

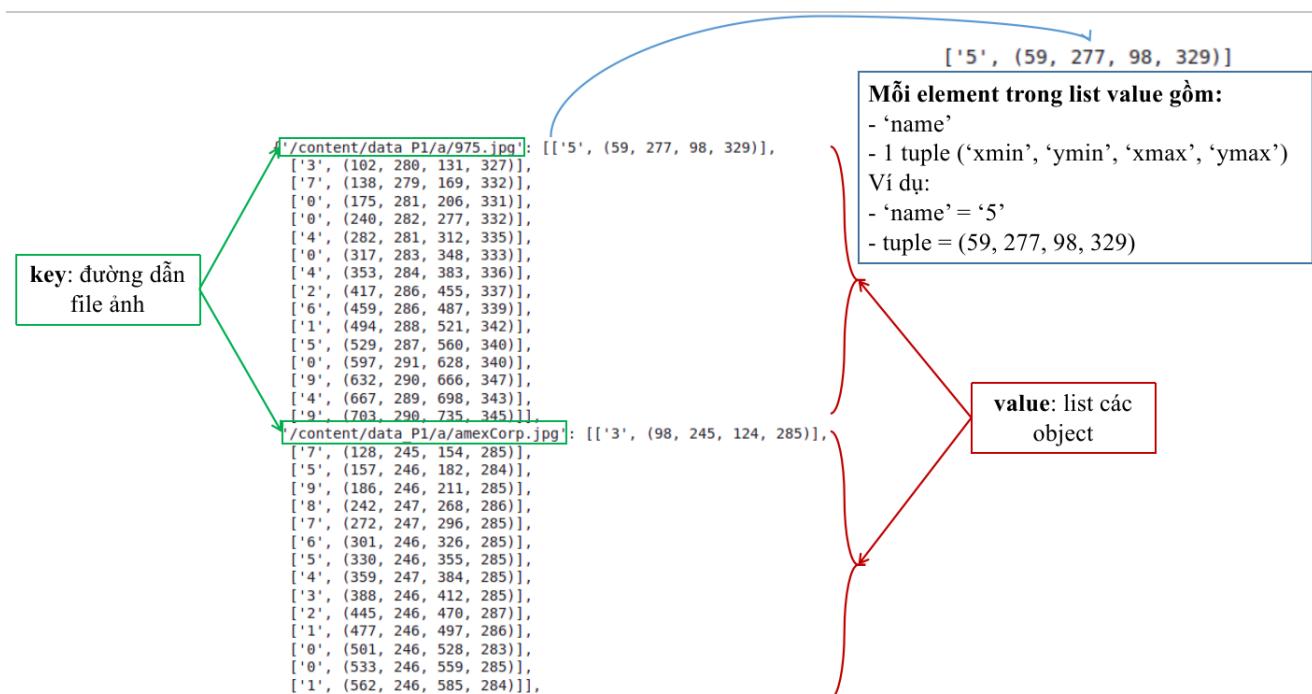
# AI VIET NAM – COURSE 2022

## Advanced Python 2 – Exercise

Ngày 26 tháng 6 năm 2022

1. Viết các hàm đọc và lấy thông tin từ tất cả các file xml hoặc json trong folder theo yêu cầu sau: (một số configuration và data đã được định nghĩa trước và đính kèm vào link file notebook bên dưới)

