```
import cv2
import numpy as np

# Load the image
image = cv2.imread('/Brain Autopsy Lateral View.jpg')

# Get the image dimensions
(h, w) = image.shape[:2]

# Define the center of the image
center = (w // 2, h // 2)

# Define the rotation matrix
angle = 45  # Rotate by 45 degrees
M = cv2.getRotationMatrix2D(center, angle, 1.0)

# Perform the rotation
rotated_image = cv2.warpAffine(image, M, (w, h))

# Save or display the rotated image
cv2.imwrite('rotated_image.jpg', rotated_image)
```

```
True
```

```
# Load the image
image = cv2.imread('/Brain Autopsy Lateral View.jpg')

# Scale the image
scale_factor = 0.5  # Scale down to 50%
scaled_image = cv2.resize(image, None, fx=scale_factor, fy=scale_factor)

# Save or display the scaled image
cv2.imwrite('scaled_image.jpg', scaled_image)
```

```
True
```

```
# Load the image
image = cv2.imread('/Brain Autopsy Lateral View.jpg')

# Define the cropping rectangle (x, y, width, height)
crop_x, crop_y, crop_width, crop_height = 100, 100, 200, 200
cropped_image = image[crop_y:crop_y+crop_height, crop_x:crop_x+crop_width]

# Save or display the cropped image
cv2.imwrite('cropped_image.jpg', cropped_image)
```

```
True
```

```
import tensorflow as tf
from tensorflow.keras import layers, models

# Load the MNIST dataset
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalize the data
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Build a simple neural network model
model = models.Sequential([
    layers.Flatten(input_shape=(28, 28)),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=5)

# Evaluate the model
```

```
# Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc}')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
/usr/local/lib/python3.10/dist-packages/keras/src/layers/reshaping/flatten.py:37: UserWarning: Do not pass an `input_shape`/
  super().__init__(**kwargs)
Epoch 1/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.8804 - loss: 0.4242
Epoch 2/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9653 - loss: 0.1193
Epoch 3/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9766 - loss: 0.0753
Epoch 4/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 5s 3ms/step - accuracy: 0.9828 - loss: 0.0565
Epoch 5/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9871 - loss: 0.0426
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9724 - loss: 0.0938
Test accuracy: 0.9751999974250793
```

Start coding or generate with AI.