note.nkmk.me

Top > Python > OpenCV

Detect and read barcodes with OpenCV in Python

Posted: 2022-10-16 | Tags: Python, OpenCV, Image Processing

This article describes how to detect and read barcodes with OpenCV in Python.

- `cv2.barcode` is included in the OpenCV contrib module
- Super Resolution Model
- Detect and read barcodes from an image
- Detect and read barcodes from camera video

See the following article on how to detect and read QR codes instead of barcodes.

- Detect and read QR codes with OpenCV in Python

You can also use ZBar (pyzbar). Although not thoroughly verified, ZBar seems to have better detection accuracy.

- Detect and read barcodes and QR codes with ZBar in Python

The version of OpenCV used in the sample code is `4.6.0` .

```
import cv2

print(cv2.__version__)
# 4.6.0
```

source: opencv_barcode.py

`cv2.barcode` is included in the OpenCV `contrib` module

The `cv2.barcode` to detect and read barcodes is included in the `contrib` module (as of version `4.6.0`).

- OpenCV: Bar code Recognition
- OpenCV: cv::barcode::BarcodeDetector Class Reference

In the output of `cv2.getBuildInformation()`, `barcode` must be included in `To be built` of `OpenCV modules`.

- Check OpenCV Build Information: getBuildInformation()

For example, on macOS, if you installed OpenCV with Homebrew, the `contrib` module should be included, but if you installed `opencv-python` with `pip`, it might not be included. You need to install OpenCV with `pip install opencv-contrib-python`.

- opencv-python · PyPI

Super Resolution Model

The official tutorial introduces the Super Resolution Model.

- OpenCV: Bar code Recognition

You can download and use `sr.prototxt`, `sr.caffemodel` from the following repository.

- WeChatCV/opencv_3rdparty: OpenCV - 3rdparty

Specify the path of the downloaded file to `cv2.barcode.BarcodeDetector()`. If omitted, no model is used.

```
bd = cv2.barcode.BarcodeDetector('/path/to/sr.prototxt', '/path/to/sr.caffemodel')
```

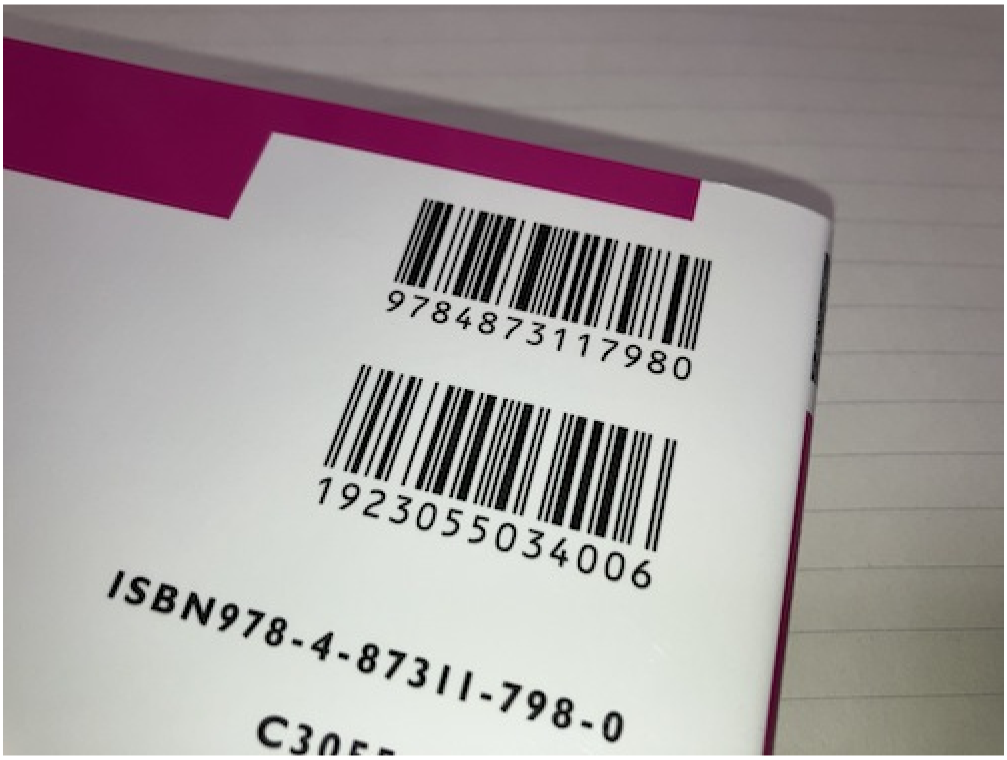
I may be doing something wrong, but I did not notice any difference in accuracy with or without the model in my environment. It is not used in the sample code below.