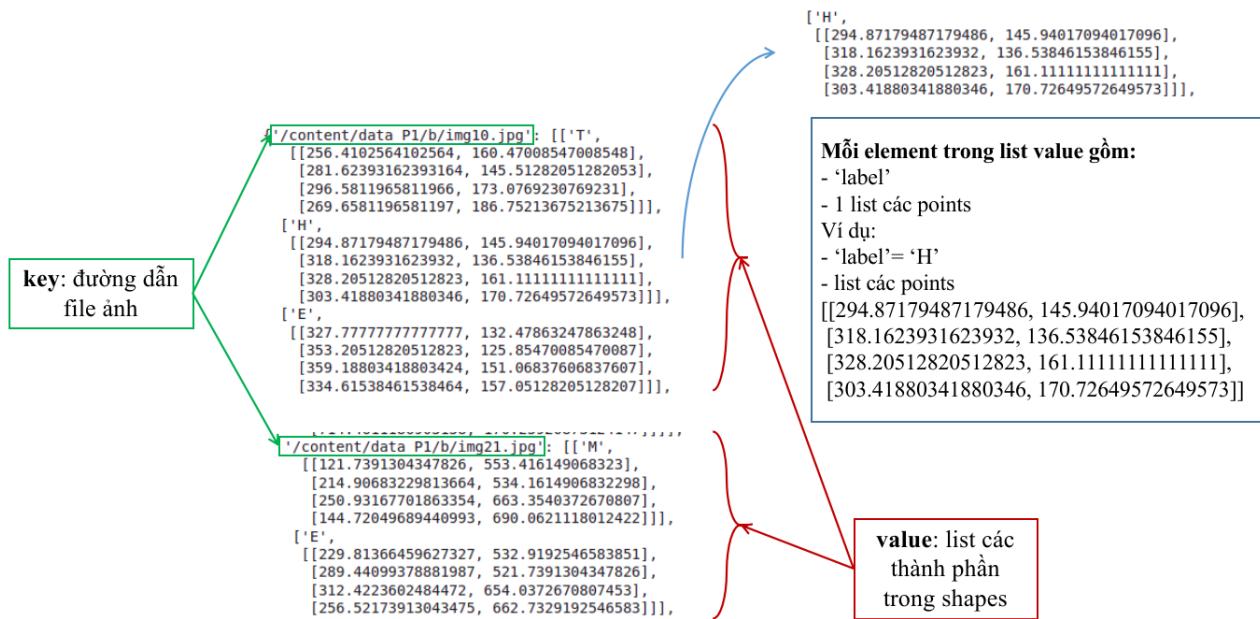
- (a) Viết hàm `readxml(path)` đọc toàn bộ file xml và lấy các thông tin sau "filename", và thông tin của các "object" bao gồm "name", "xmin", "ymin", "xmax", "ymax" và trả về một dictionary với mỗi item gồm key là tên đường dẫn ảnh, value là một list chứa thông tin của các "object"
- **Input:** path đường dẫn file chứa ảnh (.jpg) và file xml (.xml).
  - **Output:** một dictionary
    - key: đường dẫn đến file ảnh
    - value: list với mỗi element là 'name' và tuple(xmin, ymin, xmax, ymax)



Hình 1: Output 1a

- (b) Viết hàm `readjson(path)` đọc toàn bộ file json và lấy các thông tin sau "imagePath", và thông tin của "shapes" bao gồm "label", và "points". Sau đó trả về một dictionary với mỗi item gồm key là tên đường dẫn ảnh, value là một list chứa thông tin của các "shapes"

- Input:** Đường dẫn file chứa ảnh (.jpg) và file json (.json).
- Output:** một dictionary
  - key: path đường dẫn đến file ảnh
  - value: list với mỗi element là 'label' và list points



Hình 2: Output 1b

### Hướng dẫn cách làm bài:

- Toàn bộ bài tập sẽ thực hiện theo template notebook này ([link](#))

<b>▼ Exercise 1a</b> <pre> 1 ##### Your Code Here ##### 2 def readxml(path): 3     pass </pre>	<b>▼ Exercise 1b</b> <pre> 1 ##### Your Code Here ##### 2 def readjson(path): 3     pass </pre>
--	---

