epoch 301/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0218 - accuracy: 0.8812
- val_loss: 0.0209 - val_accuracy: 0.8883
Epoch 302/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0217 - accuracy: 0.8813
- val_loss: 0.0208 - val_accuracy: 0.8885
Epoch 303/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0217 - accuracy: 0.8813
- val_loss: 0.0208 - val_accuracy: 0.8885
Epoch 304/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0217 - accuracy: 0.8815
- val_loss: 0.0207 - val_accuracy: 0.8890
Epoch 305/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0216 - accuracy: 0.8816
- val_loss: 0.0207 - val_accuracy: 0.8892
Epoch 306/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0216 - accuracy: 0.8817
- val_loss: 0.0206 - val_accuracy: 0.8895
Epoch 307/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0215 - accuracy: 0.8819
- val_loss: 0.0206 - val_accuracy: 0.8895
Epoch 308/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0215 - accuracy: 0.8821
- val_loss: 0.0206 - val_accuracy: 0.8895
Epoch 309/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0214 - accuracy: 0.8823
- val_loss: 0.0205 - val_accuracy: 0.8900
Epoch 310/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0214 - accuracy: 0.8824
- val_loss: 0.0205 - val_accuracy: 0.8901
Epoch 311/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0214 - accuracy: 0.8824
- val_loss: 0.0204 - val_accuracy: 0.8903
Epoch 312/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0213 - accuracy: 0.8826
- val_loss: 0.0204 - val_accuracy: 0.8903
Epoch 313/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0213 - accuracy: 0.8827
- val_loss: 0.0204 - val_accuracy: 0.8906
Epoch 314/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0212 - accuracy: 0.8827
- val_loss: 0.0203 - val_accuracy: 0.8908
Epoch 315/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0212 - accuracy: 0.8828
- val_loss: 0.0203 - val_accuracy: 0.8910
```

```
Epoch 316/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0212 - accuracy: 0.8831
- val_loss: 0.0203 - val_accuracy: 0.8911
Epoch 317/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0211 - accuracy: 0.8831
- val_loss: 0.0202 - val_accuracy: 0.8911
Epoch 318/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0211 - accuracy: 0.8832
- val_loss: 0.0202 - val_accuracy: 0.8913
Epoch 319/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0211 - accuracy: 0.8833
- val_loss: 0.0201 - val_accuracy: 0.8914
Epoch 320/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0210 - accuracy: 0.8834
- val_loss: 0.0201 - val_accuracy: 0.8917
Epoch 321/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0210 - accuracy: 0.8836
- val_loss: 0.0201 - val_accuracy: 0.8917
Epoch 322/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0209 - accuracy: 0.8838
- val_loss: 0.0200 - val_accuracy: 0.8918
Epoch 323/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0209 - accuracy: 0.8838
- val_loss: 0.0200 - val_accuracy: 0.8918
Epoch 324/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0209 - accuracy: 0.8841
- val_loss: 0.0200 - val_accuracy: 0.8918
Epoch 325/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0208 - accuracy: 0.8842
- val_loss: 0.0199 - val_accuracy: 0.8919
Epoch 326/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0208 - accuracy: 0.8842
- val_loss: 0.0199 - val_accuracy: 0.8921
Epoch 327/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0208 - accuracy: 0.8844
- val_loss: 0.0199 - val_accuracy: 0.8923
Epoch 328/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0207 - accuracy: 0.8845
- val_loss: 0.0198 - val_accuracy: 0.8923
Epoch 329/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0207 - accuracy: 0.8847
- val_loss: 0.0198 - val_accuracy: 0.8924
Epoch 330/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0207 - accuracy: 0.8848
- val_loss: 0.0198 - val_accuracy: 0.8925
Epoch 331/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0206 - accuracy: 0.8850
- val_loss: 0.0197 - val_accuracy: 0.8928
Epoch 332/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0206 - accuracy: 0.8850
- val_loss: 0.0197 - val_accuracy: 0.8928
Epoch 333/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0206 - accuracy: 0.8852
- val_loss: 0.0197 - val_accuracy: 0.8928
Epoch 334/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0205 - accuracy: 0.8853
- val_loss: 0.0196 - val_accuracy: 0.8928
Epoch 335/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0205 - accuracy: 0.8855
- val_loss: 0.0196 - val_accuracy: 0.8927
```

```
Epoch 336/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0205 - accuracy: 0.8856
- val_loss: 0.0196 - val_accuracy: 0.8932
Epoch 337/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0204 - accuracy: 0.8857
- val_loss: 0.0195 - val_accuracy: 0.8932
Epoch 338/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0204 - accuracy: 0.8858
- val_loss: 0.0195 - val_accuracy: 0.8933
Epoch 339/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0204 - accuracy: 0.8860
- val_loss: 0.0195 - val_accuracy: 0.8935
Epoch 340/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0203 - accuracy: 0.8860
- val_loss: 0.0194 - val_accuracy: 0.8936
Epoch 341/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0203 - accuracy: 0.8862
- val_loss: 0.0194 - val_accuracy: 0.8939
Epoch 342/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0203 - accuracy: 0.8863
- val_loss: 0.0194 - val_accuracy: 0.8944
Epoch 343/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0202 - accuracy: 0.8864
- val_loss: 0.0193 - val_accuracy: 0.8944
Epoch 344/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0202 - accuracy: 0.8866
- val_loss: 0.0193 - val_accuracy: 0.8946
Epoch 345/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0202 - accuracy: 0.8866
- val_loss: 0.0193 - val_accuracy: 0.8947
Epoch 346/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0201 - accuracy: 0.8867
- val_loss: 0.0192 - val_accuracy: 0.8947
Epoch 347/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0201 - accuracy: 0.8869
- val_loss: 0.0192 - val_accuracy: 0.8948
Epoch 348/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0201 - accuracy: 0.8870
- val_loss: 0.0192 - val_accuracy: 0.8948
Epoch 349/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0201 - accuracy: 0.8872
- val_loss: 0.0192 - val_accuracy: 0.8949
Epoch 350/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0200 - accuracy: 0.8873
- val_loss: 0.0191 - val_accuracy: 0.8950
Epoch 351/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0200 - accuracy: 0.8874
- val_loss: 0.0191 - val_accuracy: 0.8952
Epoch 352/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0200 - accuracy: 0.8875
- val_loss: 0.0191 - val_accuracy: 0.8954
Epoch 353/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0199 - accuracy: 0.8876
- val_loss: 0.0190 - val_accuracy: 0.8957
Epoch 354/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0199 - accuracy: 0.8877
- val_loss: 0.0190 - val_accuracy: 0.8957
Epoch 355/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0199 - accuracy: 0.8878
- val_loss: 0.0190 - val_accuracy: 0.8959
```

```
Epoch 356/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0198 - accuracy: 0.8879
- val_loss: 0.0190 - val_accuracy: 0.8960
Epoch 357/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0198 - accuracy: 0.8881
- val_loss: 0.0189 - val_accuracy: 0.8961
Epoch 358/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0198 - accuracy: 0.8881
- val_loss: 0.0189 - val_accuracy: 0.8962
Epoch 359/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0198 - accuracy: 0.8882
- val_loss: 0.0189 - val_accuracy: 0.8961
Epoch 360/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0197 - accuracy: 0.8882
- val_loss: 0.0188 - val_accuracy: 0.8964
Epoch 361/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0197 - accuracy: 0.8883
- val_loss: 0.0188 - val_accuracy: 0.8964
Epoch 362/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0197 - accuracy: 0.8885
- val_loss: 0.0188 - val_accuracy: 0.8964
Epoch 363/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0197 - accuracy: 0.8886
- val_loss: 0.0188 - val_accuracy: 0.8964
Epoch 364/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0196 - accuracy: 0.8886
- val_loss: 0.0187 - val_accuracy: 0.8965
Epoch 365/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0196 - accuracy: 0.8887
- val_loss: 0.0187 - val_accuracy: 0.8964
Epoch 366/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0196 - accuracy: 0.8889
- val_loss: 0.0187 - val_accuracy: 0.8966
Epoch 367/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0195 - accuracy: 0.8889
- val_loss: 0.0187 - val_accuracy: 0.8967
Epoch 368/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0195 - accuracy: 0.8890
- val_loss: 0.0186 - val_accuracy: 0.8969
Epoch 369/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0195 - accuracy: 0.8891
- val_loss: 0.0186 - val_accuracy: 0.8969
Epoch 370/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0195 - accuracy: 0.8892
- val_loss: 0.0186 - val_accuracy: 0.8970
Epoch 371/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0194 - accuracy: 0.8893
- val_loss: 0.0186 - val_accuracy: 0.8971
Epoch 372/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0194 - accuracy: 0.8895
- val_loss: 0.0185 - val_accuracy: 0.8972
Epoch 373/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0194 - accuracy: 0.8895
- val_loss: 0.0185 - val_accuracy: 0.8972
Epoch 374/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0194 - accuracy: 0.8896
- val_loss: 0.0185 - val_accuracy: 0.8973
Epoch 375/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0193 - accuracy: 0.8896
- val_loss: 0.0185 - val_accuracy: 0.8974
```

```
Epoch 376/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0193 - accuracy: 0.8897
- val_loss: 0.0184 - val_accuracy: 0.8974
Epoch 377/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0193 - accuracy: 0.8898
- val_loss: 0.0184 - val_accuracy: 0.8974
Epoch 378/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0193 - accuracy: 0.8898
- val_loss: 0.0184 - val_accuracy: 0.8973
Epoch 379/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0192 - accuracy: 0.8899
- val_loss: 0.0184 - val_accuracy: 0.8975
Epoch 380/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0192 - accuracy: 0.8900
- val_loss: 0.0183 - val_accuracy: 0.8976
Epoch 381/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0192 - accuracy: 0.8901
- val_loss: 0.0183 - val_accuracy: 0.8979
Epoch 382/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0192 - accuracy: 0.8903
- val_loss: 0.0183 - val_accuracy: 0.8977
Epoch 383/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0191 - accuracy: 0.8902
- val_loss: 0.0183 - val_accuracy: 0.8978
Epoch 384/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0191 - accuracy: 0.8902
- val_loss: 0.0182 - val_accuracy: 0.8979
Epoch 385/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0191 - accuracy: 0.8904
- val_loss: 0.0182 - val_accuracy: 0.8979
Epoch 386/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0191 - accuracy: 0.8904
- val_loss: 0.0182 - val_accuracy: 0.8980
Epoch 387/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0190 - accuracy: 0.8905
- val_loss: 0.0182 - val_accuracy: 0.8982
Epoch 388/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0190 - accuracy: 0.8906
- val_loss: 0.0181 - val_accuracy: 0.8981
Epoch 389/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0190 - accuracy: 0.8907
- val_loss: 0.0181 - val_accuracy: 0.8981
Epoch 390/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0190 - accuracy: 0.8908
- val_loss: 0.0181 - val_accuracy: 0.8982
Epoch 391/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0189 - accuracy: 0.8909
- val_loss: 0.0181 - val_accuracy: 0.8984
Epoch 392/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0189 - accuracy: 0.8910
- val_loss: 0.0181 - val_accuracy: 0.8984
Epoch 393/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0189 - accuracy: 0.8910
- val_loss: 0.0180 - val_accuracy: 0.8985
Epoch 394/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0189 - accuracy: 0.8911
- val_loss: 0.0180 - val_accuracy: 0.8985
Epoch 395/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0189 - accuracy: 0.8913
- val_loss: 0.0180 - val_accuracy: 0.8985
```

