

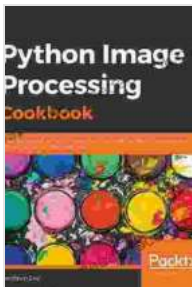
# Over 60 Recipes to Help You Perform Complex Image Processing and Computer Vision Tasks

Before you can start using the recipes in this article, you'll need to install a few software libraries. These libraries provide the necessary functionality for performing image processing and computer vision tasks.

The following libraries are required for all of the recipes in this article:

- OpenCV
- NumPy
- Matplotlib

You can install these libraries using the following commands:



## Python Image Processing Cookbook: Over 60 recipes to help you perform complex image processing and computer vision tasks with ease by Shannon Duffy

★★★★☆ 4.2 out of 5

Language	: English
File size	: 86658 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 438 pages
Paperback	: 37 pages
Item Weight	: 3.84 ounces
Dimensions	: 6 x 0.1 x 9 inches



```
pip install opencv-python pip install numpy pip install matplotlib
```

Once you have installed the required libraries, you can start using the recipes in this article.

The following recipes are divided into several categories:

- Basic Image Manipulation
- Advanced Image Processing
- Object Detection and Tracking
- Machine Learning for Image Processing

The following recipes cover basic image manipulation tasks, such as resizing, cropping, and rotating images.

- **Resize an image:**

```
python import cv2
```

## Read the image

```
image = cv2.imread("image.jpg")
```

## Resize the image to half its original size

```
resized_image = cv2.resize(image, (0, 0),fx=0.5, fy=0.5)
```

# Save the resized image

```
cv2.imwrite("resized_image.jpg", resized_image)
```

- **Crop an image:**python import cv2

# Read the image

```
image = cv2.imread("image.jpg")
```

# Crop the image to a square

```
cropped_image = image[0:512, 0:512]
```

# Save the cropped image

```
cv2.imwrite("cropped_image.jpg", cropped_image)
```

- **Rotate an image:**python import cv2

# Read the image

```
image = cv2.imread("image.jpg")
```

# Rotate the image by 90 degrees

```
rotated_image = cv2.rotate(image, cv2.ROTATE_90_CLOCKWISE)
```

# Save the rotated image

```
cv2.imwrite("rotated_image.jpg", rotated_image)
```

The following recipes cover more advanced image processing tasks, such as filtering, thresholding, and morphological operations.

- **Apply a Gaussian blur to an image:**`python import cv2`

# Read the image

```
image = cv2.imread("image.jpg")
```

# Apply a Gaussian blur to the image

```
blurred_image = cv2.GaussianBlur(image, (5, 5),0)
```

# Save the blurred image

```
cv2.imwrite("blurred_image.jpg", blurred_image)
```

- **Threshold an image:**python import cv2

# Read the image

```
image = cv2.imread("image.jpg")
```

# Convert the image to grayscale

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

# Threshold the image

```
threshold_image = cv2.threshold(gray_image, 127, 255,  
cv2.THRESH_BINARY)[1]
```

# Save the threshold image

```
cv2.imwrite("threshold_image.jpg", threshold_image)
```

- **Perform a morphological operation on an image:**`python import cv2`

## Read the image

```
image = cv2.imread("image.jpg")
```

## Convert the image to grayscale

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

## Perform a morphological operation on the image

```
morphological_image = cv2.morphologyEx(gray_image,  
cv2.MORPH_CLOSE, cv2.getStructuringElement(cv2.MORPH_RECT, (5,  
5)))
```

## Save the morphological image

```
cv2.imwrite("morphological_image.jpg", morphological_image)
```

The following recipes cover object detection and tracking tasks, such as finding objects in an image and tracking them over time.

- **Detect objects in an image:**`python import cv2`

## Read the image

```
image = cv2.imread("image.jpg")
```

## Convert the image to grayscale

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

## Detect objects in the image

```
objects = cv2.findContours(gray_image, cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)[0]
```

## Draw the objects on the image

```
cv2.drawContours(image, objects, -1, (0, 255, 0),2)
```

# Save the image with the objects

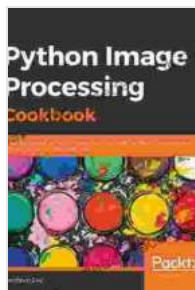
```
cv2.imwrite("objects_image.jpg", image)
```

- **Track objects in a video:**`python import cv2`

# Create a video capture object

```
cap = cv2.VideoCapture("video.mp4")
```

# Create an object tracker



**Python Image Processing Cookbook: Over 60 recipes to help you perform complex image processing and computer vision tasks with ease** by Shannon Duffy

★★★★☆ 4.2 out of 5

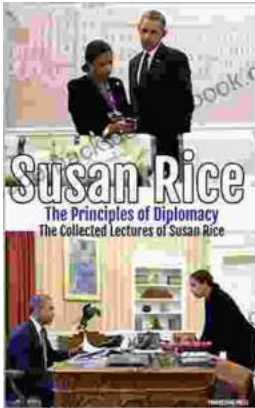
Language	: English
File size	: 86658 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 438 pages
Paperback	: 37 pages
Item Weight	: 3.84 ounces



Dimensions : 6 x 0.1 x 9 inches

FREE

DOWNLOAD E-BOOK



## Susan Rice: The Principles of Diplomacy

Susan Rice is a leading expert on diplomacy. She has served as the U.S. Ambassador to the United Nations and as National Security Advisor. In these roles, she...



## The Symphony Listener's Guide: Unlocking the Beauty of Orchestral Music

Immerse yourself in the captivating world of symphonic music with our comprehensive Symphony Listener's Guide. Designed to illuminate the intricate layers of...