

Forum:[Hack Forums](#)

Thread: [Interact with program on background](#)

Mohinder - June 23, 2018, 3:38 am

Ola fellow-coders,

I want to use pytesseract to read text out of an screen once in a while.

The screen I want to extract information out can be on the background, would it be possible to interact with it (leaving it in the background)?

Thanks in advance,

Mo

hunter24957 - June 23, 2018, 5:40 am

When you say "screen" you mean a "window", right?

Assuming you are using Windows and that pytesseract will need an image for character recognition, we'd need to take a snapshot of the background window, more specifically the area of the window where the text lies and then feed that to pytesseract.

Looking at the MSDN Windows API documentation we can see a function called [PrintWindow](#). The [PrintWindow](#) function is defined as follows:

PHP Code

```
<?php
// The PrintWindow function copies a visual window into the specified device context (DC), typically a printer DC.
```

```
BOOL PrintWindow(
    HWND hwnd,
    HDC  hdcBlt,
    UINT nFlags
);
```

This function will allow us to take a snapshot of the given background window and save the resulting bitmap to a buffer.

With a little bit of research we can find out how to call this function in Python:

<http://www.upgradeyourtechlife.com/blog/?p=82>

Ok, so now we have a way of obtaining a snapshot of the Window area of a background window, now we have to crop the snapshot to the area of the window containing the text.

Since the documentation I have linked you uses Pillow to cast the buffer, we can simply use the [Image.crop\(\)](#) method that takes a rectangle of the region to crop to. It is defined below:

Code `Image.crop(box=None)`

Returns a rectangular region from this image. The box is a 4-tuple defining the left, upper, right, and lower pixel coordinate.

This is a lazy operation. Changes to the source image may or may not be reflected in the cropped image. To break the connection, call the load() method on the cropped copy.

Parameters: box – The crop rectangle, as a (left, upper, right, lower)-tuple.

Return type: Image

Returns: An Image object.

Now we can successfully grab an image of the text on a background window without having to move the window to the front. I have included some code I have written for this down below, enjoy! ?

PHP Code

```
<?php
import win32gui
import ctypes
from PIL import Image

# takes a snapshot of the specified window
def window_snapshot(title):
    # finds the specified window
    hwnd = win32gui.FindWindow(None, title)
    # validates the handle
    if hwnd:
        # gets the client area of the window
        left, top, right, bottom = win32gui.GetWindowRect(hwnd)
        # calculates the window width
        width = right - left
        # calculates the window height
        height = bottom - top
        # creates objects to store bitmap
        hwndDC = win32gui.GetWindowDC(hwnd)
        mfcDC = win32ui.CreateDCFromHandle(hwndDC)
        saveDC = mfcDC.CreateCompatibleDC()
        saveBitMap = win32ui.CreateBitmap()
        saveBitMap.CreateCompatibleBitmap(mfcDC, width, height)
        saveDC.SelectObject(saveBitMap)
        # takes a snapshot of the window and saves it into our buffer
        result = ctypes.windll.user32.PrintWindow(hwnd, saveDC.GetSafeHdc(), 0)
        # gets the attributes of the bitmap image
        bmpinfo = saveBitMap.GetInfo()
        # gets the bitmap decoder
        bmpstr = saveBitMap.GetBitmapBits(True)
        # uses pillow to create an image from the buffer we have obtained
        image = Image.frombuffer('RGB', (bmpinfo['bmWidth'], bmpinfo['bmHeight']), bmpstr, 'raw', 'BGRX',
0, 1)
        # releases bitmap objects
        win32gui.DeleteObject(saveBitMap.GetHandle())
        saveDC.DeleteDC()
        mfcDC.DeleteDC()
        win32gui.ReleaseDC(hwnd, hwndDC)
        # validates the result
        if result:
            # returns the image
            return image
```

```
# takes a snapshot of an area of the specified window
def window_snapshot_area(title, area):
    # takes a window snapshot
    snapshot = window_snapshot(title)
    # validates the result
    if snapshot:
        # crops and returns the snapshot
        return snapshot.crop(area)

if __name__ == '__main__':
    # takes a snapshot of the window
    snapshot = window_snapshot_area("BackgroundWindowTitle", (0, 0, 100, 100))
    # saves the snapshot to the disk, optionally you can send the image to pytesseract
    snapshot.save("BackgroundWindowSnapshot.png")
    # waits to exit
    input()
```

Mohinder - June 24, 2018, 2:03 am

Thanks for this very detailed information. :)

hunter24957 - June 24, 2018, 6:31 am

(Jun 24, 2018 02:03 AM)Mohinder Wrote: [\[CITE\]](#) Thanks for this very detailed information. :)
No problem man, enjoy! ?