```
Epoch 396/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0188 - accuracy: 0.8913
- val_loss: 0.0180 - val_accuracy: 0.8986
Epoch 397/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0188 - accuracy: 0.8914
- val_loss: 0.0179 - val_accuracy: 0.8987
Epoch 398/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0188 - accuracy: 0.8915
- val_loss: 0.0179 - val_accuracy: 0.8988
Epoch 399/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0188 - accuracy: 0.8916
- val_loss: 0.0179 - val_accuracy: 0.8989
Epoch 400/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0187 - accuracy: 0.8918
- val_loss: 0.0179 - val_accuracy: 0.8990
Epoch 401/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0187 - accuracy: 0.8918
- val_loss: 0.0179 - val_accuracy: 0.8990
Epoch 402/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0187 - accuracy: 0.8919
- val_loss: 0.0178 - val_accuracy: 0.8991
Epoch 403/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0187 - accuracy: 0.8920
- val_loss: 0.0178 - val_accuracy: 0.8992
Epoch 404/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0187 - accuracy: 0.8921
- val_loss: 0.0178 - val_accuracy: 0.8990
Epoch 405/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0186 - accuracy: 0.8922
- val_loss: 0.0178 - val_accuracy: 0.8992
Epoch 406/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0186 - accuracy: 0.8923
- val_loss: 0.0178 - val_accuracy: 0.8992
Epoch 407/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0186 - accuracy: 0.8924
- val_loss: 0.0177 - val_accuracy: 0.8992
Epoch 408/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0186 - accuracy: 0.8924
- val_loss: 0.0177 - val_accuracy: 0.8994
Epoch 409/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0186 - accuracy: 0.8925
- val_loss: 0.0177 - val_accuracy: 0.8994
Epoch 410/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0185 - accuracy: 0.8926
- val_loss: 0.0177 - val_accuracy: 0.8996
Epoch 411/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0185 - accuracy: 0.8926
- val_loss: 0.0176 - val_accuracy: 0.8999
Epoch 412/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0185 - accuracy: 0.8927
- val_loss: 0.0176 - val_accuracy: 0.8997
Epoch 413/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0185 - accuracy: 0.8927
- val_loss: 0.0176 - val_accuracy: 0.9000
Epoch 414/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0184 - accuracy: 0.8928
- val_loss: 0.0176 - val_accuracy: 0.9000
Epoch 415/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0184 - accuracy: 0.8928
- val_loss: 0.0176 - val_accuracy: 0.9000
```

```
Epoch 416/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0184 - accuracy: 0.8929
- val_loss: 0.0175 - val_accuracy: 0.9000
Epoch 417/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0184 - accuracy: 0.8930
- val_loss: 0.0175 - val_accuracy: 0.9000
Epoch 418/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0184 - accuracy: 0.8931
- val_loss: 0.0175 - val_accuracy: 0.9000
Epoch 419/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0183 - accuracy: 0.8931
- val_loss: 0.0175 - val_accuracy: 0.9000
Epoch 420/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0183 - accuracy: 0.8932
- val_loss: 0.0175 - val_accuracy: 0.9002
Epoch 421/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0183 - accuracy: 0.8934
- val_loss: 0.0175 - val_accuracy: 0.9003
Epoch 422/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0183 - accuracy: 0.8932
- val_loss: 0.0174 - val_accuracy: 0.9003
Epoch 423/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0183 - accuracy: 0.8934
- val_loss: 0.0174 - val_accuracy: 0.9003
Epoch 424/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0183 - accuracy: 0.8935
- val_loss: 0.0174 - val_accuracy: 0.9004
Epoch 425/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0182 - accuracy: 0.8937
- val_loss: 0.0174 - val_accuracy: 0.9004
Epoch 426/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0182 - accuracy: 0.8937
- val_loss: 0.0174 - val_accuracy: 0.9004
Epoch 427/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0182 - accuracy: 0.8939
- val_loss: 0.0173 - val_accuracy: 0.9006
Epoch 428/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0182 - accuracy: 0.8938
- val_loss: 0.0173 - val_accuracy: 0.9008
Epoch 429/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0182 - accuracy: 0.8941
- val_loss: 0.0173 - val_accuracy: 0.9009
Epoch 430/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0181 - accuracy: 0.8941
- val_loss: 0.0173 - val_accuracy: 0.9009
Epoch 431/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0181 - accuracy: 0.8943
- val_loss: 0.0173 - val_accuracy: 0.9009
Epoch 432/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0181 - accuracy: 0.8943
- val_loss: 0.0173 - val_accuracy: 0.9009
Epoch 433/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0181 - accuracy: 0.8944
- val_loss: 0.0172 - val_accuracy: 0.9008
Epoch 434/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0181 - accuracy: 0.8944
- val_loss: 0.0172 - val_accuracy: 0.9009
Epoch 435/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0180 - accuracy: 0.8946
- val_loss: 0.0172 - val_accuracy: 0.9010
```

```
Epoch 436/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0180 - accuracy: 0.8948
- val_loss: 0.0172 - val_accuracy: 0.9009
Epoch 437/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0180 - accuracy: 0.8948
- val_loss: 0.0172 - val_accuracy: 0.9012
Epoch 438/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0180 - accuracy: 0.8948
- val_loss: 0.0171 - val_accuracy: 0.9012
Epoch 439/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0180 - accuracy: 0.8950
- val_loss: 0.0171 - val_accuracy: 0.9012
Epoch 440/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0180 - accuracy: 0.8950
- val_loss: 0.0171 - val_accuracy: 0.9012
Epoch 441/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0179 - accuracy: 0.8950
- val_loss: 0.0171 - val_accuracy: 0.9013
Epoch 442/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0179 - accuracy: 0.8951
- val_loss: 0.0171 - val_accuracy: 0.9013
Epoch 443/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0179 - accuracy: 0.8952
- val_loss: 0.0171 - val_accuracy: 0.9013
Epoch 444/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0179 - accuracy: 0.8953
- val_loss: 0.0170 - val_accuracy: 0.9013
Epoch 445/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0179 - accuracy: 0.8954
- val_loss: 0.0170 - val_accuracy: 0.9012
Epoch 446/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0178 - accuracy: 0.8954
- val_loss: 0.0170 - val_accuracy: 0.9012
Epoch 447/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0178 - accuracy: 0.8955
- val_loss: 0.0170 - val_accuracy: 0.9013
Epoch 448/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0178 - accuracy: 0.8954
- val_loss: 0.0170 - val_accuracy: 0.9014
Epoch 449/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0178 - accuracy: 0.8956
- val_loss: 0.0170 - val_accuracy: 0.9015
Epoch 450/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0178 - accuracy: 0.8957
- val_loss: 0.0169 - val_accuracy: 0.9016
Epoch 451/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0178 - accuracy: 0.8958
- val_loss: 0.0169 - val_accuracy: 0.9016
Epoch 452/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0177 - accuracy: 0.8959
- val_loss: 0.0169 - val_accuracy: 0.9016
Epoch 453/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0177 - accuracy: 0.8960
- val_loss: 0.0169 - val_accuracy: 0.9016
Epoch 454/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0177 - accuracy: 0.8960
- val_loss: 0.0169 - val_accuracy: 0.9017
Epoch 455/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0177 - accuracy: 0.8961
- val_loss: 0.0169 - val_accuracy: 0.9017
```

