



BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin



LAB REPORT 05

Student's ID :
Student's name : Hồ Phúc Lâm
Subject : PTHTDPT
Instructor : Nguyễn Thành Thái
Faculty : Công Nghệ Thông Tin
Completed Date : 18/09/2024

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

LAB 3-1 IMAGE

1) Mục đích yêu cầu :

- +Củng cố kiến thức cơ bản về IMAGE, các khái niệm về hệ màu, độ phân giải,...
- +Tiếp cận thư viện xử lý image như PIL(pillow), opencv,... các API hỗ trợ,...

2) Mở link sau :

<https://neptune.ai/blog/pil-image-tutorial-for-machine-learning>

3) Thực hiện:

- Tìm hiểu khái niệm Essential PIL image concepts
- Tìm hiểu thuộc tính BANDS (kênh: channels)
- Tìm hiểu thuộc tính Size
- Tìm hiểu chức năng Loading and saving images.
- Tìm hiểu chức năng Reading from a string
- Tìm hiểu chức năng Convert to JPEG, Create JPEG thumbnails
- Tìm hiểu chức năng Image manipulation: Cropping images, Rotation images, Merging images, Flip images
- Tìm hiểu chức năng PIL image to numPy Array
- Tìm hiểu chức năng Color transformations

4) Công cụ hỗ trợ : Python programming language

a)Spyder IDE hoặc PYTHON commandline

b)Cài đặt các thư viện hỗ trợ :

Câu lệnh cài đặt : pip install <gói cài đặt>

Vd : pip install opencv-python

pip install pillow

pip install matplotlib

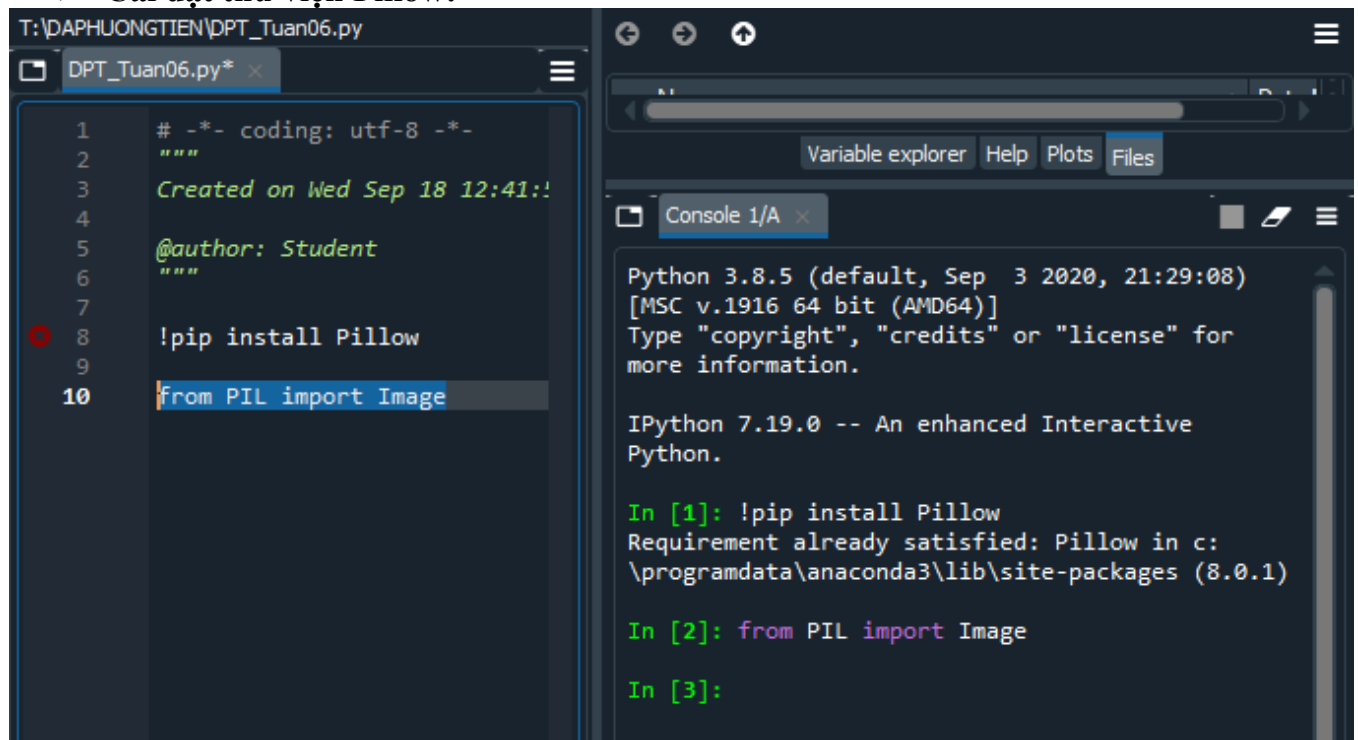
- Cài đặt thư viện (nếu chưa có): mở CMD trên Windows, gõ pip install simpleaudio, cài xong chạy python -> gõ lệnh import simpleaudio as sa kiểm tra lỗi...
- Vào trang <https://file-examples.com/index.php/sample-images-download/sample-jpg-download/> download file mẫu *.jpg và lưu trong thư mục
- Mở notepad viết chương trình *.py lưu trong một thư mục {thường là chung với thư mục của file Wav}
- Chạy thử code : >python baitap1.py
- Lưu các bài tập trong thư mục, nén và nộp (cuối giờ thực hành)

