Classify English Handwritten Characters through CNN

In [ ]:

```
import numpy as np
import pandas as pd

import os
for dirname, _, filenames in os.walk('archive'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
archive\english.csv
archive\Img\img001-001.png
archive\Img\img001-002.png
archive\Img\img001-003.png
archive\Img\img001-004.png
archive\Img\img001-005.png
archive\Img\img001-006.png
archive\Img\img001-007.png
archive\Img\img001-008.png
archive\Img\img001-009.png
archive\Img\img001-010.png
archive\Img\img001-011.png
archive\Img\img001-012.png
archive\Img\img001-013.png
archive\Img\img001-014.png
archive\Img\img001-015.png
archive\Img\img001-016.png
archive\Img\img001-017.png
archive\Img\img001-018.png
archive\Img\img001-019.png
archive\Img\img001-020.png
archive\Img\img001-021.png
archive\Img\img001-022.png
archive\Img\img001-023.png
archive\Img\img001-024.png
archive\Img\img001-025.png
archive\Img\img001-026.png
archive\Img\img001-027.png
archive\Img\img001-028.png
archive\Img\img001-029.png
archive\Img\img001-030.png
archive\Img\img001-031.png
archive\Img\img001-032.png
archive\Img\img001-033.png
archive\Img\img001-034.png
archive\Img\img001-035.png
archive\Img\img001-036.png
archive\Img\img001-037.png
archive\Img\img001-038.png
archive\Img\img001-039.png
archive\Img\img001-040.png
archive\Img\img001-041.png
archive\Img\img001-042.png
archive\Img\img001-043.png
archive\Img\img001-044.png
archive\Img\img001-045.png
archive\Img\img001-046.png
archive\Img\img001-047.png
archive\Img\img001-048.png
archive\Img\img001-049.png
archive\Img\img001-050.png
archive\Img\img001-051.png
archive\Img\img001-052.png
archive\Img\img001-053.png
archive\Img\img001-054.png
archive\Img\img001-055.png
archive\Img\img002-001.png
archive\Img\img002-002.png
archive\Img\img002-003.png
archive\Img\img002-004.png
```

```
archive\Img\img062-005.png
archive\Img\img062-006.png
archive\Img\img062-007.png
archive\Img\img062-008.png
archive\Img\img062-009.png
archive\Img\img062-010.png
archive\Img\img062-011.png
archive\Img\img062-012.png
archive\Img\img062-013.png
archive\Img\img062-014.png
archive\Img\img062-015.png
archive\Img\img062-016.png
archive\Img\img062-017.png
archive\Img\img062-018.png
archive\Img\img062-019.png
archive\Img\img062-020.png
archive\Img\img062-021.png
archive\Img\img062-022.png
archive\Img\img062-023.png
archive\Img\img062-024.png
archive\Img\img062-025.png
archive\Img\img062-026.png
archive\Img\img062-027.png
archive\Img\img062-028.png
archive\Img\img062-029.png
archive\Img\img062-030.png
archive\Img\img062-031.png
archive\Img\img062-032.png
archive\Img\img062-033.png
archive\Img\img062-034.png
archive\Img\img062-035.png
archive\Img\img062-036.png
archive\Img\img062-037.png
archive\Img\img062-038.png
archive\Img\img062-039.png
archive\Img\img062-040.png
archive\Img\img062-041.png
archive\Img\img062-042.png
archive\Img\img062-043.png
archive\Img\img062-044.png
archive\Img\img062-045.png
archive\Img\img062-046.png
archive\Img\img062-047.png
archive\Img\img062-048.png
archive\Img\img062-049.png
archive\Img\img062-050.png
archive\Img\img062-051.png
archive\Img\img062-052.png
archive\Img\img062-053.png
archive\Img\img062-054.png
archive\Img\img062-055.png
```

Problem statement

The dataset contains 3410 images containing handwritten letters (0-9 numbers, a-z alphabets small and in caps) The goal is to train the model to recognize and predict the characters efficiently and categorize between 62 unique characters

I'm trying the classification through CNN

import the libraries

```
In [ ]: import pandas
import random
import tensorflow as tf
from keras_preprocessing.image import ImageDataGenerator
import matplotlib.image as img
import matplotlib.pyplot as plt
```

Split the dataset

In this step, we'll split the data into 3 datasets - training set, validation test and test set
Out of total 3410 images, 2910 to training set, 490 added to validation set, 5 to test set
Removed the images added to validation, test set from training set to test its accuracy

```
In [ ]: data_path = r"archive"

dataset = pandas.read_csv(data_path + '/english.csv')
rand = random.sample(range(len(dataset)), 500)
validation_set = pandas.DataFrame(dataset.iloc[rand, :].values, columns=['image'
# remove the added data
dataset.drop(rand, inplace=True)

rand = random.sample(range(len(validation_set)), 12)
test_set = pandas.DataFrame(validation_set.iloc[rand, :].values, columns=['image
# remove the added data
validation_set.drop(rand, inplace=True)

print(test_set)
```

```
                  image label
0   Img/img004-043.png     3
1   Img/img029-039.png     S
2   Img/img023-048.png     M
3   Img/img006-017.png     5
4   Img/img020-027.png     J
5   Img/img020-043.png     J
6   Img/img023-046.png     M
7   Img/img059-028.png     w
8   Img/img060-029.png     x
9   Img/img009-020.png     8
10  Img/img027-050.png     Q
11  Img/img029-053.png     S
```

Data preprocessing

Now that the data is split, lets start with preprocessing step

Load the images through **flow_from_dataframe** method This method is convinient since the data file (english.csv) contains the image names along with the classification class details

In [ ]:
```
train_data_generator = ImageDataGenerator(rescale=1/255, shear_range=0.2, zoom_r
data_generator = ImageDataGenerator(rescale=1/255)
training_data_frame = train_data_generator.flow_from_dataframe(dataframe=dataset
                                                               target_size=(64,
validation_data_frame = data_generator.flow_from_dataframe(dataframe=validation_
                                                           target_size=(64, 64),
test_data_frame = data_generator.flow_from_dataframe(dataframe=test_set, directo
                                                     target_size=(64, 64), class
```

```
Found 2910 validated image filenames belonging to 62 classes.
Found 488 validated image filenames belonging to 62 classes.
Found 12 validated image filenames belonging to 9 classes.
```

Building the CNN model

We are about to build CNN model using libraries provided through **TensorFlow**

Code block breakdown:

- Create Convolution layer: to read/process the image, one feature or one part at a time
- Create Pooling layer: used to reduce the spatial size of convolved image
- Create Flattening layer: used to flatten the result, whose output would be the input for the neural network

We can create multiple convolution and pooling layer depending upon the need/complexity of the dataset

In [ ]:
```
cnn = tf.keras.models.Sequential()

# add convolutional and pooling layer
cnn.add(tf.keras.layers.Conv2D(filters=30, kernel_size=3, activation='relu', inp
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Conv2D(filters=30, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Conv2D(filters=30, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

cnn.add(tf.keras.layers.Flatten())
```

Building, compiling and training the neural network

From the above step we have received the flattened matrix of the images that we processed We are going to feed it to our neural network and train it

In this section, created fully connected Neural network aka Dense network, chosen sigmoid function for activation type In below the model will learn from the training set and predicts the data from validation set