Detect and read barcodes from an image

The following barcode image is used as an example.

```
img = cv2.imread('data/src/barcode.jpg')
```

source: opencv_barcode.py



Create an instance of `cv2.barcode.BarcodeDetector` and execute `detectAndDecode()`. Although `detect()` to only detect and `decode()` to decode based on the detected coordinates are also provided, they are not mentioned here.

Note that the results may differ depending on the version and environment.

```
bd = cv2.barcode.BarcodeDetector()
# bd = cv2.barcode.BarcodeDetector('path/to/sr.prototxt', 'path/to/sr.caffemodel')
```

```
retval, decoded_info, decoded_type, points = bd.detectAndDecode(img)
```

source: opencv_barcode.py

`retval` is `True` if a barcode is detected and `False` if none is detected.

```
print(retval)
# True
```

source: opencv_barcode.py

`decoded_info` is a tuple whose elements are strings stored in barcodes. If it can be detected but not decoded, it is an empty string `''`.

```
print(decoded_info)
# ('1923055034006', '9784873117980')
```

source: opencv_barcode.py

`decoded_type` is a tuple whose elements are numbers representing barcode types.

- OpenCV: Barcode detecting and decoding methods

```
print(decoded_type)
# (2, 2)

print(cv2.barcode.EAN_13)
# 2
```

source: opencv_barcode.py

`points` is a `numpy.ndarray` representing the coordinates of the four corners of the detected QR Code.

```
print(type(points))
# <class 'numpy.ndarray'>

print(points)
# [[[142.38849 221.83641]
#   [156.36218 172.35411]
#   [356.90564 228.98714]
#   [342.93195 278.46942]]
#
#  [[180.30583 128.89304]
#   [191.59013  88.83808]
#   [371.00458 139.38284]
#   [359.72028 179.4378 ]]]

print(points.shape)
# (2, 4, 2)
```