THỰC HIỆN

➤ Tìm hiểu khái niệm Essential PIL image concepts

- Essential Pil (Pillow) Image và PIL (Python Imaging Library):
- PIL là viết tắt của Python Image Library. Trong bài viết của liên kết được đưa ra ở mục 2.
- Thư viện này cho phép xử lý và thao tác hình ảnh trong python. Pillow là một nhánh mở rộng và cải tiến của PIL, cung cấp nhiều tính năng mới và sửa lỗi.
- Cung cấp hỗ trợ cho nhiều định dạng hình ảnh khác nhau gồm các định dạng JPEG và PNG khá phổ biến.
- Lý do nên sử dụng Pillow vì nó khá dễ sử dụng và phổ biến với python. Cung cấp nhiều phương pháp xử lý hình ảnh khác nhau, khá hữu ích trong việc tăng cường đào tạo cho các vấn đề về thị giác máy tính (computer vision problems).

➤ Cài đặt thư viện Pillow:



The screenshot displays a Jupyter Notebook interface. On the left, a code editor shows a Python script with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Sep 18 12:41:!!
4
5 @author: Student
6 """
7
8 !pip install Pillow
9
10 from PIL import Image
```

On the right, the console output shows the execution of the commands:

```
Python 3.8.5 (default, Sep 3 2020, 21:29:08)
[MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for
more information.

IPython 7.19.0 -- An enhanced Interactive
Python.

In [1]: !pip install Pillow
Requirement already satisfied: Pillow in c:
\programdata\anaconda3\lib\site-packages (8.0.1)

In [2]: from PIL import Image

In [3]:
```

Tải file ảnh peacock theo liên kết và thực hiện lệnh sau:

```
10 from PIL import Image
11
12 # Bước đầu tiên là nhập Imagelớp từ PI
13 im = Image.open("peacock.jpg")
14 print(im)
15
```

```
Console 1/A x
In [6]: im = Image.open("peacock.jpg")
In [7]: print(im)
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=4000x6000 at 0x28F31F6C4F0>
In [8]:
```

➤ **Tìm hiểu thuộc tính BANDS (kênh: channels)**

- Mỗi hình ảnh có một hoặc nhiều dải. Sử dụng Pillow, chúng ta có thể lưu trữ một hoặc nhiều dải trong một hình ảnh. Ví dụ, một hình ảnh màu thường sẽ có các dải 'R', 'G' và 'B' cho Đỏ, Xanh lam và Xanh lục tương ứng.
- Sau đây là cách chúng ta có thể lấy được các dải màu cho hình ảnh đã nhập ở trên.

```
15
16 #lấy các dải màu cho hình ảnh đã nhập ở trên.
17 im.getbands()
```

```
Console 1/A x
In [9]: im.getbands()
Out[9]: ('R', 'G', 'B')
```

➤ **Tìm hiểu thuộc tính Size**

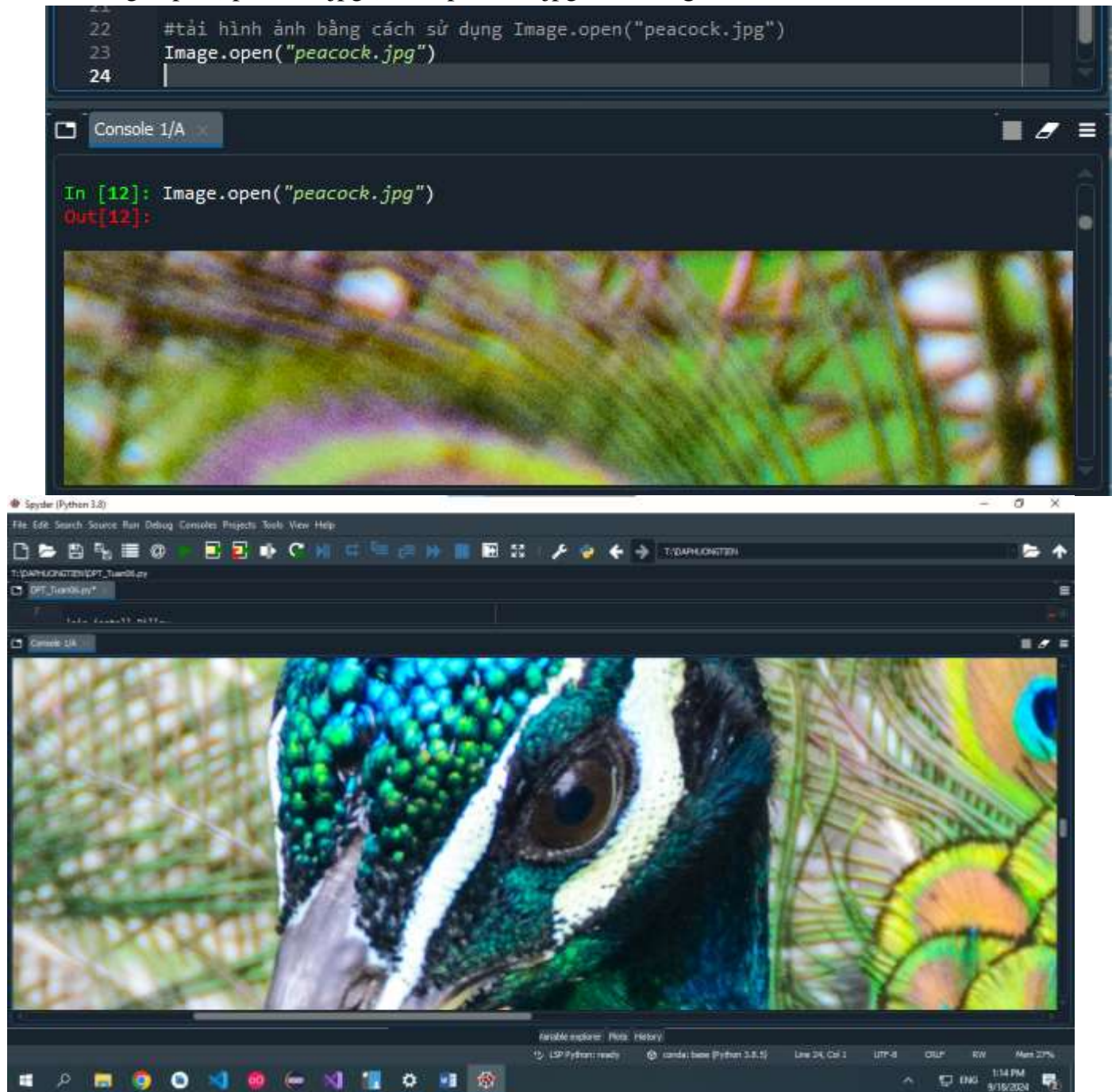
- Chúng ta cũng có thể lấy được kích thước của hình ảnh thông qua thuộc tính image.