Hình 3: Cell làm bài tập 1a và 1b

- Các bạn vào teamplate notebook ở trên, chạy hết các cell ở phần có tên là Exercise1 để download các file, data và giải nén data, cũng như import các thư viện cần thiết.
- Tiếp theo các bạn tìm đến các cell như hình 3 ở trên để hoàn thành bài tập
- Các bạn phải đọc nội dung các file trong path1a, và path1b** để biết folder chứa cả ảnh và file xml hoặc json. Đồng thời đọc nội dung các file để biết cấu trúc và vị trí các thông tin mà mình cần lấy theo yêu cầu của đề .
- Sau đó viết nội dung cho hàm readxml (nhận input path1a) và readjson (nhận input path1b) và return về dictionary kết quả

- Đối với key của dictionary là đường dẫn đến file ảnh (file ảnh và file xml hoặc json sẽ cùng tên nhưng khác đuôi mở rộng .jpg, .xml hoặc .json) các bạn sẽ phải gắn đường dẫn đến folder chứa ảnh và tên file ảnh. Tên file ảnh chính là "filename" (1a) và "imagePath" (1b). Để thu được đường dẫn đến ảnh cũng chính là key
- **lưu ý 1a:** xmin, ymin, xmax, ymax khi lấy ra sẽ có dạng string các bạn nhớ ép kiểu về dạng int
- **lưu ý 2a:** các bạn lấy trực tiếp "points" là list và không cần xử lý gì thêm.
- Sau khi có được dictionary, các bạn convert sang iterator và dùng hàm có sẵn (draw\_one\_img với is\_from\_xml=True (1a) và is\_from\_xml=False (1b)) để vẽ các kết quả lên ảnh như ví dụ bên dưới



Hình 4: Kết quả 1a



Hình 5: Kết quả 1b

```

1 # Examples 1a
2 res = readxml(path1a)
3 it = iter(res.items())
4
5 img_path, data = next(it)
6 draw_one_img(img_path=img_path,
7               data=data,
8               is_from_xml=True)
9 ...
10 # Only 5 times
11 img_path, data = next(it)

```

```

12 draw_one_img(img_path=img_path,
13             data=data,
14             is_from_xml=True)
15
16 # Examples 1b
17 res2 = readjson(path1b)
18 it2 = iter(res2.items())
19
20 img_path, data = next(it2)
21 draw_one_img(img_path=img_path,
22               data=data,
23               is_from_xml=False)
24
25 ...
26 # Only 5 times
27 img_path, data = next(it2)
28 draw_one_img(img_path=img_path,
29               data=data,
30               is_from_xml=False)

```

2. Bài toán này cung cấp một class gồm có một method `__getJSONResponse` để search tên một cuốn sách trên google và trả về 5 kết quả đầu tiên theo kiểu json. Bạn hãy thực hiện 3 method còn lại để:

- (a) Hoàn thiện method `exportJSONResult` để có thể xuất ra được file json kết quả sau khi search.

- **Input:** query=tên sách, json\_path=đường dẫn lưu file json kết quả.
- **Ouput:** file json kết quả tìm kiếm cuốn sách được lưu theo đường dẫn.
- **Lưu ý:** khi write file json dùng `json.dump()` với parameter `indent = 4` để dễ nhìn nếu kết quả khó nhìn có thể sử dụng <https://codebeautify.org/jsonviewer> để dễ nhìn kết quả.
- Tham khảo file **TTGBCT\_1.json** và **aaa\_1.json**

- (b) Quan sát file json đã được save, để viết method `parseJSON` lấy các thông tin như sau:

- totalItems: số lượng search được từ query
- Trong một list các sách trả về, chỉ cần lấy 3 cuốn sách đầu tiên gồm các thông tin sau (GIA TRỊ NÀO KHÔNG KIỂM ĐƯỢC THÌ TRẢ VỀ STRING 'Unknow'):
  - title
  - authors
  - publisher
  - publishedDate
  - description
  - buyLink
  - textToSpeechPermission
  - pdf
- Thêm 1 cặp key-value, với key là "query" và value là tên cuốn sách bạn tìm kiếm vào dictionary output
- **Input:** là đường dẫn file json được ghi xuống từ function của câu 2a ở trên
- **Ouput:** là dictionary với các thông tin muốn lấy
- Tham khảo file **TTGBCT\_2.json** và **aaa\_2.json**

- (c) Viết method `exportXMLFinalResult` và Dùng kết quả từ `parseJSON` tạo thành 1 file xml tương ứng.

- Nếu kết quả trả về có key 'booksInfo' là list rỗng (trường hợp không tìm thấy tên sách) thì không cần tạo element này
- **Input:** json\_path là đường dẫn file json được ghi xuống từ method của câu 2a ở trên, và xml\_path là đường dẫn ghi file xml
- **Output:** là file xml có format như **TTGBCT\_3.xml** và **aaa\_3.xml**
- **Lưu ý:** đối với 'booksInfo' là list chứa thông tin từng cuốn sách lấy được, thì mỗi element sẽ tạo ra element <book> </book> trong xml với attribute id= 0, 1, 2 theo thứ tự trong list của dicitonary. Nếu 'booksInfo' là list rỗng thì bỏ qua không cần làm gì hết.
- **Lưu ý:** authors là list nên trong xml sẽ join lại tên các tác giả và cách nhau bằng dấu ','

### Hướng dẫn cách làm bài:

- Toàn bộ bài tập sẽ thực hiện theo template notebook này ([link](#)) mục Exercise 2

```

20     def exportJSONResult(self, query, json_path='./book.json'):
21         self.query = query
22         response = self.__getJSONResponse()
23         # 2a
24         ##### Your Code Here #####
25
26
27     def parseJSON(self, json_path=""):
28         # Sử dụng giá trị UNKNOW_VAL khi không kiêm được thông tin cần tìm.
29         # Ví dụ không kiêm được "authors" thi cho value = UNKNOW_VAL
30         UNKNOW_VAL = 'Unknow'
31         # Các bạn có thể sử dụng dictionary mâu này là dict trả về:
32         # "booksInfo" là list và mỗi element là các thông tin các cuốn sách kiêm được
33         # (ở đây chỉ yêu cầu tối đa là 3 cuốn sách)
34         dict_res = {
35             'query': self.query,
36             'totalItems': 0,
37             'booksInfo': []
38         }
39         # 2b
40         ##### Your Code Here #####
41
42     def exportXMLFinalResult(self, json_path="", xml_path ""):
43         # 2c
44         ##### Your Code Here #####

```

Hình 6: Cell làm bài tập 2

- Các bạn tìm đến cell Exercise 2 sẽ có một cell code mẫu như hình 6 các bạn thực hiện các câu 2a, 2b và 2c. Các bạn nhớ đọc hàm \_\_init\_\_ và \_\_getJSONResponse để nắm được idea. Có thể tùy ý thêm attribute cho hàm \_\_init\_\_
- **Các file mẫu "TTGBCT\_x" và "aaa\_x" ở [link](#)**
- **LƯU Ý:** FILE CÓ TÊN "TTGBCT" LÀ CÁC FILE CÓ TÊN BOOK TÌM THÀNH CÔNG, CÒN "aaa" LÀ CÁC FILE CÓ TÊN BOOK KHÔNG TÌM ĐƯỢC
- **CÓ THỂ TÙY Ý CHỈNH SỬA CODE ĐƯỢC ĐƯA TRƯỚC, CHỈ CẦN RA KẾT QUẢ NHƯ YÊU CẦU ĐỀ BÀI**
- Đây là file kết quả khi chạy 2 ví dụ của listing code bên dưới [link](#)

```

1 # Examples
2 # Find an existing book
3 search_books = SearchingGoogleBook()
4 search_books.exportJSONResult('Tôi tài giỏi bạn cũng thế', 'TTGBCT.json')