The model accuracy improves as the epochs iteration progresses

In [ ]:
```
# add full connection, output layer
cnn.add(tf.keras.layers.Dense(units=600, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=62, activation='sigmoid'))

# compile cnn
cnn.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accurac
cnn.fit(x=training_data_frame, validation_data=validation_data_frame, epochs=30)
```

```
Epoch 1/30
91/91 [==============================] - 49s 523ms/step - loss: 3.7126 - accurac
y: 0.1058 - val_loss: 2.6056 - val_accuracy: 0.3525
Epoch 2/30
91/91 [==============================] - 35s 382ms/step - loss: 1.8850 - accurac
y: 0.5055 - val_loss: 1.5906 - val_accuracy: 0.5799
Epoch 3/30
91/91 [==============================] - 34s 370ms/step - loss: 1.1412 - accurac
y: 0.6845 - val_loss: 1.3657 - val_accuracy: 0.6639
Epoch 4/30
91/91 [==============================] - 35s 380ms/step - loss: 0.8307 - accurac
y: 0.7491 - val_loss: 1.3678 - val_accuracy: 0.6455
Epoch 5/30
91/91 [==============================] - 34s 376ms/step - loss: 0.6413 - accurac
y: 0.7997 - val_loss: 1.2448 - val_accuracy: 0.6783
Epoch 6/30
91/91 [==============================] - 36s 399ms/step - loss: 0.4814 - accurac
y: 0.8546 - val_loss: 1.2550 - val_accuracy: 0.6885
Epoch 7/30
91/91 [==============================] - 36s 393ms/step - loss: 0.3842 - accurac
y: 0.8777 - val_loss: 1.2418 - val_accuracy: 0.7234
Epoch 8/30
91/91 [==============================] - 37s 407ms/step - loss: 0.2803 - accurac
y: 0.9117 - val_loss: 1.2328 - val_accuracy: 0.7070
Epoch 9/30
91/91 [==============================] - 36s 399ms/step - loss: 0.2674 - accurac
y: 0.9137 - val_loss: 1.3054 - val_accuracy: 0.7090
Epoch 10/30
91/91 [==============================] - 34s 378ms/step - loss: 0.2088 - accurac
y: 0.9357 - val_loss: 1.2232 - val_accuracy: 0.7418
Epoch 11/30
91/91 [==============================] - 36s 396ms/step - loss: 0.2107 - accurac
y: 0.9344 - val_loss: 1.3556 - val_accuracy: 0.7070
Epoch 12/30
91/91 [==============================] - 39s 435ms/step - loss: 0.1824 - accurac
y: 0.9419 - val_loss: 1.4141 - val_accuracy: 0.6967
Epoch 13/30
91/91 [==============================] - 46s 502ms/step - loss: 0.1518 - accurac
y: 0.9522 - val_loss: 1.5065 - val_accuracy: 0.7275
Epoch 14/30
91/91 [==============================] - 52s 573ms/step - loss: 0.1732 - accurac
y: 0.9385 - val_loss: 1.2945 - val_accuracy: 0.7131
Epoch 15/30
91/91 [==============================] - 63s 692ms/step - loss: 0.1509 - accurac
y: 0.9519 - val_loss: 1.3216 - val_accuracy: 0.7172
Epoch 16/30
91/91 [==============================] - 52s 570ms/step - loss: 0.1179 - accurac
y: 0.9605 - val_loss: 1.6675 - val_accuracy: 0.7029
Epoch 17/30
91/91 [==============================] - 38s 415ms/step - loss: 0.1285 - accurac
y: 0.9581 - val_loss: 1.6104 - val_accuracy: 0.7070
Epoch 18/30
91/91 [==============================] - 43s 471ms/step - loss: 0.1058 - accurac
y: 0.9653 - val_loss: 1.5495 - val_accuracy: 0.7295
Epoch 19/30
91/91 [==============================] - 43s 479ms/step - loss: 0.1103 - accurac
y: 0.9643 - val_loss: 1.6781 - val_accuracy: 0.7193
Epoch 20/30
91/91 [==============================] - 34s 371ms/step - loss: 0.1255 - accurac
y: 0.9625 - val_loss: 1.4071 - val_accuracy: 0.7336
```

```
Epoch 21/30
91/91 [==============================] - 43s 471ms/step - loss: 0.1093 - accurac
y: 0.9656 - val_loss: 1.7305 - val_accuracy: 0.7254
Epoch 22/30
91/91 [==============================] - 50s 556ms/step - loss: 0.0947 - accurac
y: 0.9722 - val_loss: 1.5441 - val_accuracy: 0.7439
Epoch 23/30
91/91 [==============================] - 42s 465ms/step - loss: 0.1103 - accurac
y: 0.9649 - val_loss: 1.7748 - val_accuracy: 0.7172
Epoch 24/30
91/91 [==============================] - 47s 514ms/step - loss: 0.1093 - accurac
y: 0.9663 - val_loss: 1.7726 - val_accuracy: 0.7254
Epoch 25/30
91/91 [==============================] - 50s 548ms/step - loss: 0.0823 - accurac
y: 0.9711 - val_loss: 1.7765 - val_accuracy: 0.6988
Epoch 26/30
91/91 [==============================] - 48s 528ms/step - loss: 0.0872 - accurac
y: 0.9759 - val_loss: 1.5370 - val_accuracy: 0.7377
Epoch 27/30
91/91 [==============================] - 46s 506ms/step - loss: 0.0770 - accurac
y: 0.9766 - val_loss: 1.5058 - val_accuracy: 0.7459
Epoch 28/30
91/91 [==============================] - 38s 422ms/step - loss: 0.0713 - accurac
y: 0.9753 - val_loss: 1.7650 - val_accuracy: 0.7172
Epoch 29/30
91/91 [==============================] - 44s 482ms/step - loss: 0.0555 - accurac
y: 0.9804 - val_loss: 1.6092 - val_accuracy: 0.7254
Epoch 30/30
91/91 [==============================] - 46s 512ms/step - loss: 0.0646 - accurac
y: 0.9804 - val_loss: 1.7664 - val_accuracy: 0.7131
```

Out[ ]:

```
<keras.src.callbacks.History at 0x2cac9a80150>
```

Predicting the testset images

Since the model is trained, lets pass the testset images and see how well our model predicts class_indices function gives us the neural network mapping for our 62 characters

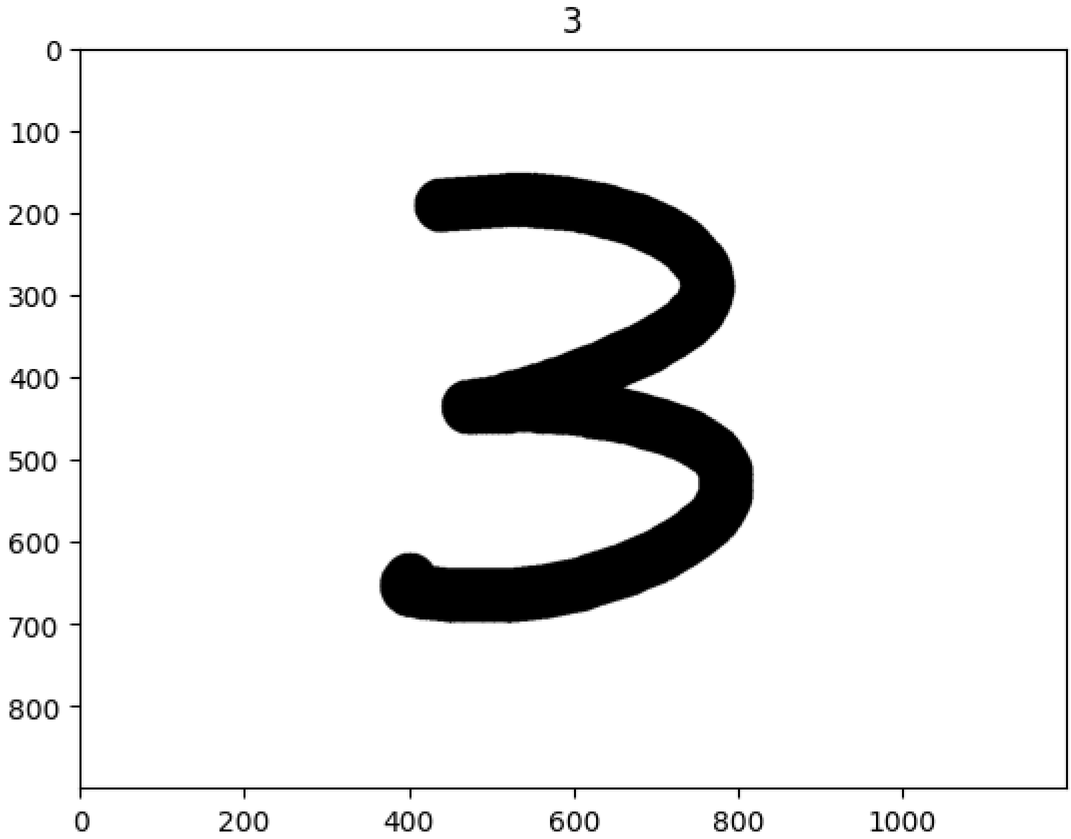
The result image's name is the predicted character by our model

In [ ]:

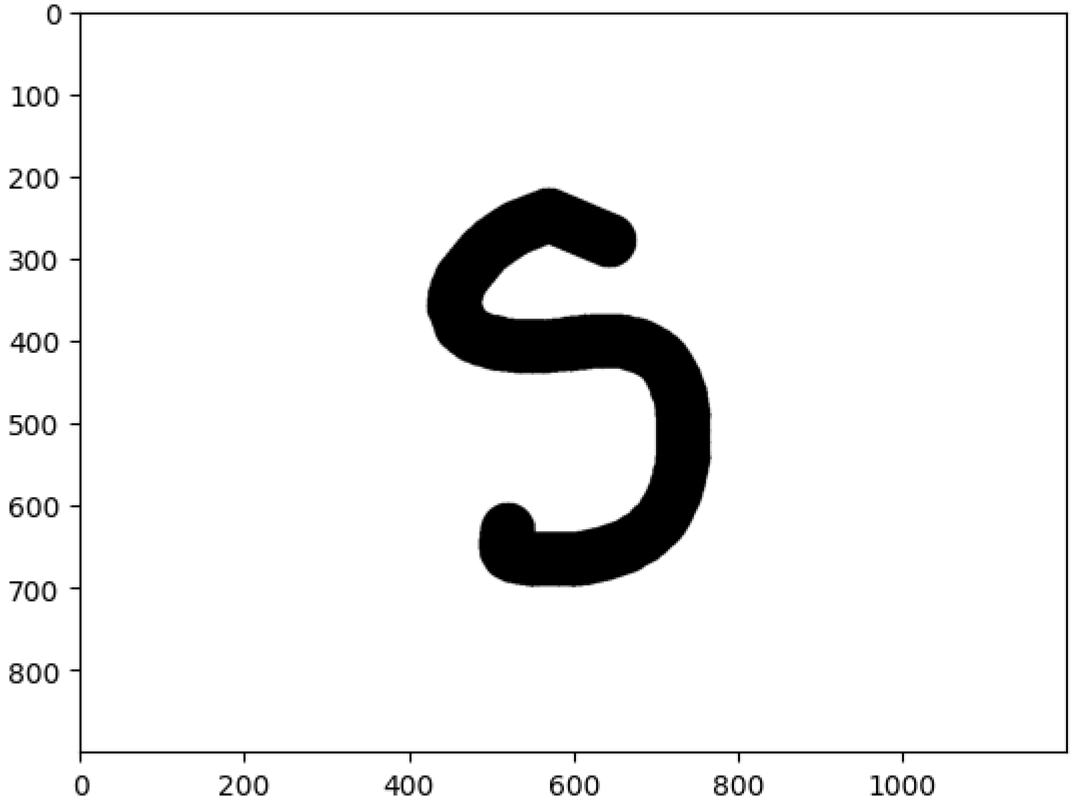
```
print("Prediction mapping: ", training_data_frame.class_indices)
pred = cnn.predict(test_data_frame)

# switcher shows our network mapping to the prediction
switcher = {
            0: "0", 1: "1", 2: "2", 3: "3", 4: "4", 5: "5", 6: "6", 7: "7", 8: "
            11: "B", 12: "C", 13: "D", 14: "E", 15: "F", 16: "G", 17: "H", 18: "
            21: "L", 22: "M", 23: "N", 24: "O", 25: "P", 26: "Q", 27: "R", 28: "
            31: "V", 32: "W", 33: "X", 34: "Y", 35: "Z", 36: "a", 37: "b", 38: "
            41: "f", 42: "g", 43: "h", 44: "i", 45: "j", 46: "k", 47: "l", 48: "
            51: "p", 52: "q", 53: "r", 54: "s", 55: "t", 56: "u", 57: "v", 58: "
            61: "z"}

outputDf = pandas.DataFrame(pred)
maxIndex = list(outputDf.idxmax(axis=1))
print("Max index: ", maxIndex)
for i in range(len(test_set)):
```

```python
    image = img.imread(data_path + '/' + test_set.at[i, 'image'])
    plt.title(switcher.get(maxIndex[i], "error"))
    plt.imshow(image)
    plt.show()
```

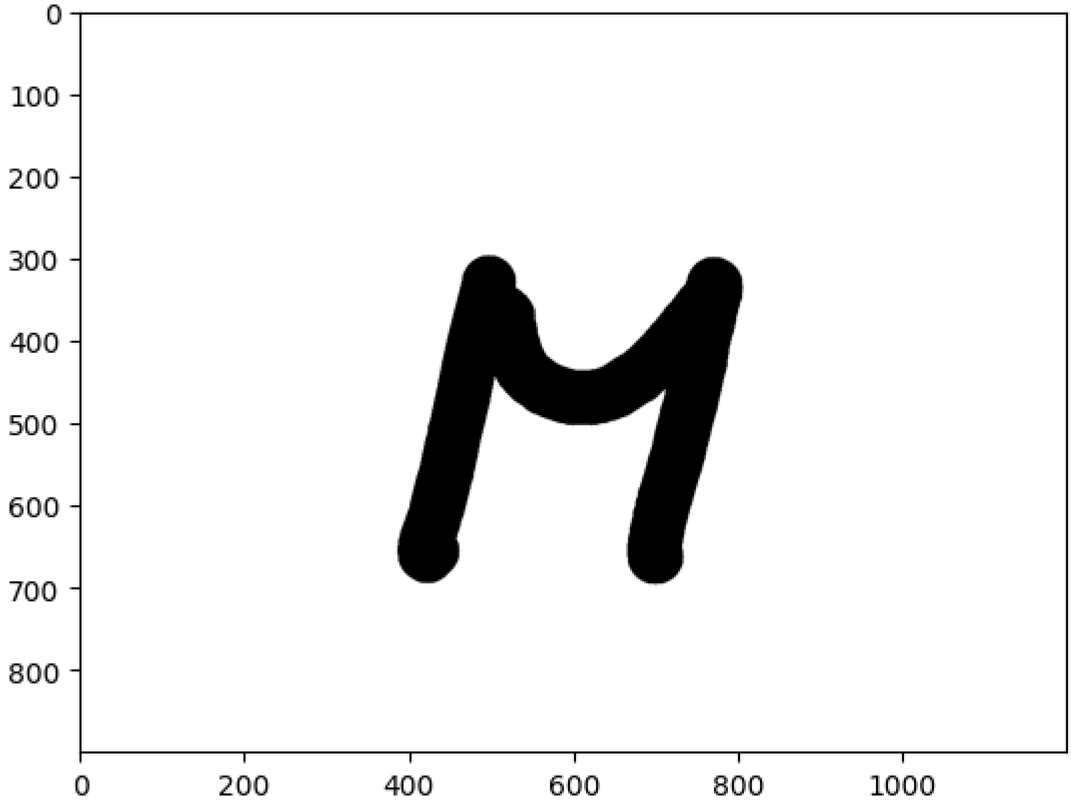
```
Prediction mapping:  {'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6,
'7': 7, '8': 8, '9': 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15,
'G': 16, 'H': 17, 'I': 18, 'J': 19, 'K': 20, 'L': 21, 'M': 22, 'N': 23, 'O': 24,
'P': 25, 'Q': 26, 'R': 27, 'S': 28, 'T': 29, 'U': 30, 'V': 31, 'W': 32, 'X': 33,
'Y': 34, 'Z': 35, 'a': 36, 'b': 37, 'c': 38, 'd': 39, 'e': 40, 'f': 41, 'g': 42,
'h': 43, 'i': 44, 'j': 45, 'k': 46, 'l': 47, 'm': 48, 'n': 49, 'o': 50, 'p': 51,
'q': 52, 'r': 53, 's': 54, 't': 55, 'u': 56, 'v': 57, 'w': 58, 'x': 59, 'y': 60,
'z': 61}
1/1 [==============================] - 0s 246ms/step
Max index:  [3, 54, 22, 3, 19, 19, 22, 58, 59, 8, 26, 28]
```



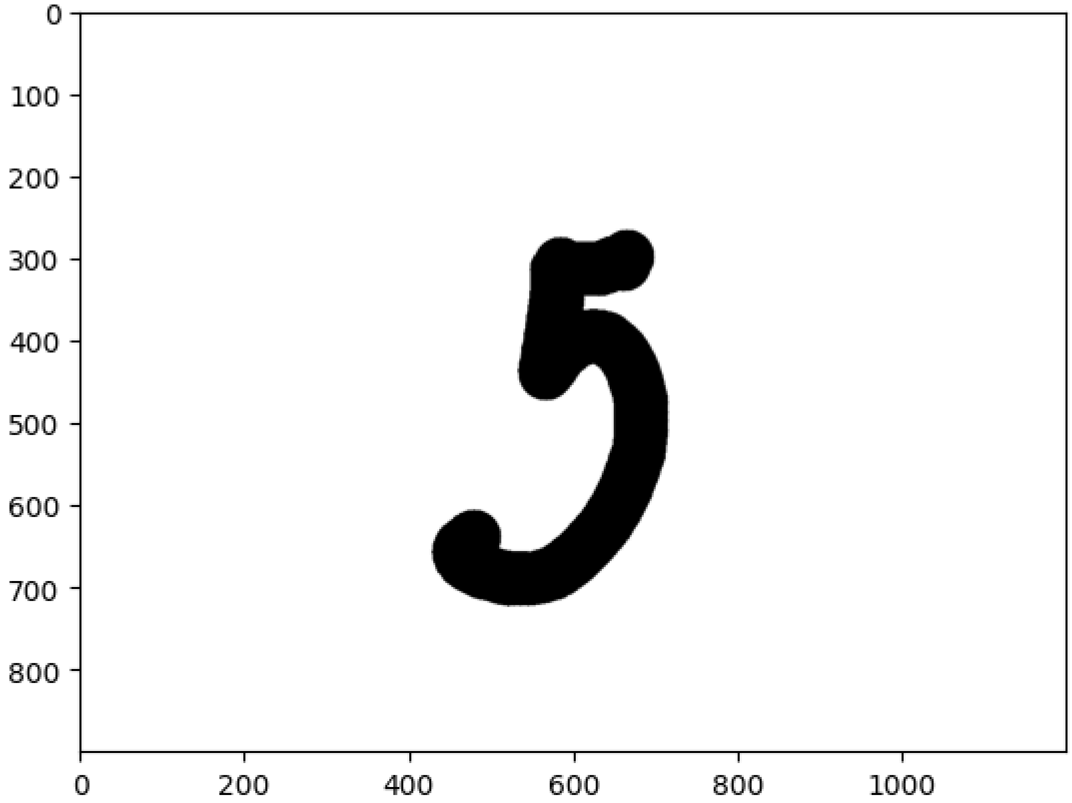
s


0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

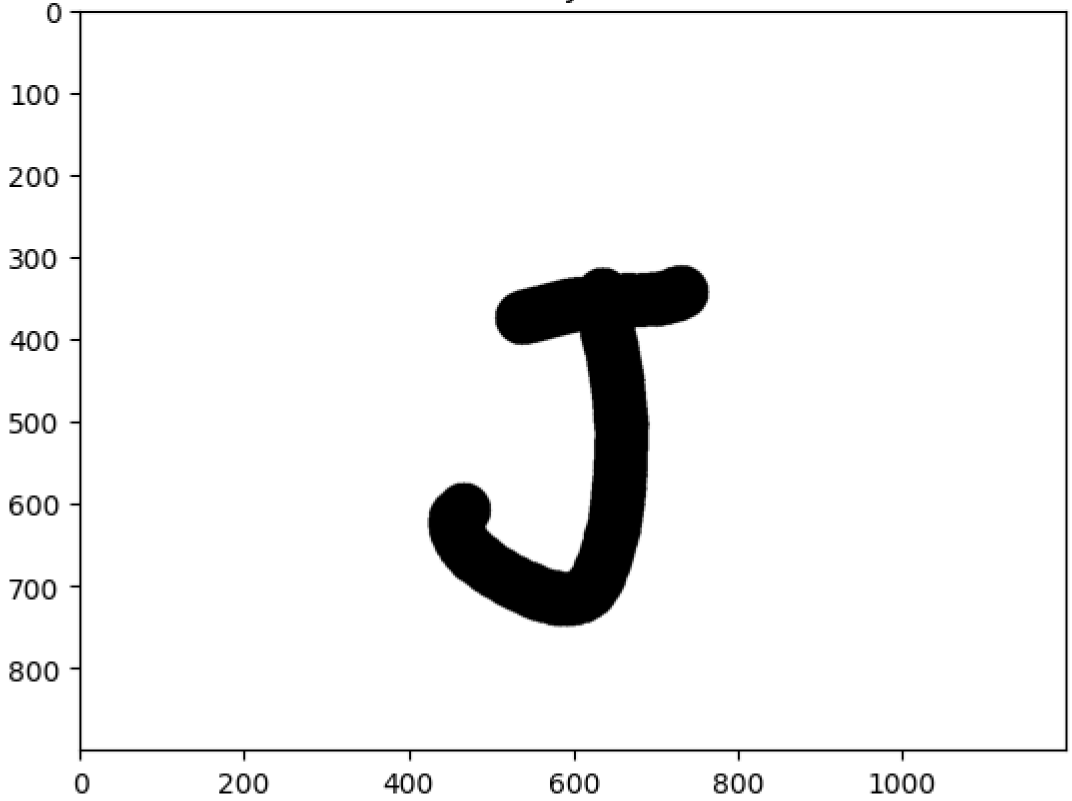
M


0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

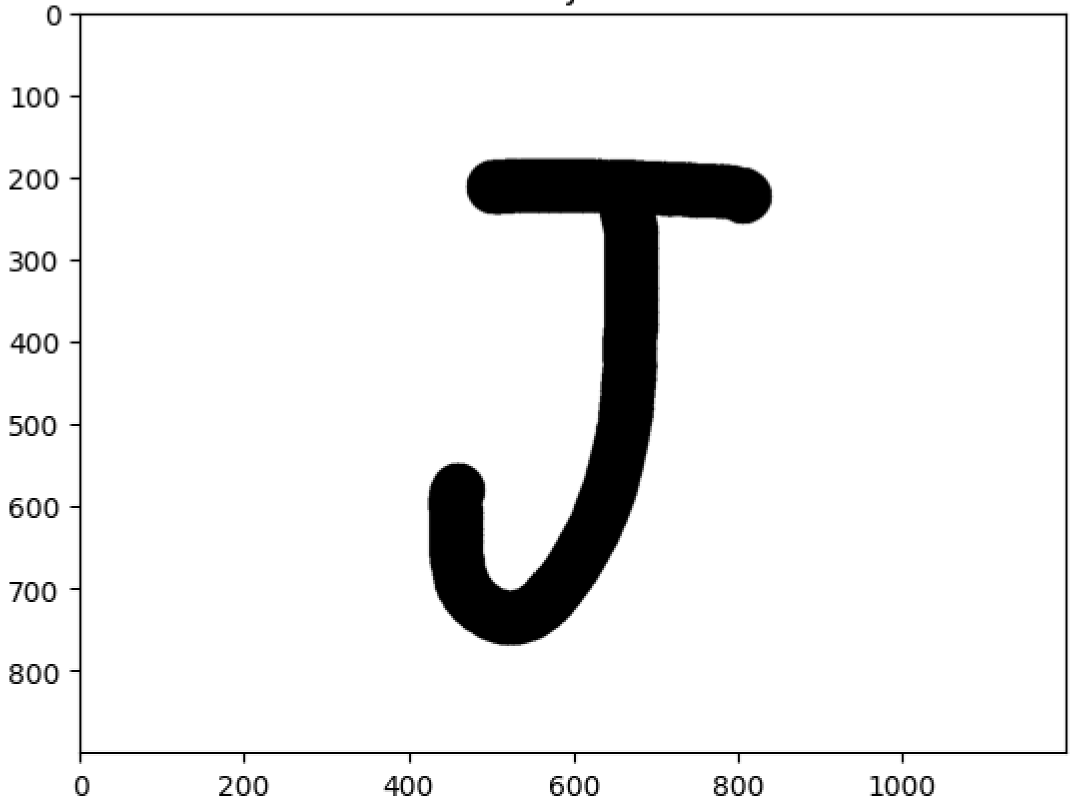
3