```
18
19 #kích thước của hình ảnh
20 im.size
```

```
Console 1/A x
In [11]: im.size
Out[11]: (4000, 6000)
In [12]:
```

➤ **Tìm hiểu chức năng Loading and saving images.**

- Chúng ta đã thấy rằng chúng ta có thể tải hình ảnh bằng cách sử dụng `Image.open("peacock.jpg")` với "peacock.jpg" là đường dẫn đến vị trí của hình ảnh.



➤ **Tìm hiểu chức năng Reading from a string**

- Để chứng minh cách đọc chuỗi hình ảnh bằng Pillow, chúng ta sẽ bắt đầu bằng cách chuyển đổi hình ảnh thành chuỗi thông qua “base64”.


```
23 image.open("peacock.jpg")
24
25 #Đọc từ một chuỗi, bằng cách chuyển đổi hình ảnh thành chuỗi thông qua "base64".
26 import base64
27 with open("peacock.jpg", "rb") as image:
28     image_string = base64.b64encode(image.read())
```

```
Console 1/A x
In [14]:
...: import base64
...: with open("peacock.jpg", "rb") as image:
...:     image_string = base64.b64encode(image.read())
In [15]:
```

- Bây giờ chúng ta có thể giải mã chuỗi hình ảnh và tải nó dưới dạng hình ảnh bằng cách sử dụng Imagelớp từ PIL.