source: opencv_barcode.py

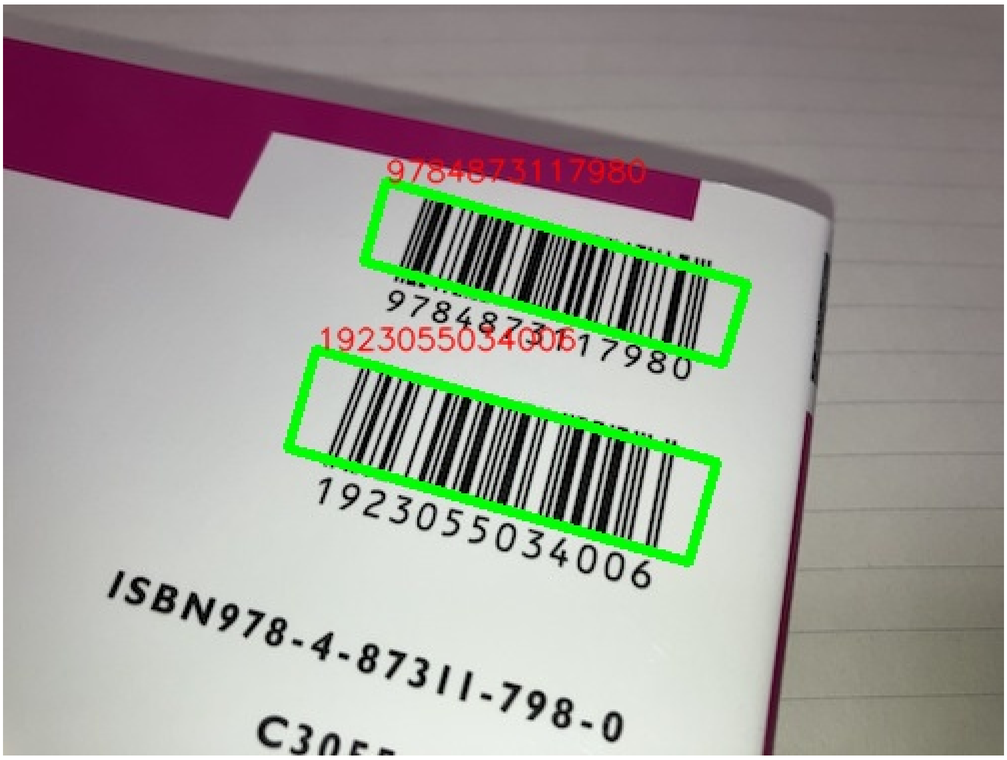
Draw frames for the detected barcode and superimpose the decoded text.

```
img = cv2.polylines(img, points.astype(int), True, (0, 255, 0), 3)

for s, p in zip(decoded_info, points):
    img = cv2.putText(img, s, p[1].astype(int),
                      cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)

cv2.imwrite('data/dst/barcode_opencv.jpg', img)
```

source: opencv_barcode.py



Detect and read barcodes from camera video

The following is a sample code that detects and reads barcodes from real-time camera video.

See the following article for more information on the handling of videos in OpenCV.

- Capture video from camera/file with OpenCV in Python

Press `q` on the keyboard to exit.

```python
import cv2

camera_id = 0
delay = 1
window_name = 'OpenCV Barcode'

bd = cv2.barcode.BarcodeDetector()
cap = cv2.VideoCapture(camera_id)

while True:
    ret, frame = cap.read()

    if ret:
        ret_bc, decoded_info, _, points = bd.detectAndDecode(frame)
        if ret_bc:
            frame = cv2.polylines(frame, points.astype(int), True, (0, 255, 0), 3)
            for s, p in zip(decoded_info, points):
                if s:
                    print(s)
                    frame = cv2.putText(frame, s, p[1].astype(int),
                                        cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 2, cv2.LINE_A
        cv2.imshow(window_name, frame)

    if cv2.waitKey(delay) & 0xFF == ord('q'):
        break

cv2.destroyWindow(window_name)
```

source: opencv_barcode_videocapture.py

In an actual application, the `while` loop would be terminated by `break` when the `decoded_info` string is obtained, and the application would move on to operation using the string.

Related Categories

- Python
- OpenCV
- Image Processing

Related Articles

- Convert BGR and RGB with Python, OpenCV (cvtColor)
- Binarize image with Python, NumPy, OpenCV
- Detect and read barcodes and QR codes with ZBar in Python
- Get image size (width, height) with Python, OpenCV, Pillow (PIL)
- Reading and saving image files with Python, OpenCV (imread, imwrite)
- Detect and read QR codes with OpenCV in Python
- Alpha blending and masking of images with Python, OpenCV, NumPy
- Concatenate images with Python, OpenCV (hconcat, vconcat, np.tile)
- OpenCV, NumPy: Rotate and flip image
- Check OpenCV Build Information: getBuildInformation()
- Concatenate images with Python, Pillow
- Create a montage of images with Python, scikit-image (skimage.util.montage)
- Capture video from camera/file with OpenCV in Python
- How to create animated GIF with Pillow in Python
- Invert image with Python, Pillow (Negative-positive inversion)