```
Epoch 456/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0177 - accuracy: 0.8961
- val_loss: 0.0168 - val_accuracy: 0.9018
Epoch 457/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0177 - accuracy: 0.8961
- val_loss: 0.0168 - val_accuracy: 0.9018
Epoch 458/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0176 - accuracy: 0.8962
- val_loss: 0.0168 - val_accuracy: 0.9019
Epoch 459/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0176 - accuracy: 0.8962
- val_loss: 0.0168 - val_accuracy: 0.9019
Epoch 460/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0176 - accuracy: 0.8963
- val_loss: 0.0168 - val_accuracy: 0.9019
Epoch 461/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0176 - accuracy: 0.8963
- val_loss: 0.0168 - val_accuracy: 0.9019
Epoch 462/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0176 - accuracy: 0.8964
- val_loss: 0.0168 - val_accuracy: 0.9021
Epoch 463/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0176 - accuracy: 0.8965
- val_loss: 0.0167 - val_accuracy: 0.9021
Epoch 464/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0176 - accuracy: 0.8965
- val_loss: 0.0167 - val_accuracy: 0.9023
Epoch 465/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0175 - accuracy: 0.8966
- val_loss: 0.0167 - val_accuracy: 0.9024
Epoch 466/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0175 - accuracy: 0.8967
- val_loss: 0.0167 - val_accuracy: 0.9024
Epoch 467/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0175 - accuracy: 0.8966
- val_loss: 0.0167 - val_accuracy: 0.9024
Epoch 468/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0175 - accuracy: 0.8967
- val_loss: 0.0167 - val_accuracy: 0.9026
Epoch 469/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0175 - accuracy: 0.8968
- val_loss: 0.0166 - val_accuracy: 0.9027
Epoch 470/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0175 - accuracy: 0.8968
- val_loss: 0.0166 - val_accuracy: 0.9026
Epoch 471/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0174 - accuracy: 0.8969
- val_loss: 0.0166 - val_accuracy: 0.9026
Epoch 472/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8969
- val_loss: 0.0166 - val_accuracy: 0.9026
Epoch 473/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8970
- val_loss: 0.0166 - val_accuracy: 0.9026
Epoch 474/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8970
- val_loss: 0.0166 - val_accuracy: 0.9028
Epoch 475/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8971
- val_loss: 0.0166 - val_accuracy: 0.9028
```

```
Epoch 476/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8972
- val_loss: 0.0165 - val_accuracy: 0.9029
Epoch 477/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0174 - accuracy: 0.8972
- val_loss: 0.0165 - val_accuracy: 0.9029
Epoch 478/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0173 - accuracy: 0.8973
- val_loss: 0.0165 - val_accuracy: 0.9028
Epoch 479/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0173 - accuracy: 0.8974
- val_loss: 0.0165 - val_accuracy: 0.9028
Epoch 480/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0173 - accuracy: 0.8974
- val_loss: 0.0165 - val_accuracy: 0.9028
Epoch 481/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0173 - accuracy: 0.8975
- val_loss: 0.0165 - val_accuracy: 0.9028
Epoch 482/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0173 - accuracy: 0.8975
- val_loss: 0.0165 - val_accuracy: 0.9030
Epoch 483/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0173 - accuracy: 0.8975
- val_loss: 0.0164 - val_accuracy: 0.9027
Epoch 484/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0172 - accuracy: 0.8976
- val_loss: 0.0164 - val_accuracy: 0.9029
Epoch 485/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0172 - accuracy: 0.8977
- val_loss: 0.0164 - val_accuracy: 0.9030
Epoch 486/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0172 - accuracy: 0.8978
- val_loss: 0.0164 - val_accuracy: 0.9030
Epoch 487/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0172 - accuracy: 0.8978
- val_loss: 0.0164 - val_accuracy: 0.9030
Epoch 488/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0172 - accuracy: 0.8978
- val_loss: 0.0164 - val_accuracy: 0.9032
Epoch 489/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0172 - accuracy: 0.8979
- val_loss: 0.0164 - val_accuracy: 0.9032
Epoch 490/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0172 - accuracy: 0.8980
- val_loss: 0.0163 - val_accuracy: 0.9034
Epoch 491/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0171 - accuracy: 0.8980
- val_loss: 0.0163 - val_accuracy: 0.9034
Epoch 492/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0171 - accuracy: 0.8980
- val_loss: 0.0163 - val_accuracy: 0.9034
Epoch 493/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0171 - accuracy: 0.8981
- val_loss: 0.0163 - val_accuracy: 0.9034
Epoch 494/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0171 - accuracy: 0.8982
- val_loss: 0.0163 - val_accuracy: 0.9034
Epoch 495/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0171 - accuracy: 0.8982
- val_loss: 0.0163 - val_accuracy: 0.9034
```

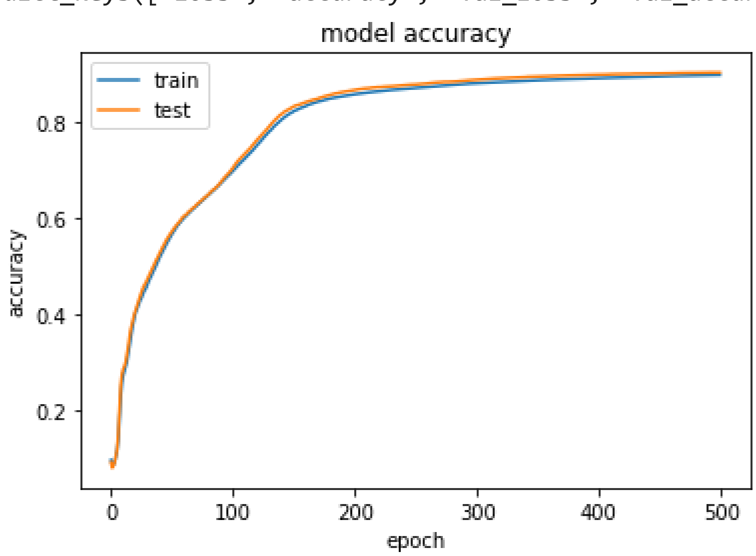
```
Epoch 496/500
469/469 [==============================] - 2s 4ms/step - loss: 0.0171 - accuracy: 0.8982
- val_loss: 0.0163 - val_accuracy: 0.9036
Epoch 497/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0171 - accuracy: 0.8983
- val_loss: 0.0163 - val_accuracy: 0.9035
Epoch 498/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0171 - accuracy: 0.8984
- val_loss: 0.0162 - val_accuracy: 0.9035
Epoch 499/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0170 - accuracy: 0.8984
- val_loss: 0.0162 - val_accuracy: 0.9034
Epoch 500/500
469/469 [==============================] - 2s 3ms/step - loss: 0.0170 - accuracy: 0.8984
- val_loss: 0.0162 - val_accuracy: 0.9034
```

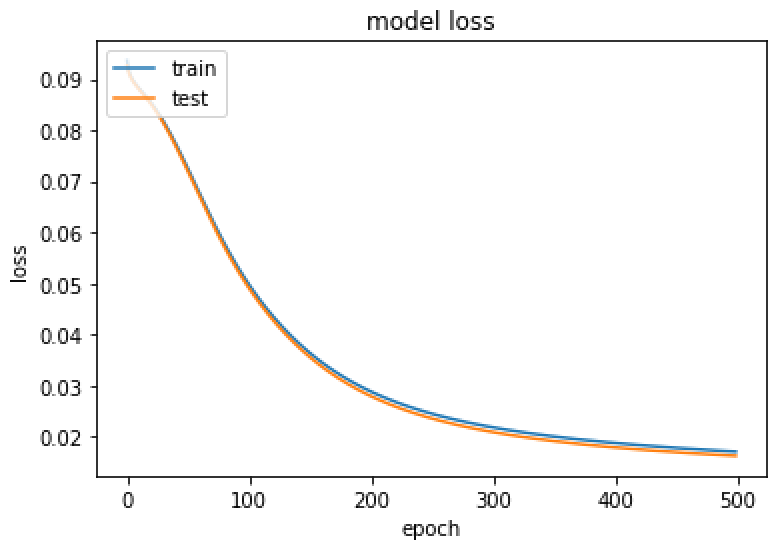
Plot learning curves

In [21]:

```
# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```





Question 3

Does the model show any indication of overfitting? Why (not)?

Overfitting is when the model performs well on the training data but does not perform well on the validation data. In the graphs above, the model is actually shown to be more accurate with the validation data than the training data and the loss for the validation data is also less than the loss for the training data. Additionally, both datasets perform very closely. This indicates that the model does not show any indication of overfitting because the model does not perform better with the training data than with the validation data. This means that the model is not specific to the training data and will perform similarly with data that it has not seen before, which is important if the model will be used to classify outside data.

Evaluate the model

In [22]:
```
model.evaluate(X_validation, y_validation)
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.0162 - accuracy: 0.9034
```

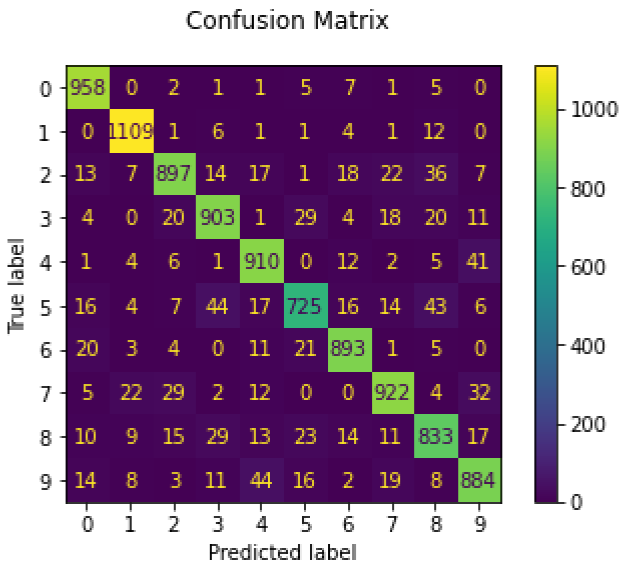
Out[22]:
```
[0.016211673617362976, 0.9034000039100647]
```

Confusion Matrix

The confusion matrix below shows the mistakes that the classifier made on the validation data. The matrix shows that the classifier performed reasonably well, but still made a significant number of mistakes. The multi-colored diagonal represents the data that was correctly classified while the purple squares represent the incorrectly classified data. Additionally, many of these mistakes are quite random, such as predicting a 1 to be an 8. While many digits share common features, a 1 and an 8 do not, showing us that the classifier's performance could be improved and that it makes mistakes on non-ambiguous data.

In [23]:
```
prediction = model.predict(X_validation)
```

```
prediction_classes = np.argmax(prediction, axis=1)

disp = metrics.ConfusionMatrixDisplay.from_predictions(y_valid, prediction_classes)
disp.figure_.suptitle("Confusion Matrix")

plt.show()
```

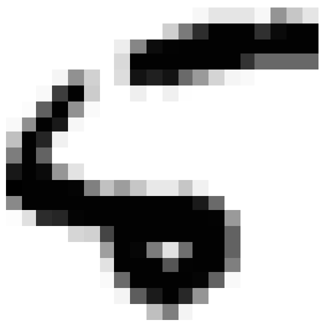


Mistakes

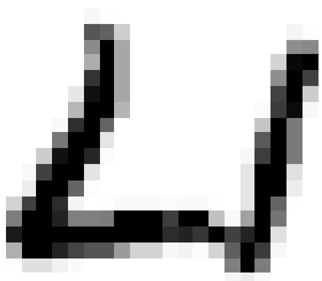
The following ten examples display some of the mistakes that the classifier made. As you can see, some of the samples could be confused with the prediction, but many of them do not share any resemblance between the prediction and the actual class.