```
29
30 #giải mã chuỗi hình ảnh và tải nó dưới dạng hình ảnh bằng Imagelớp từ PIL.
31 import io
32 image = io.BytesIO(base64.b64decode(image_string))
33 Image.open(image)
```

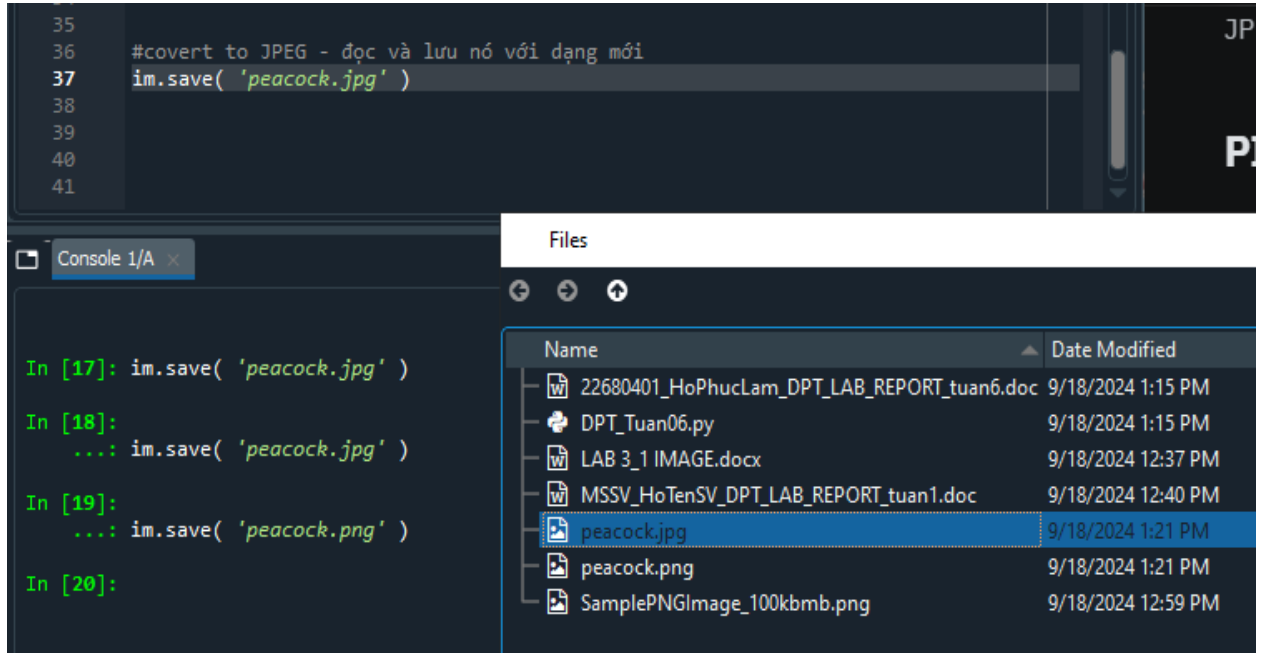
```
Console 1/A x
In [15]:
...: import io
...: image = io.BytesIO(base64.b64decode(image_string))
...: Image.open(image)
Out[15]:
```



➤ **Tìm hiểu chức năng Convert to JPEG, Create JPEG thumbnails**

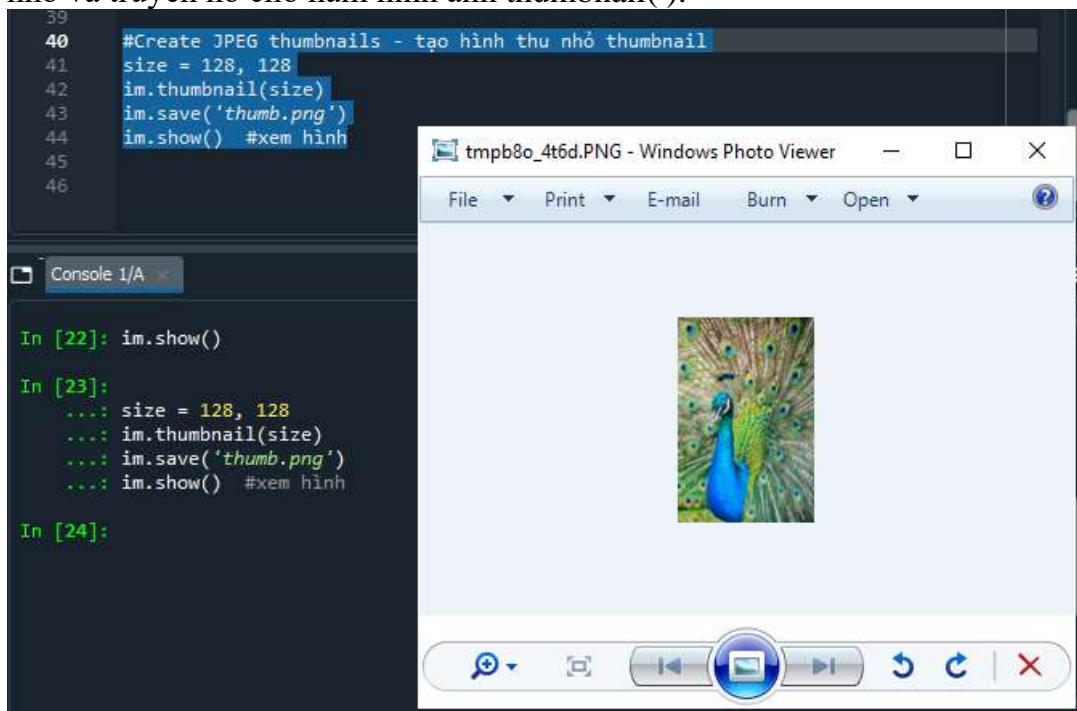
➤ **Convert to JPEG – Chuyển hình ảnh sang định dạng JPEG**

- Chuyển đổi hình ảnh được thực hiện bằng cách đọc hình ảnh và lưu nó với định dạng mới. Sau đây là cách chúng tôi chuyển đổi hình ảnh PNG con công sang định dạng JPEG.



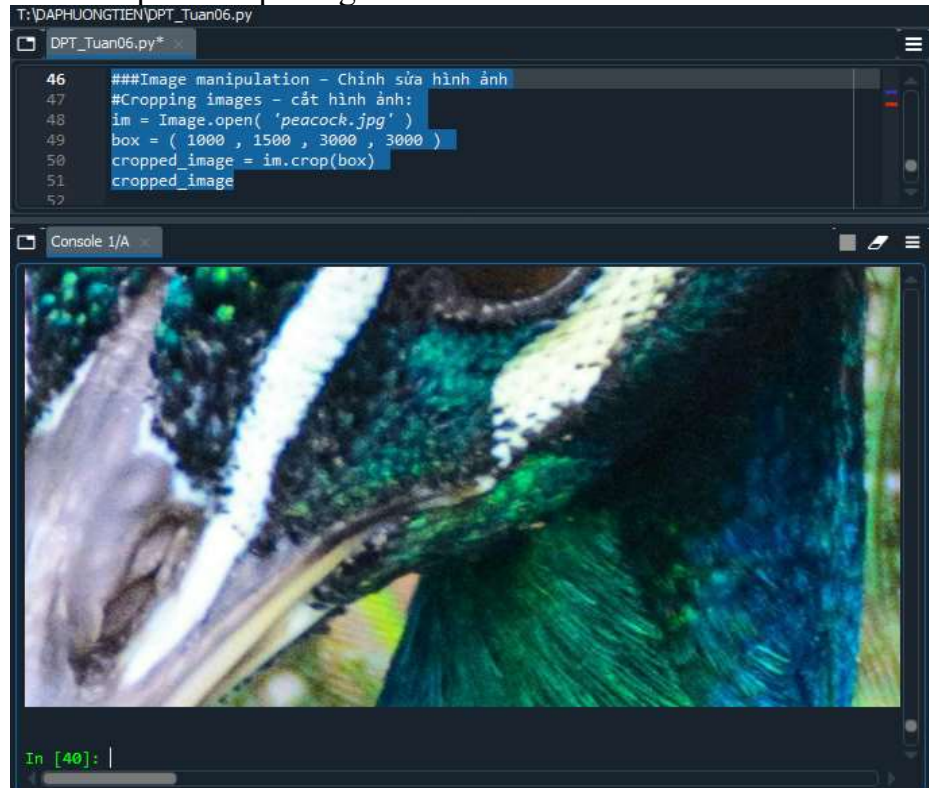
○
➤ **Create JPEG thumbnails - Tạo hình thu nhỏ JPEG:**

- Trong một số trường hợp, chúng tôi muốn giảm kích thước hình ảnh.
- Ví dụ, người ta có thể giảm kích thước của một hình ảnh để có được hình thu nhỏ của hình ảnh. Điều này có thể thực hiện được bằng cách xác định kích thước của hình thu nhỏ và truyền nó cho hàm hình ảnh thumbnail().



➤ **Tìm hiểu chức năng Image manipulation: Cropping images, Rotation images, Merging images, Flip images**

- **Cropping images – cắt hình ảnh:** để cắt một hình ảnh, chúng ta bắt đầu bằng cách định nghĩa một hộp chỉ ra vùng hình ảnh mà chúng ta muốn cắt. Tiếp theo, chúng ta truyền hộp đó cho hàm `crop` của lớp Image.



- box = (100, 150, 300, 300)
- box = (left, upper, right, lower)
 - **left:** Tọa độ x của điểm góc trên bên trái (100).
 - **upper:** Tọa độ y của điểm góc trên bên trái (150).
 - **right:** Tọa độ x của điểm góc dưới bên phải (300).
 - **lower:** Tọa độ y của điểm góc dưới bên phải (300).



○

- **Rotation images – Xoay hình ảnh:** Việc xoay hình ảnh được thực hiện thông qua hàm `rotate()` của lớp `Image`.