0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

J

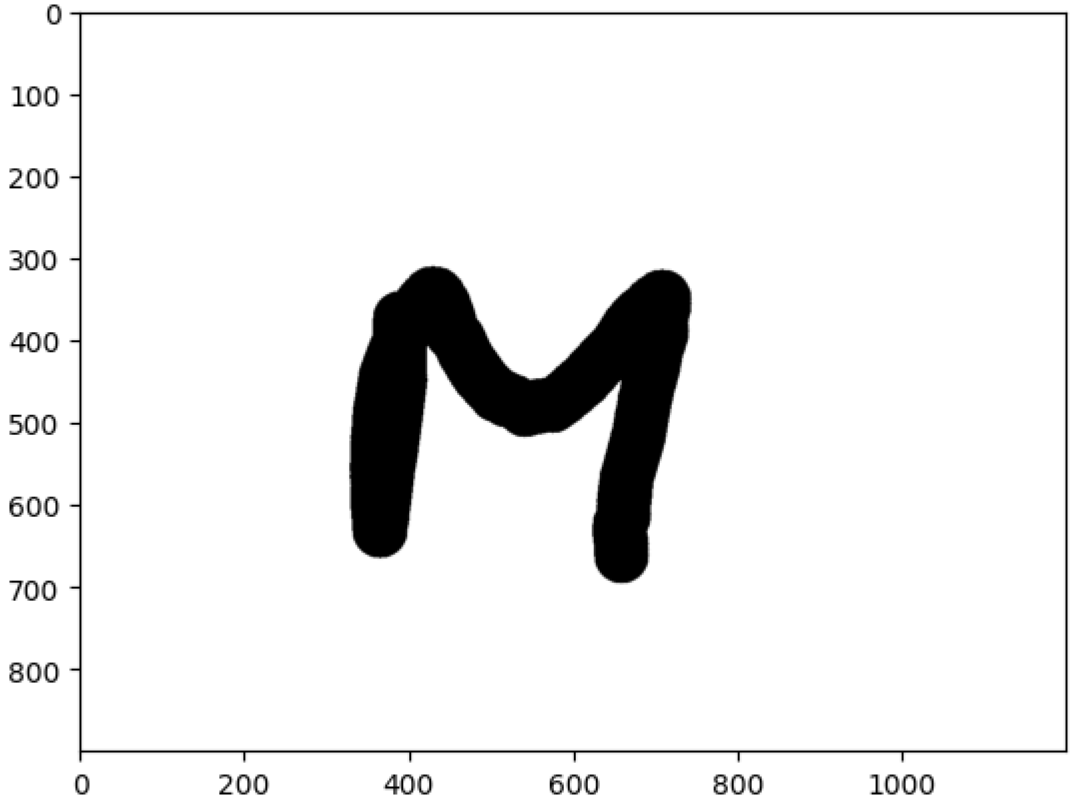

0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

J



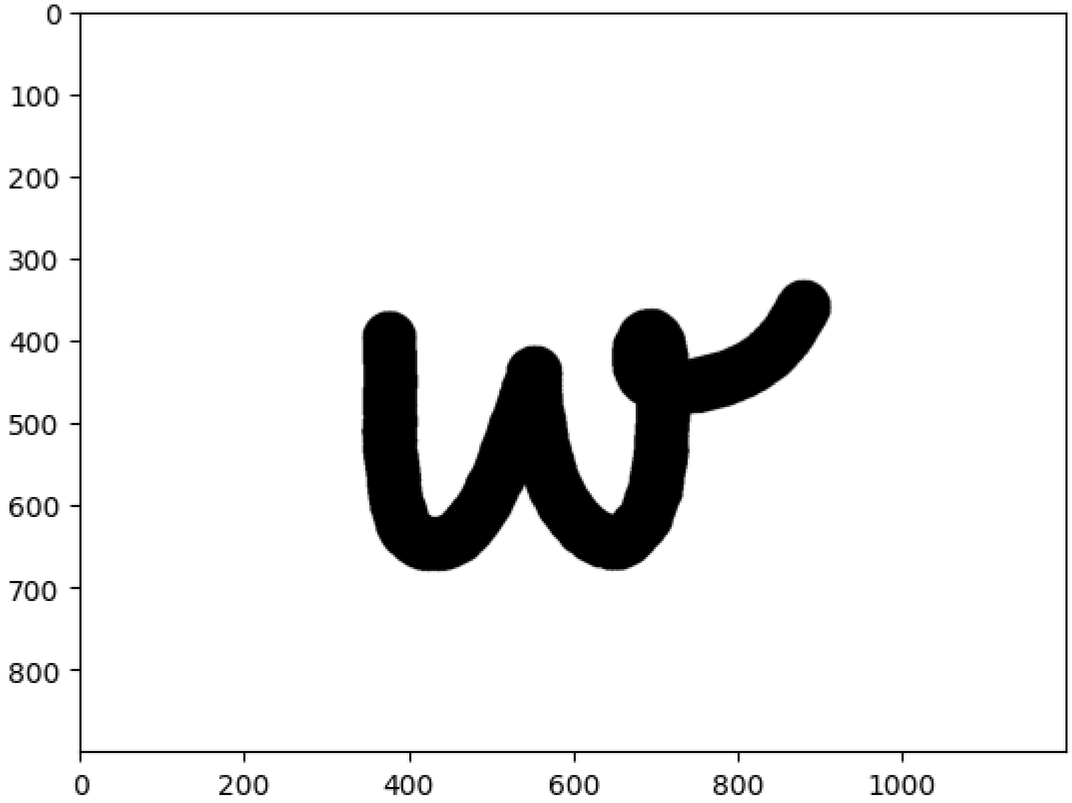
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