```

```

5 search_books.parseJSON('TTGBCT.json')
6 search_books.exportXMLFinalResult('/content/TTGBCT.json', 'TTGBCT.xml')
7
8
9 # Find non-existing book
10 search_books = SearchingGoogleBook()
11 search_books.exportJSONResult(
12     'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa', 'aaa.json')
12 search_books.parseJSON('aaa.json')
13 search_books.exportXMLFinalResult('aaa.json', 'aaa.xml')

```

### 3. Thực hiện xây dựng một hàm có chức năng compensate focal length cho một cặp image với focal length khác nhau

- **Input:** Hai ảnh có cùng kích thước nhưng khác nhau về focal length như hình 9
- **Output:** Hai ảnh, một ảnh được chụp với long focal length camera (ảnh trái) hình 14 và một ảnh được chụp với short focal length camera đã được compensate như hình 14 (ảnh phải)

#### Hướng dẫn cách làm bài:

- Toàn bộ bài tập sẽ thực hiện theo template notebook này ([link](#)) mục Exercise 3

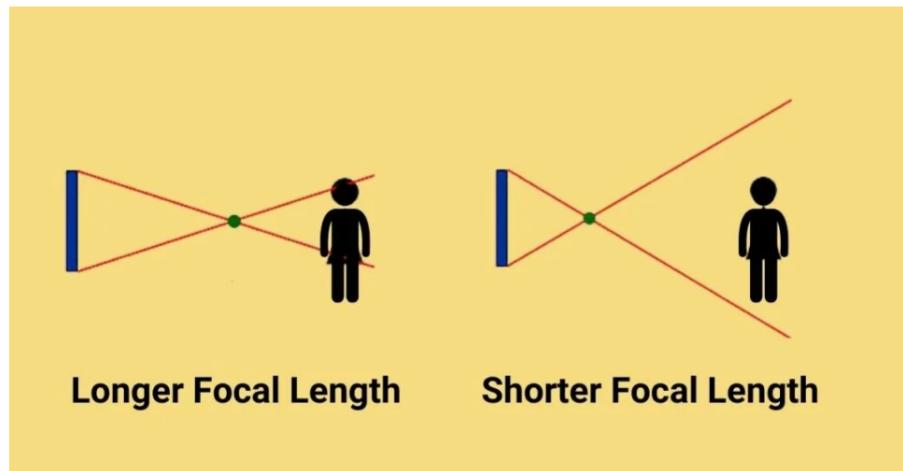


Hình 7: Cell làm bài tập 3

- Các bạn vào teamplate notebook ở trên, chạy hết các cell ở phần có tên là Exercise 3 để download các file, ảnh cũng như import các thư viện và function mẫu cần thiết.
- Các bước để compensate focal length cho ảnh được chụp với short focal length như sau
  - **Step 1:** Tìm keypoints của cả hai ảnh input (keypoint là đường tròn trong hình 10 (hàng trên))
  - **Step 2:** Map các keypoints tương ứng với nhau giữa hai ảnh (hình 10 (hàng dưới))
  - Step 1 và Step 2 đã được code mẫu trong link notebook trên. Các bạn dùng hàm `get_pair_keypoints` (input đường dẫn hai ảnh input, và output bao gồm hai ảnh đã input vào và hai list keypoints của hai ảnh đã được map với nhau (hai list gồm các element có cùng index với nhau thì là keypoint tương ứng với nhau))
  - **Step 3** các bạn viết function `compensate` sử dụng hàm `get_pair_keypoints` lấy hai ảnh và hai list keypoints . Sau đó tìm inner-distance giữa các keypoints của từng ảnh bằng Euclidean distance. Có nghĩa là trong mỗi list keypoint tìm Eulidean distance một điểm với tất cả các điểm còn lại. Thu được 2 list inner-distance, tìm ratio của các inner-distance và tính giá trị trung bình (được gọi là scale) như hình 11
    - \* Euclidean distance:  $d(p1, p2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
  - **Step 4** sử dụng scale thu được ở Step 3 resize ảnh Short Focal Length bằng ảnh mới có `width*scale` và `height*scale` như hình 12. Thu được ảnh Resized Short Focal Length by scale
  - **Step 5:** Crop ảnh Resized Short Focal Length by scale như hình 13. Tìm tọa độ `x_start`, `y_start`, bằng cách lấy 1 keypoint bất kỳ ở ảnh Long Focal Length và tìm keypoint tương ứng của keypoint ở ảnh Short Focal Length (**cùng index giữa hai keypoint list**). Lấy