```
61 #Rotation images - Xoay hình ảnh
62 #Lấy hình đã thumbnail để dễ thao tác
63 #do hình ban đầu có kích thước quá lớn
64 im = Image.open('thumb.png')
65 rotated = im.rotate(180)
66 rotated
```


```
In [47]: rotated = im.rotate(180)
...: rotated
Out[47]:
```

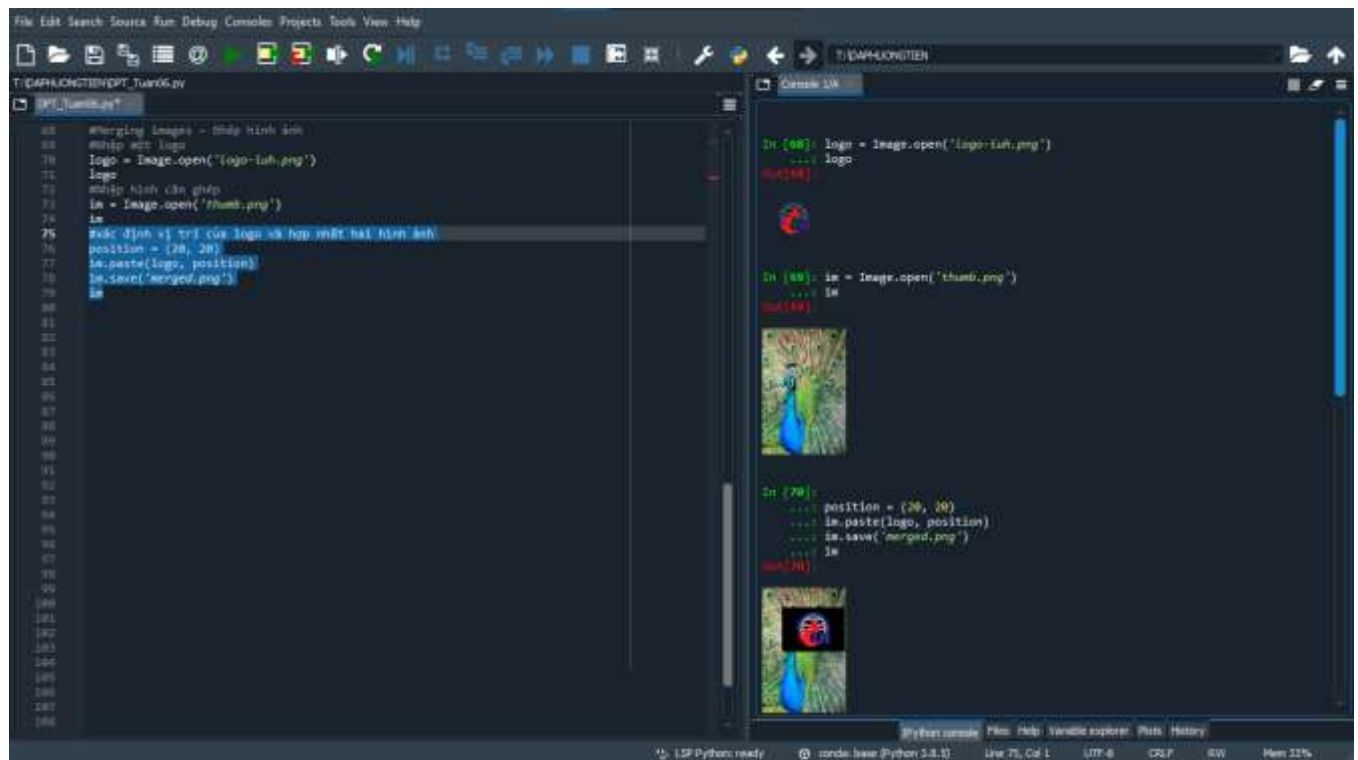


- **Merging images – Ghép hình ảnh:** Gói Pillow này cũng cho phép chúng ta hợp nhất hai hình ảnh. Hãy minh họa điều đó bằng cách hợp nhất một logo vào hình ảnh con công. Chúng ta bắt đầu bằng cách nhập nó.

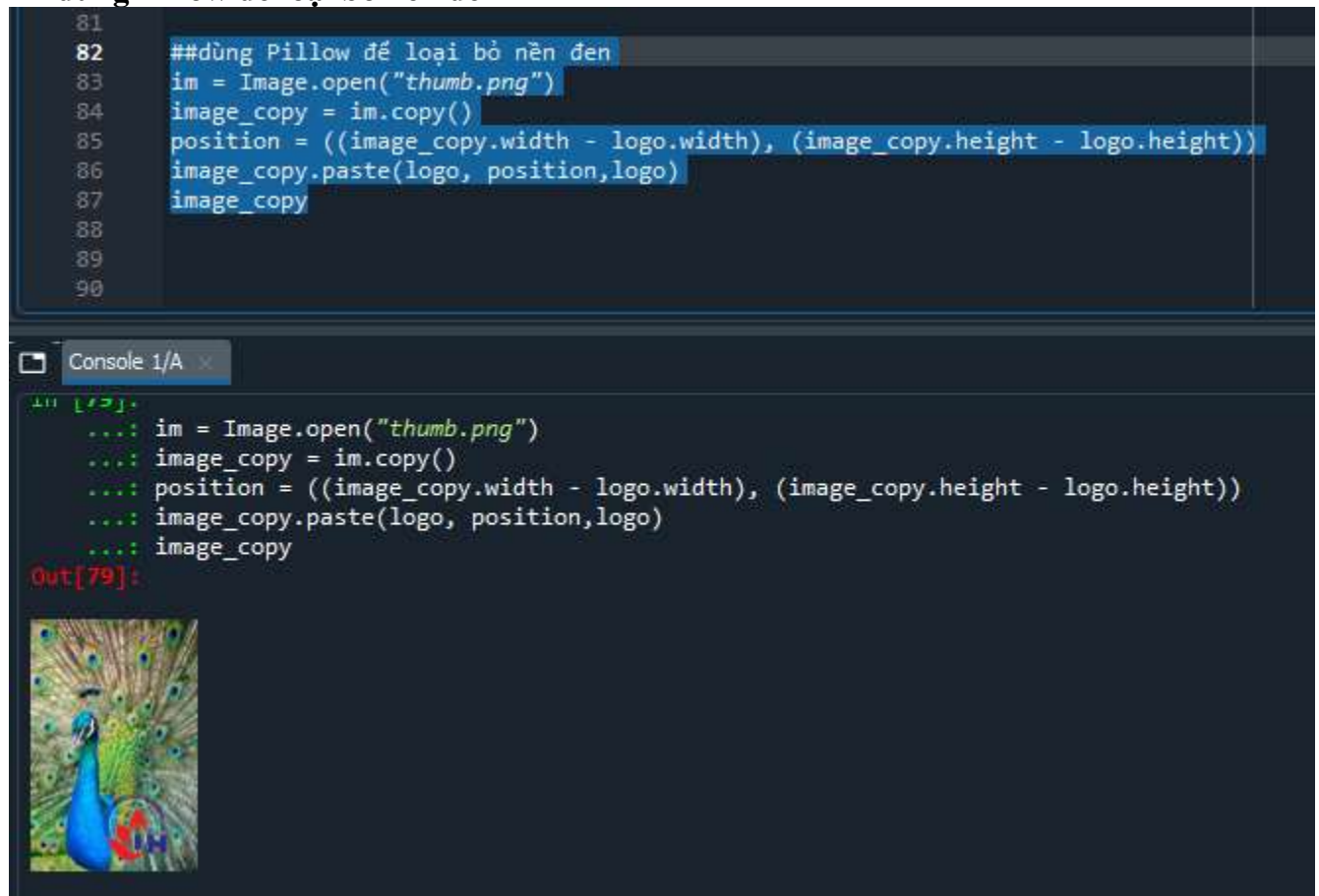
```
75 #xác định vị trí của logo và hợp nhất hai hình ảnh
76 position = (20, 20)
77 im.paste(logo, position)
78 im.save('merged.png')
79 im
```