M

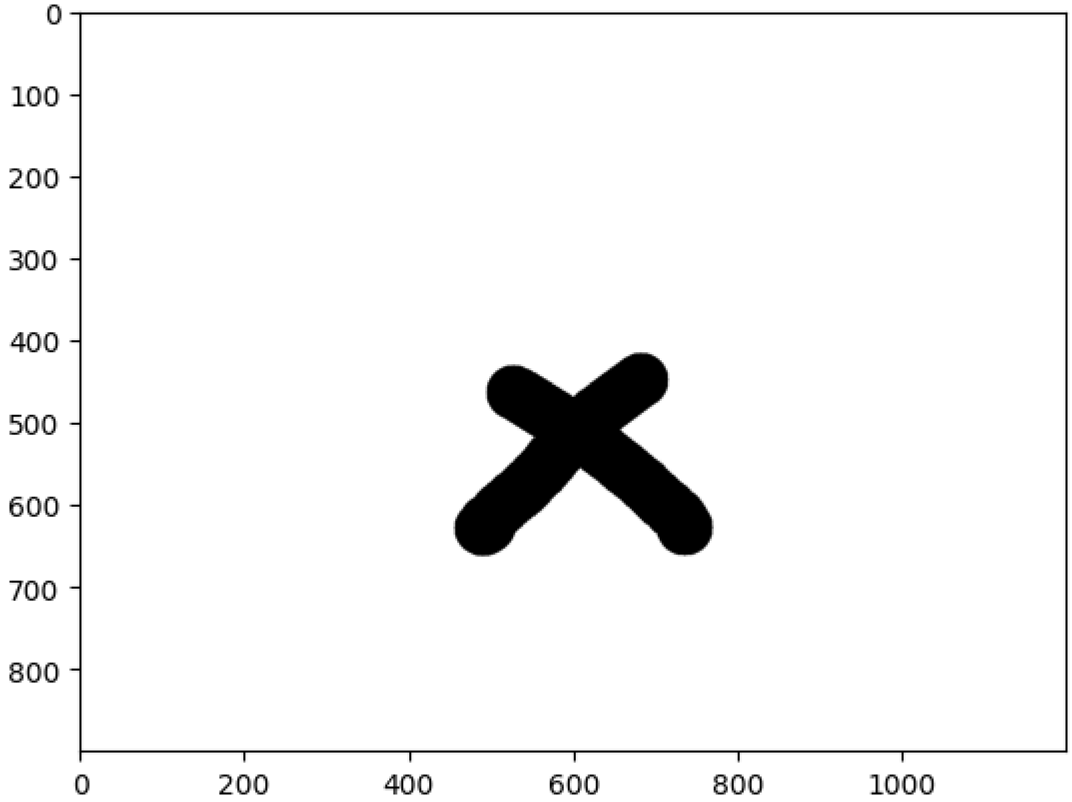


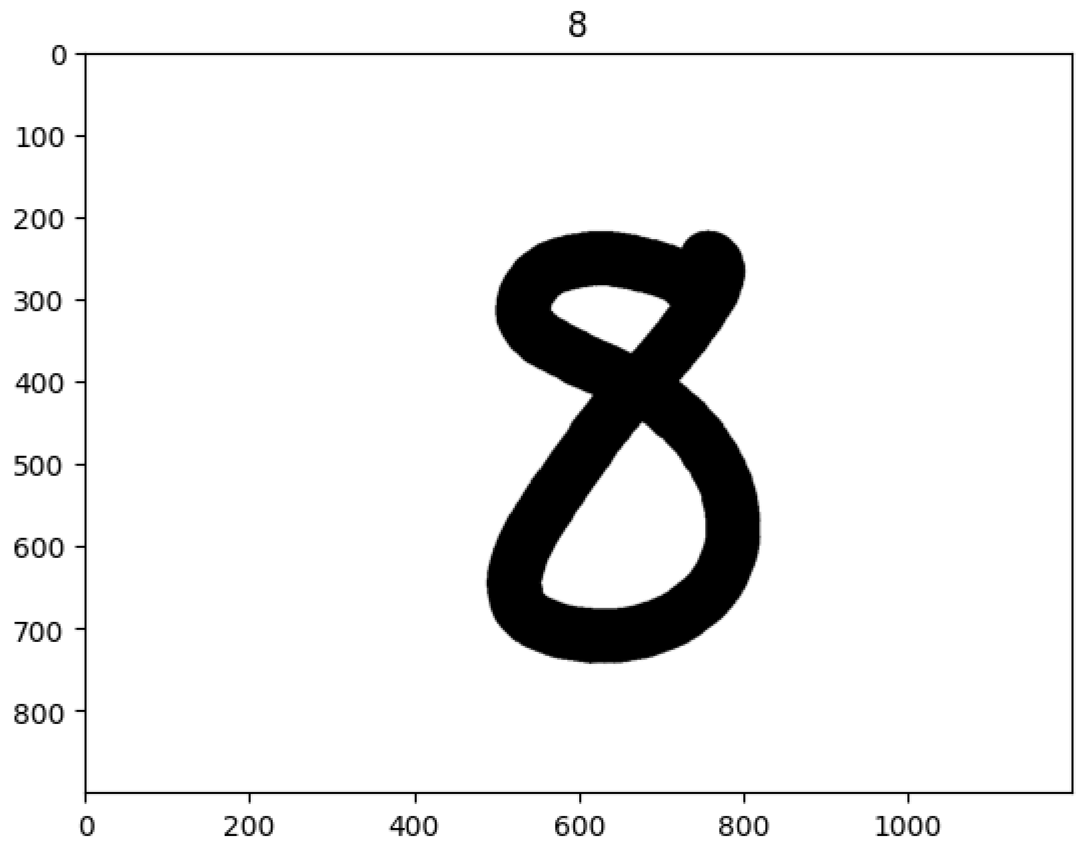
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