keypoint ở ảnh Short Focal Length nhân với scale. Lấy x và y ở điểm vừa thu được trừ cho x và y của keypoint trong ảnh Long Focal Length và đó chính là x\_start và y\_start. Đối với x\_end, y\_end thì cộng một lượng tương ứng width và height của ảnh Long Focal Length

- **LƯU Ý CÁC BẠN CHỈ CẦN TRẢ VỀ ẢNH LONG FOCAL LENGTH VÀ COMPENSATED SHORT FOCAL LENGTH (ẢNH SAU KHI ĐƯỢC COMPENSATE FOCAL LENGTH). KHÔNG CẦN PHẢI VẼ HÌNH COMAPRE NHU HÌNH 14**



Hình 8: Long focal length và short focal length



Hình 9: Input ảnh trái và phải khác nhau về focal length



Hình 10: Các keypoints của hai ảnh input(hàng trên) và map keypoints tương ứng giữa hai ảnh (hàng dưới)



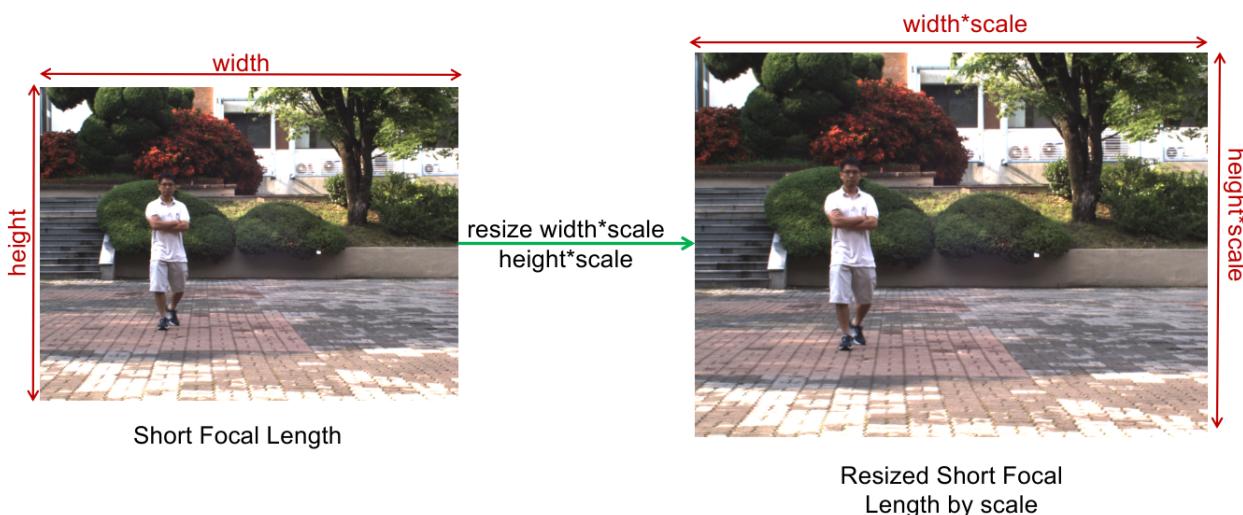
Euclidean distance:  $d(p1, p2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Inner-distance: tìm Euclidean distance từ một point đến tất cả các point trong cùng một ảnh. Lặp hết tất cả các point