In [24]:
```
wrong = []
for i in range(len(prediction_classes)):
  if prediction_classes[i] != y_valid[i]:
    wrong.append(i)

print("Number of mistakes made: " + str(len(wrong)))

for i in range(10):
  print("----------------")
  plt.imshow(X_valid[wrong[i]], cmap='Greys')
  plt.axis('off')
  plt.show()

  print("Actual class: " + str(y_valid[wrong[i]]))
  print("Predicted class: " + str(prediction_classes[wrong[i]]))
print("----------------")
```

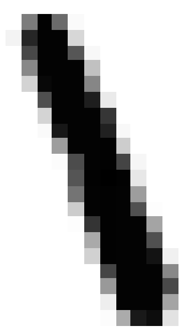
```
Number of mistakes made: 966
----------------
```



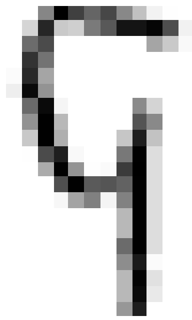
```
Actual class: 5
Predicted class: 6
----------------
```



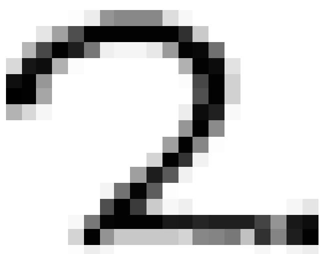
```
Actual class: 4
Predicted class: 6
----------------
```



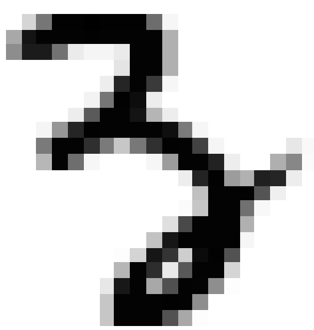
```
Actual class: 1
Predicted class: 3
----------------
```



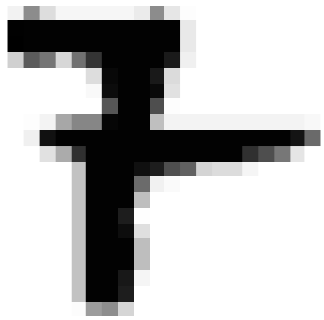
Actual class: 9
Predicted class: 4



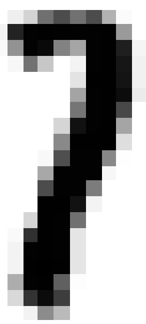
Actual class: 2
Predicted class: 7



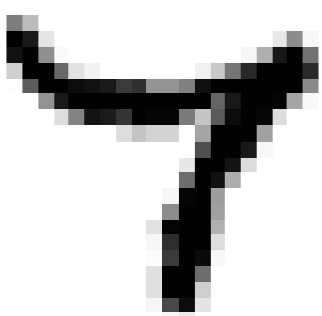
Actual class: 3
Predicted class: 5



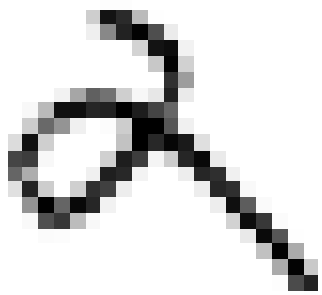
```
Actual class: 7
Predicted class: 1
----------------
```



```
Actual class: 7
Predicted class: 1
----------------
```



```
Actual class: 7
Predicted class: 4
----------------
```



```
Actual class: 2
Predicted class: 9
----------------
```

PART 2 - Convolutional neural network (CNN) architecture

In [25]:

```
model_cnn = keras.Sequential(
    [
        keras.layers.InputLayer(input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model_cnn.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2D  (None, 13, 13, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 flatten (Flatten)           (None, 1600)              0

 dropout (Dropout)           (None, 1600)              0

 dense_2 (Dense)             (None, 10)                16010

=================================================================
```

```
Total params: 34,826
Trainable params: 34,826
Non-trainable params: 0
_________________________________________________________________
```

Configure model

In [26]:

```
model_cnn.compile(
    loss="categorical_crossentropy",
    optimizer="adam",
    metrics=["accuracy"]
)
```

Prepare the data

The CNN does not expect the images to be flattened.

In [27]:

```
# Reload the data, just in case
(X_train, y_train), (X_valid, y_valid) = mnist.load_data()

# convert class vectors to binary class matrices
y_training = keras.utils.to_categorical(y_train, num_classes)
y_validation = keras.utils.to_categorical(y_valid, num_classes)

# Scale images to the [0, 1] range
X_train_cnn = X_train.astype("float32") / 255
X_valid_cnn = X_valid.astype("float32") / 255

# Redefine  dimension of train/test inputs
X_train_cnn = np.expand_dims(X_train_cnn, -1)
X_valid_cnn = np.expand_dims(X_valid_cnn, -1)

# Make sure images have shape (28, 28, 1)
print("x_train shape:", X_train_cnn.shape)
print(X_train_cnn.shape[0], "train samples")
print(X_valid_cnn.shape[0], "test samples")
```

```
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
```

Train!

In [28]:

```
batch_size=128
epochs=15

history = model_cnn.fit(
  X_train_cnn, # training data
  y_training, # training targets
  epochs=epochs,
  batch_size=batch_size,
  verbose=1,
  validation_data=(X_valid_cnn, y_validation)
)
```

```
Epoch 1/15
469/469 [==============================] - 12s 6ms/step - loss: 0.3363 - accuracy: 0.899
0 - val_loss: 0.0794 - val_accuracy: 0.9763
Epoch 2/15
469/469 [==============================] - 3s 5ms/step - loss: 0.1018 - accuracy: 0.9691
- val_loss: 0.0521 - val_accuracy: 0.9846
Epoch 3/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0755 - accuracy: 0.9767
- val_loss: 0.0430 - val_accuracy: 0.9864
Epoch 4/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0639 - accuracy: 0.9807
- val_loss: 0.0381 - val_accuracy: 0.9866
Epoch 5/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0567 - accuracy: 0.9824
- val_loss: 0.0333 - val_accuracy: 0.9886
Epoch 6/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0521 - accuracy: 0.9837
- val_loss: 0.0304 - val_accuracy: 0.9898
Epoch 7/15
469/469 [==============================] - 3s 5ms/step - loss: 0.0465 - accuracy: 0.9850
- val_loss: 0.0280 - val_accuracy: 0.9905
Epoch 8/15
469/469 [==============================] - 3s 5ms/step - loss: 0.0440 - accuracy: 0.9862
- val_loss: 0.0256 - val_accuracy: 0.9912
Epoch 9/15
469/469 [==============================] - 3s 6ms/step - loss: 0.0414 - accuracy: 0.9868
- val_loss: 0.0253 - val_accuracy: 0.9911
Epoch 10/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0381 - accuracy: 0.9883
- val_loss: 0.0237 - val_accuracy: 0.9919
Epoch 11/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0365 - accuracy: 0.9882
- val_loss: 0.0252 - val_accuracy: 0.9910
Epoch 12/15
469/469 [==============================] - 3s 5ms/step - loss: 0.0345 - accuracy: 0.9894
- val_loss: 0.0250 - val_accuracy: 0.9916
Epoch 13/15
469/469 [==============================] - 3s 6ms/step - loss: 0.0343 - accuracy: 0.9892
- val_loss: 0.0239 - val_accuracy: 0.9918
Epoch 14/15
469/469 [==============================] - 3s 5ms/step - loss: 0.0323 - accuracy: 0.9898
- val_loss: 0.0223 - val_accuracy: 0.9925
Epoch 15/15
469/469 [==============================] - 2s 5ms/step - loss: 0.0292 - accuracy: 0.9905
- val_loss: 0.0224 - val_accuracy: 0.9928
```