```
In [70]:
...: position = (20, 20)
...: im.paste(logo, position)
...: im.save('merged.png')
...: im
Out[70]:
```







➤ dùng Pillow để loại bỏ nền đen



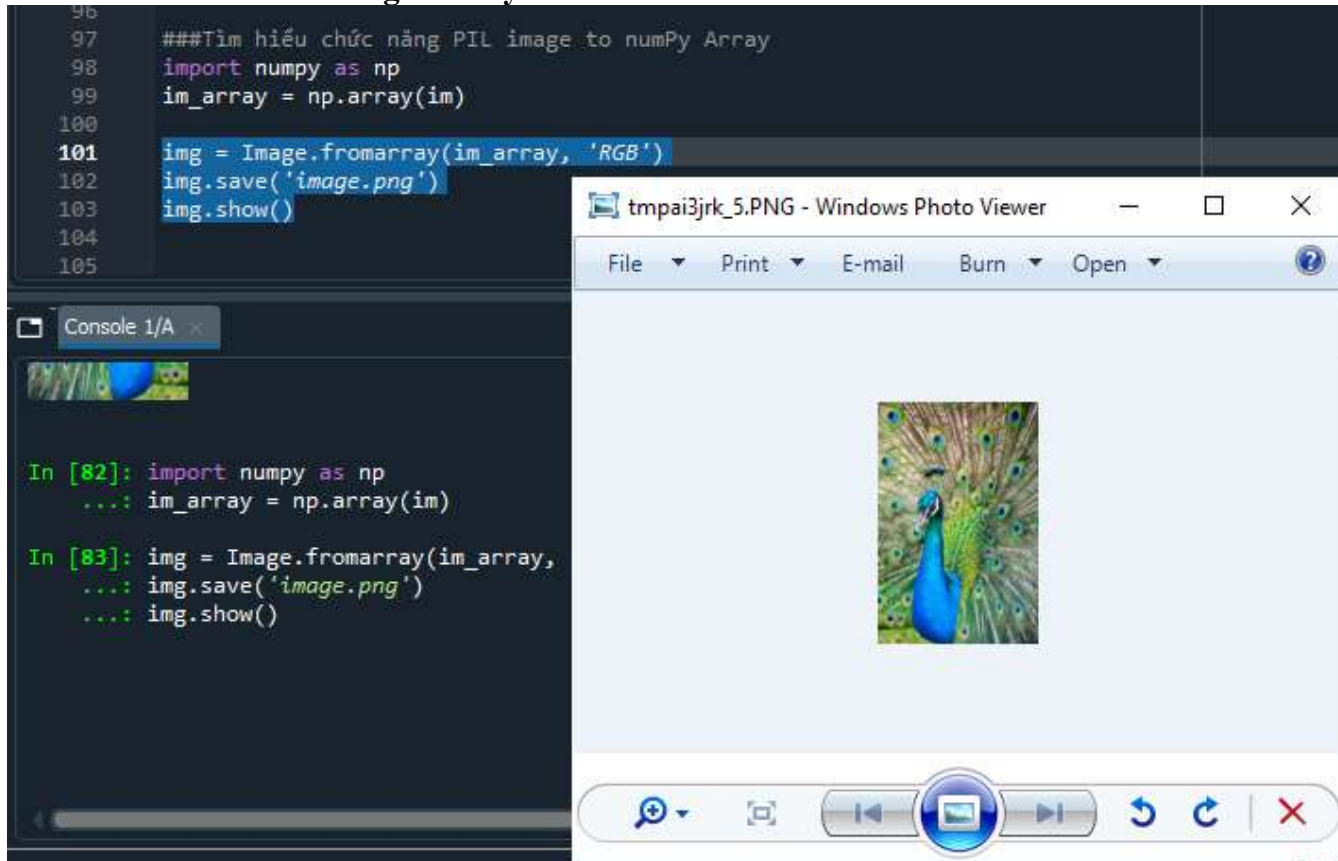
- **Flip images - Lật hình ảnh:** điều này được thực hiện bằng phương pháp "Lật". Một số tùy chọn lật là FLIP_TOP_BOTTOM và FLIP_LEFT_RIGHT.

```
T:\DAPHUONGTIEN\DPT_Tuan06.py
DPT_Tuan06.py
90  ###Flip images - Lật hình ảnh
91  #FLIP_TOP_BOTTOM
92  im.transpose(Image.FLIP_TOP_BOTTOM)
93
94  #FLIP_LEFT_RIGHT
95  im.transpose(Image.FLIP_LEFT_RIGHT)
96
```

```
Console 1/A
In [80]: im.transpose(Image.FLIP_TOP_BOTTOM)
Out[80]:

In [81]:
...: im.transpose(Image.FLIP_LEFT_RIGHT)
Out[81]:

```

➤ **Tìm hiểu chức năng PIL image to numPy Array**

○ **Hình ảnh PIL thành mảng NumPy**



➤ **Tìm hiểu chức năng Color transformations**


- Color transformations – biến đổi màu sắc: chúng ta có thể chuyển đổi hình ảnh từ màu sang đen trắng và ngược lại. Điều này được thực hiện thông qua hàm chuyển đổi và truyền định dạng màu ưa thích.