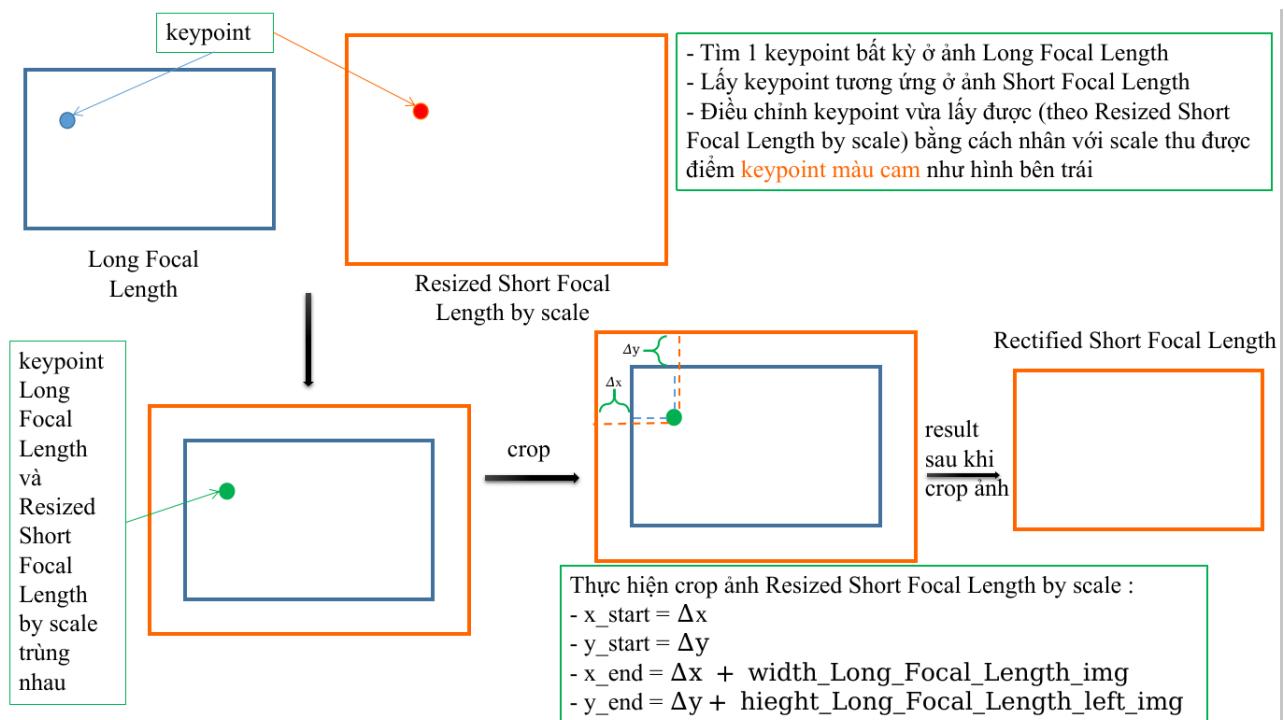
$$\text{inner-distance\_list 1} \quad \text{inner-distance\_list 1} \\ \text{inner-distance\_list 2} \quad \text{inner-distance\_list 2} \\ \frac{\text{inner-distance\_list 1}}{\text{inner-distance\_list 2}} = \text{ratio\_inner-distance\_list}$$

$$\text{scale} = \text{mean}(\text{ratio\_inner-distance\_list})$$

Hình 11: Tìm scale bằng cách tính trung bình ratio của inner-distance giữa các point trong cùng ảnh. (Hình chỉ minh họa một điểm đến một vài point. Trong thực tế phải tính tất cả các point)



Hình 12: Resize ảnh Short Focal Length thành ảnh mới có width\*scale và height\*scale



Hình 13: Crop ảnh Resized Short Focal Length by scale thu được ảnh output cuối cùng có cùng size với ảnh Long Focal Length)



Hình 14: Ảnh thể hiện kết quả sau khi xử lý vấn đề focal length khác nhau giữa hai ảnh. So sánh ảnh long focal length (ảnh trái) và output short focal length sau khi được compensate (ảnh phải)