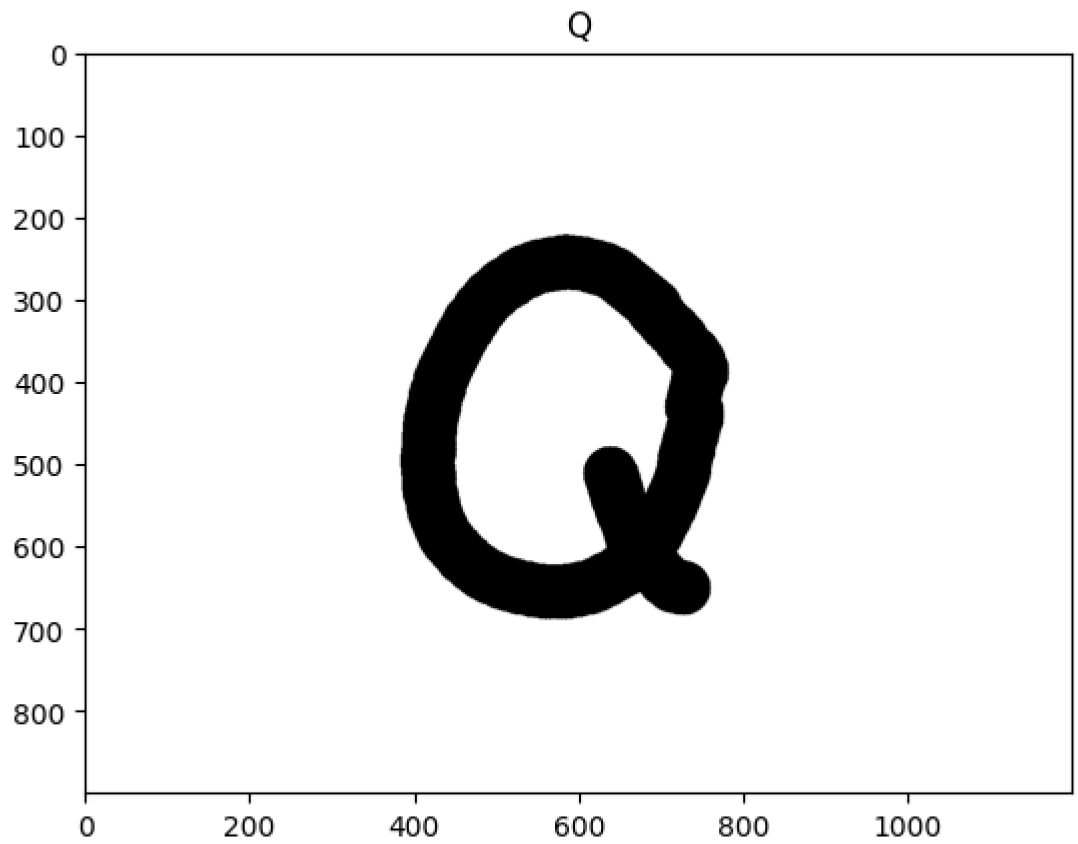
w

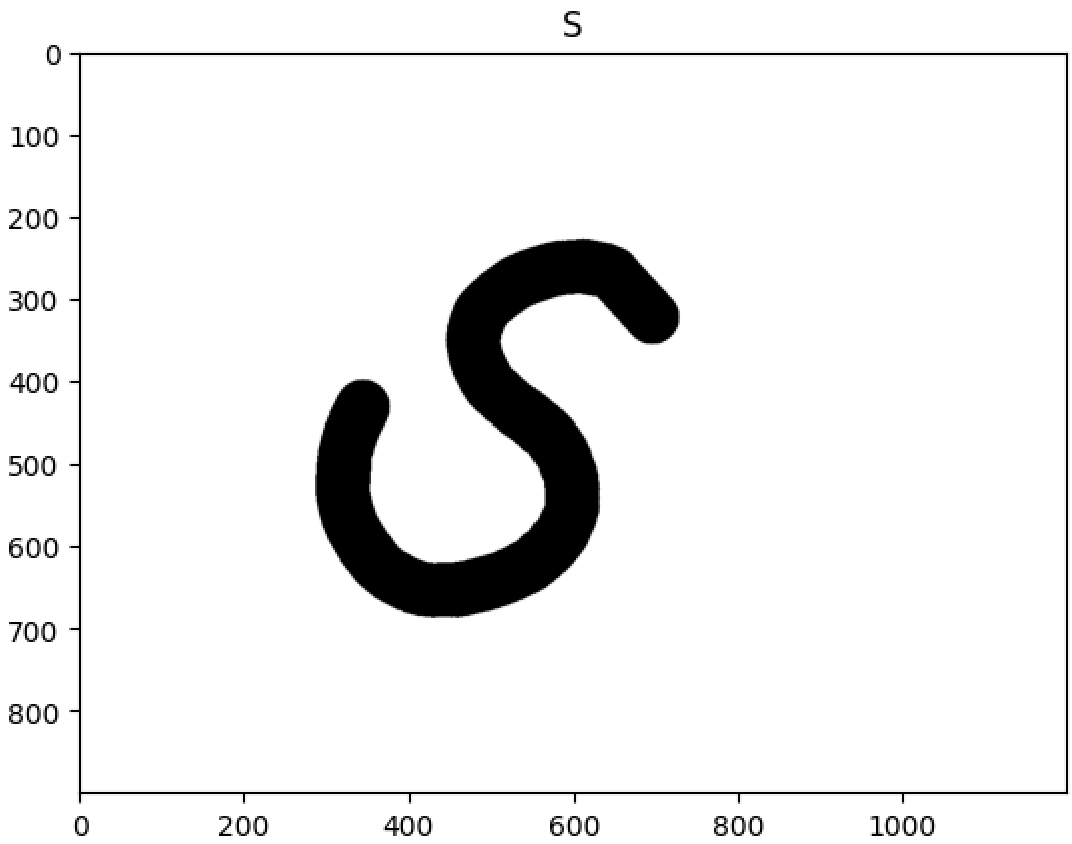

0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000

x


0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000


8
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000


Q
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000




```
In [ ]: from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
import numpy as np

# Function to convert labels to one-hot encoding
def convert_to_one_hot(labels, num_classes):
    one_hot_labels = np.zeros((len(labels), num_classes))
    for i in range(len(labels)):
        one_hot_labels[i, labels[i]] = 1
    return one_hot_labels

# Convert labels to one-hot encoding for training and test sets
train_labels_one_hot = convert_to_one_hot(training_data_frame.classes, 62)
test_labels_one_hot = convert_to_one_hot(test_data_frame.classes, 62)

# Predict probabilities for the test set
test_pred_prob = cnn.predict(test_data_frame)

# Calculate ROC curve and AUC for each class
fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(62):
    fpr[i], tpr[i], _ = roc_curve(test_labels_one_hot[:, i], test_pred_prob[:, i
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curve for each class
plt.figure(figsize=(10, 8))
for i in range(62):
    plt.plot(fpr[i], tpr[i], label='Class {} (AUC = {:.2f})'.format(i, roc_auc[i

plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label='Random')
```

```python
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for each class')
plt.legend()
plt.show()
```

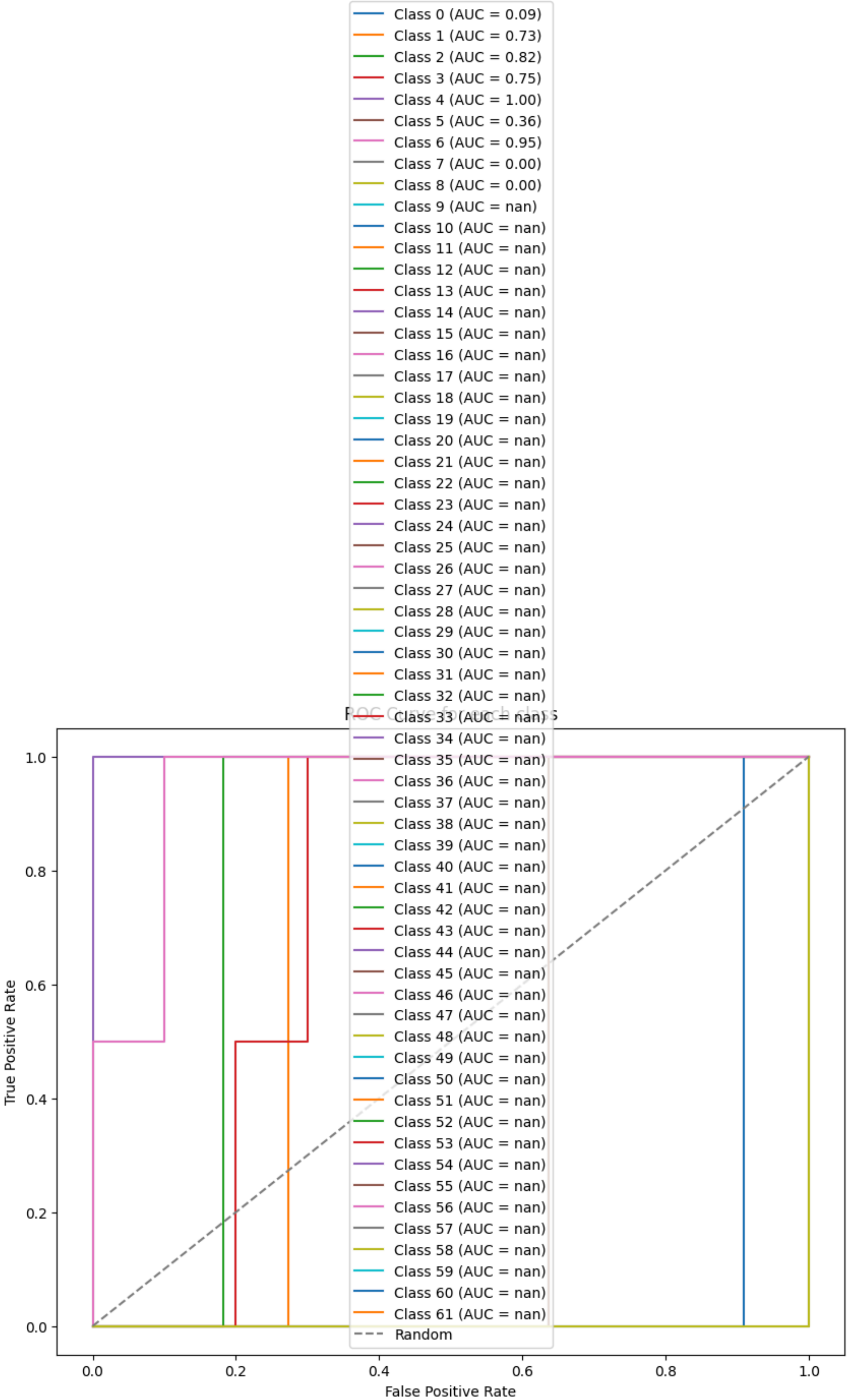
```
1/1 [==============================] - 0s 153ms/step
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```



In [ ]:
```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
```