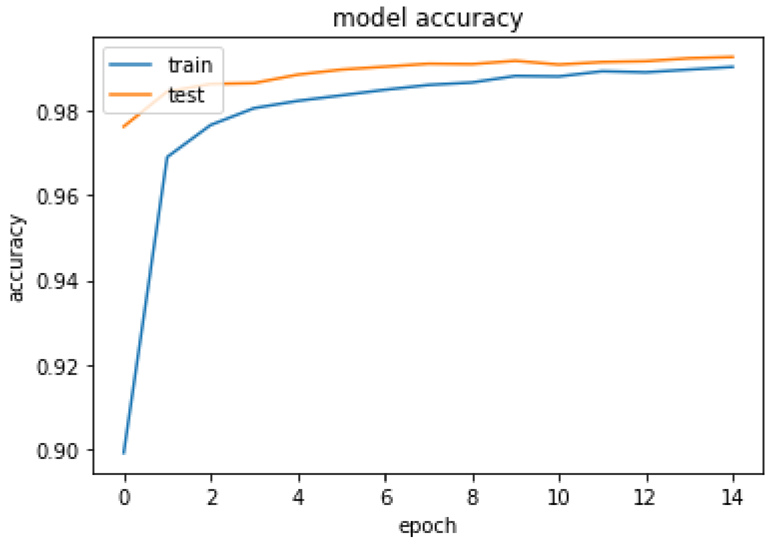
Plot learning curves

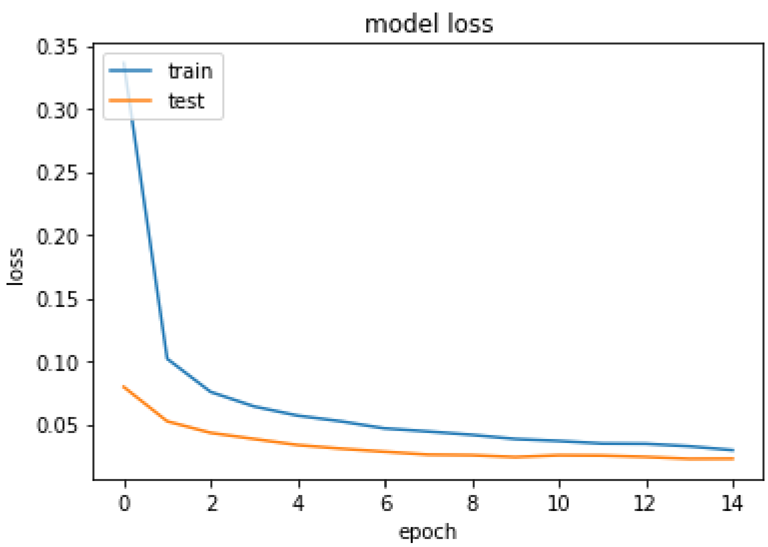
In [29]:

```python
# list all data in history
print(history.history.keys())

# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```python
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```





Evaluate the model

In [30]:

```python
model_cnn.evaluate(X_valid_cnn, y_validation)
```

```
313/313 [==============================] - 1s 3ms/step - loss: 0.0224 - accuracy: 0.9928
```

Out[30]:

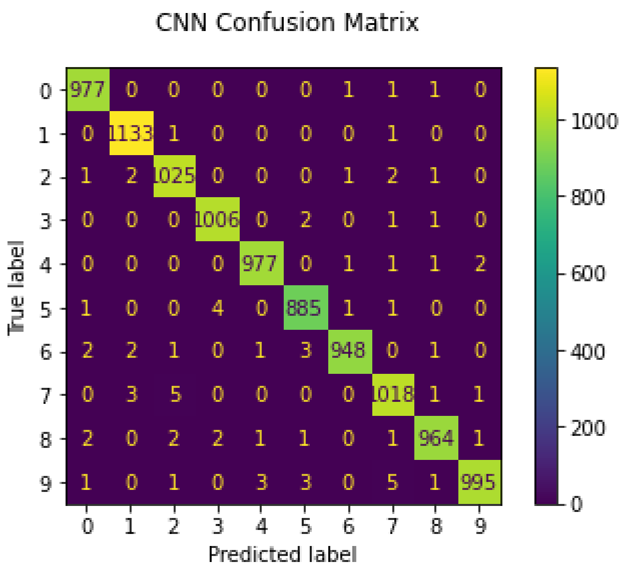
```
[0.022425470873713493, 0.9927999973297119]
```

Confusion Matrix

The following confusion matrix shows the mistakes that the model made. It can clearly be seen that the CNN performed significantly better than the shallow neural network because less data was classified incorrectly as shown by the numbers in the purple squares. Additionally, it can be seen that when the model did make mistakes, it often made mistakes between similar digits such as a 6 and a 5. This suggests that the mistakes that the model makes occur when the data contains ambiguity.

In [31]:

```
cnn_prediction = model_cnn.predict(X_valid_cnn)
cnn_prediction_classes = np.argmax(cnn_prediction, axis=1)

disp = metrics.ConfusionMatrixDisplay.from_predictions(y_valid, cnn_prediction_classes)
disp.figure_.suptitle("CNN Confusion Matrix")

plt.show()
```



Mistakes

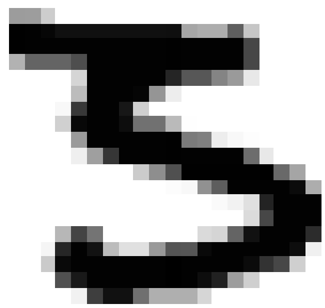
The following ten examples display mistakes that the classifier made. As you can see, many of these digits had elements of ambiguity to them that made it difficult for even a person to distinguish them between two similar digits. However, a few examples were still incorrectly classified despite the true classification being clear.

In [32]:

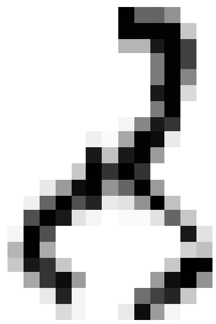
```
wrong = []
for i in range(len(cnn_prediction_classes)):
  if cnn_prediction_classes[i] != y_valid[i]:
    wrong.append(i)

print("Number of mistakes made: " + str(len(wrong)))


for i in range(10):
  print("----------------")
  plt.imshow(X_valid[wrong[i]], cmap='Greys')
  plt.axis('off')
  plt.show()
```

```
    print("Actual class: " + str(y_valid[wrong[i]]))
    print("Predicted class: " + str(cnn_prediction_classes[wrong[i]]))
  print("----------------")
```

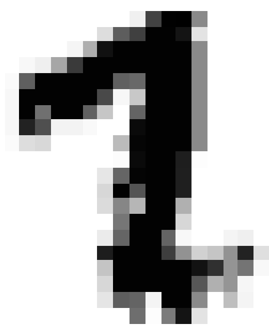
```
Number of mistakes made: 72
----------------
```



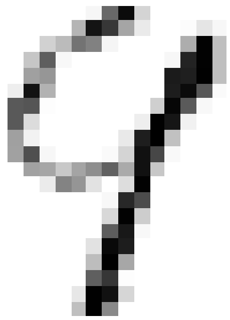
```
Actual class: 3
Predicted class: 5
----------------
```



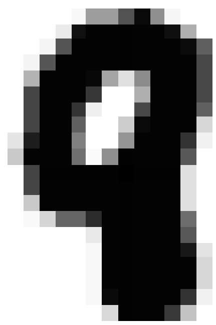
```
Actual class: 8
Predicted class: 2
----------------
```



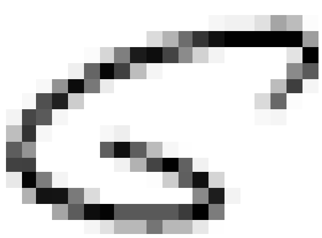
```
Actual class: 2
Predicted class: 1
----------------
```



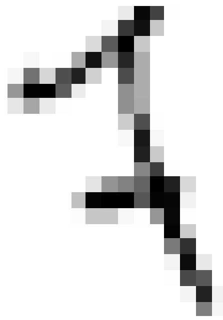
Actual class: 9
Predicted class: 7



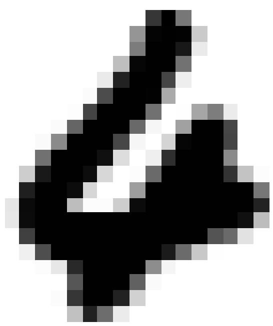
Actual class: 8
Predicted class: 9



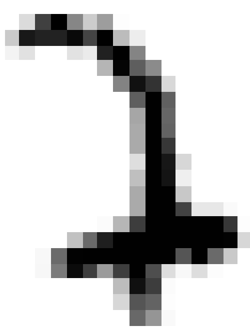
Actual class: 6
Predicted class: 5



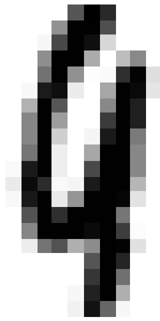
```
Actual class: 7
Predicted class: 1
----------------
```



```
Actual class: 4
Predicted class: 6
----------------
```



```
Actual class: 7
Predicted class: 2
----------------
```



```
Actual class: 9
Predicted class: 4
----------------
```

Question 4

How do the accuracy and loss compare to the previous model? What can you infer from this comparison?

The accuracy is around 9% greater in the CNN than in the original shallow neural network while the loss values are similar. Additionally, the confusion matrices show that the CNN was significantly more accurate and more often made mistakes between similar digits when compared to the original shallow neural network. The loss value tells us how significant the errors that the neural network made were and the accuracy tells us how many data points were incorrectly classified when the probabilities are converted to discrete predictions. The accuracy is a more useful metric to compare the two classifiers because it is a measure of how useful the model is. A highly accurate model is more useful in classifying data than one with low accuracy because more data points will be correctly classified. Based on the comparison, it can be inferred that the CNN is a better model to classify the MNIST dataset because it was significantly more accurate in classifying the data. The CNN’s more complex architecture is better for this classification task than the simpler neural networks architecture. Moreover, the CNN took less time to train and needed fewer epochs than the shallow neural network.

Discussions and Conclusions

Overall, this assignment introduced Keras by creating a classifier for the MNIST dataset. The two classifiers that were created clearly showed that the simple neural network performed significantly worse at correctly classifying the data compared to the CNN. Through the visualization of the results by using a confusion matrix it also became apparent that the CNN made fewer mistakes and the mistakes that it did make were often between similar digits. Furthermore, the shallow neural network made mistakes in a more random way. This was further supported by viewing a few examples of the mistakes that the classifiers made. The CNN's mistakes were made with data that contained greater ambiguity when compared with the shallow neural network's mistakes. Finally, the shallow neural network took a much longer time to train and had more epochs than the CNN despite the CNN's superior performance. This suggests that the architecture of the neural network is more important than the number of training iterations.

This project demonstrated the steps involved in creating a classifier from a dataset. The data and labels needed to be prepared in a way that the classifier could use, then the model's architecture needed to be defined and configured. The model was then trained using the training data. Finally, the progress and results of the model were viewed using learning curves and confusion matrixes. The different parameters and steps that this project involved helped me understand what creating and evaluating a deep learning model entails and what the Keras workflow looks like. I now have a greater understanding of how deep learning models are created and how they work. The next steps to increase my level of understanding and proficiency in using Keras for deep learning would be to explore the different types of neural networks and their applications.