```
104
105     ###Tìm hiểu chức năng Color transformations
106     im.convert('L')
107
```

Console 1/A

In [84]: im.convert('L')

Out[84]:



-
- Chuyển sang màu sắc:

```
104
105     ###Tìm hiểu chức năng Color transformations
106     #chuyển trắng đen
107     im.convert('L')
108
109     #chuyển màu sắc
110     im.convert('RGBA')
111
112
113
```

Console 1/A



In [87]:

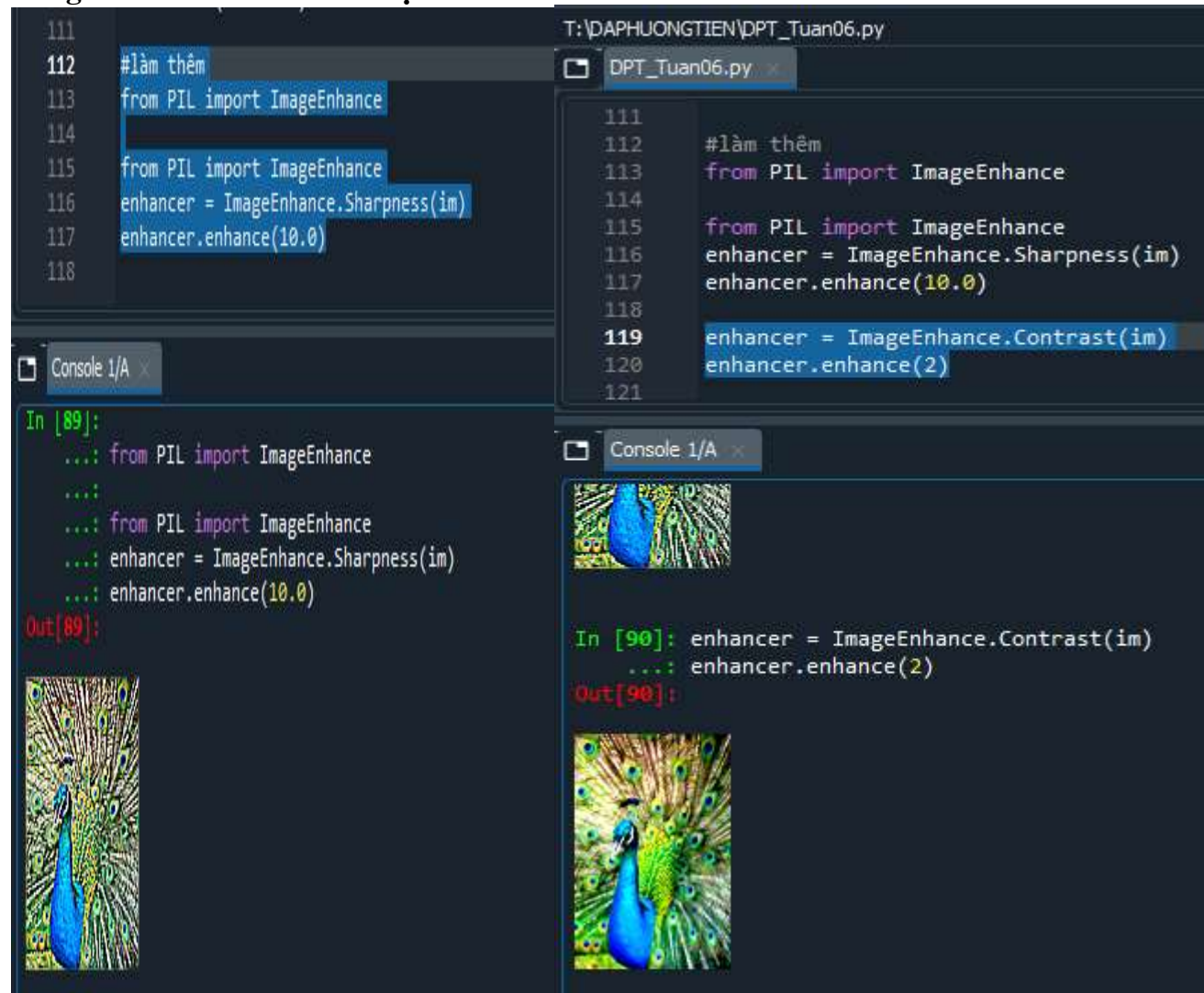
...: im.convert('RGBA')

Out[87]:



SINH VIÊN LÀM THÊM

Image enhancement – cải thiện hình ảnh



The screenshot displays a Jupyter Notebook interface with two code cells and their corresponding outputs. The top code cell, labeled 'DPT_Tuan06.py', contains Python code for image enhancement using PIL's ImageEnhance module. The bottom code cell, labeled 'Console 1/A', shows the execution of the code and the resulting enhanced images.

```
111  
112 #làm thêm  
113 from PIL import ImageEnhance  
114  
115 from PIL import ImageEnhance  
116 enhancer = ImageEnhance.Sharpness(im)  
117 enhancer.enhance(10.0)  
118
```

```
111  
112 #làm thêm  
113 from PIL import ImageEnhance  
114  
115 from PIL import ImageEnhance  
116 enhancer = ImageEnhance.Sharpness(im)  
117 enhancer.enhance(10.0)  
118  
119 enhancer = ImageEnhance.Contrast(im)  
120 enhancer.enhance(2)  
121
```

The output of the first code cell shows the execution of the code, including the import of ImageEnhance and the application of the Sharpness enhancer. The output of the second code cell shows the execution of the code, including the application of the Contrast enhancer. Below the code cells, two images of a peacock are displayed side-by-side, showing the result of the image enhancement process.



Filters – Bộ lọc blur

```
127
128     #fillter - bộ lọc
129
130     from PIL import ImageFilter
131     im = Image.open("thumb.png")
132     im.filter(ImageFilter.BLUR)
133     im
```

Console 1/A

```
In [94]: from PIL import ImageFilter
...: im = Image.open("thumb.png")
...: im.filter(ImageFilter.BLUR)
...: im
Out[94]:
```