```python
import numpy as np

def convert_to_one_hot(labels, num_classes):
    one_hot_labels = np.zeros((len(labels), num_classes))
    for i in range(len(labels)):
        one_hot_labels[i, labels[i]] = 1
    return one_hot_labels

# Convert labels to one-hot encoding for training and test sets
train_labels_one_hot = convert_to_one_hot(training_data_frame.classes, 62)
test_labels_one_hot = convert_to_one_hot(test_data_frame.classes, 62)

# Predict probabilities for the test set
test_pred_prob = cnn.predict(test_data_frame)

# Calculate ROC curve and AUC for each class
fpr = dict()
tpr = dict()
roc_auc = dict()

plt.figure(figsize=(10, 8))

for i in range(62):
    fpr[i], tpr[i], _ = roc_curve(test_labels_one_hot[:, i], test_pred_prob[:, i
    roc_auc[i] = auc(fpr[i], tpr[i])

    # Plot ROC curve for each class
    plt.plot(fpr[i], tpr[i], label=f'Class {i} (AUC = {roc_auc[i]:.2f})')

# Plot the random classifier
plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label='Random')

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for each class')
plt.legend()
plt.show()
```

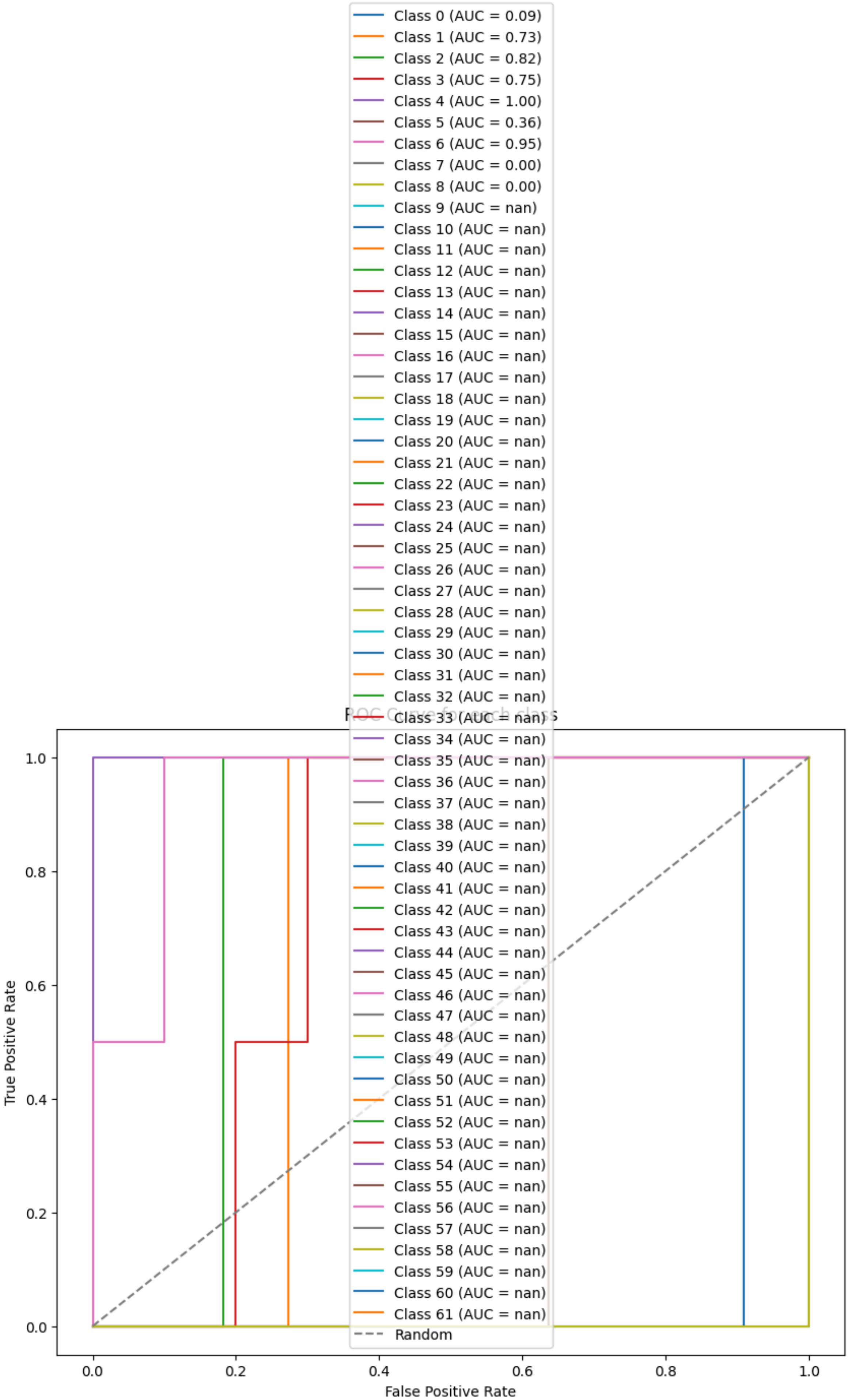
```
1/1 [==============================] - 0s 179ms/step
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```

```
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
C:\Users\Ananya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn
\metrics\_ranking.py:1132: UndefinedMetricWarning: No positive samples in y_true,
true positive value should be meaningless
  warnings.warn(
```


ROC Curve for each class
Class 0 (AUC = 0.09)
Class 1 (AUC = 0.73)
Class 2 (AUC = 0.82)
Class 3 (AUC = 0.75)
Class 4 (AUC = 1.00)
Class 5 (AUC = 0.36)
Class 6 (AUC = 0.95)
Class 7 (AUC = 0.00)
Class 8 (AUC = 0.00)
Class 9 (AUC = nan)
Class 10 (AUC = nan)
Class 11 (AUC = nan)
Class 12 (AUC = nan)
Class 13 (AUC = nan)
Class 14 (AUC = nan)
Class 15 (AUC = nan)
Class 16 (AUC = nan)
Class 17 (AUC = nan)
Class 18 (AUC = nan)
Class 19 (AUC = nan)
Class 20 (AUC = nan)
Class 21 (AUC = nan)
Class 22 (AUC = nan)
Class 23 (AUC = nan)
Class 24 (AUC = nan)
Class 25 (AUC = nan)
Class 26 (AUC = nan)
Class 27 (AUC = nan)
Class 28 (AUC = nan)
Class 29 (AUC = nan)
Class 30 (AUC = nan)
Class 31 (AUC = nan)
Class 32 (AUC = nan)
Class 33 (AUC = nan)
Class 34 (AUC = nan)
Class 35 (AUC = nan)
Class 36 (AUC = nan)
Class 37 (AUC = nan)
Class 38 (AUC = nan)
Class 39 (AUC = nan)
Class 40 (AUC = nan)
Class 41 (AUC = nan)
Class 42 (AUC = nan)
Class 43 (AUC = nan)
Class 44 (AUC = nan)
Class 45 (AUC = nan)
Class 46 (AUC = nan)
Class 47 (AUC = nan)
Class 48 (AUC = nan)
Class 49 (AUC = nan)
Class 50 (AUC = nan)
Class 51 (AUC = nan)
Class 52 (AUC = nan)
Class 53 (AUC = nan)
Class 54 (AUC = nan)
Class 55 (AUC = nan)
Class 56 (AUC = nan)
Class 57 (AUC = nan)
Class 58 (AUC = nan)
Class 59 (AUC = nan)
Class 60 (AUC = nan)
Class 61 (AUC = nan)
Random
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0