Operating on images using OpenCV-Python

Let's take a look at how to operate on images using OpenCV-Python. In this recipe, we will see how to load and display an image. We will also look at how to crop, resize, and save an image to an output file.

How to do it...

1. Create a new Python file, and import the following packages:

```
import sys
import cv2
import numpy as np
```

2. Specify the input image as the first argument to the file, and read it using the image read function. We will use forest.jpg, as follows:

```
# Load and display an image -- 'forest.jpg'
input_file = sys.argv[1]
img = cv2.imread(input_file)
```

3. Display the input image, as follows:

```
cv2.imshow('Original', img)
```

4. We will now crop this image. Extract the height and width of the input image, and then specify the boundaries:

```
# Cropping an image
h, w = img.shape[:2]
start_row, end_row = int(0.21*h), int(0.73*h)
start_col, end_col= int(0.37*w), int(0.92*w)
```

5. Crop the image using NumPy style slicing and display it:

```
img_cropped = img[start_row:end_row, start_col:end_col]
cv2.imshow('Cropped', img_cropped)
```

6. Resize the image to 1.3 times its original size and display it:

```
# Resizing an image
scaling_factor = 1.3
img_scaled = cv2.resize(img, None, fx=scaling_factor, fy=scaling_factor)
interpolation=cv2.INTER_LINEAR)
cv2.imshow('Uniform resizing', img_scaled)
```

7. The previous method will uniformly scale the image on both dimensions. Let's assume that we want to skew the image based on specific output dimensions. We use the following code:

```
img_scaled = cv2.resize(img, (250, 400), interpolation=cv2.INTER_AREA)
cv2.imshow('Skewed resizing', img_scaled)
```

8. Save the image to an output file:

```
# Save an image
output_file = input_file[:-4] + '_cropped.jpg'
cv2.imwrite(output_file, img_cropped)
cv2.waitKey()
```

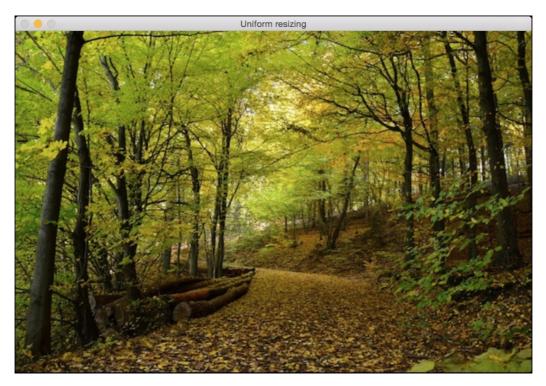
- 9. The waitKey() function displays the images until you hit a key on the keyboard.
- 10. The full code is given in the <code>operating_on_images.py</code> file that is already provided to you. If you run the code, you will see the following input image:



11. The second output is the cropped image:



12. The third output is the uniformly resized image:



13. The fourth output is the skewed image:

