Multiscale Template Matching: [3]

Như ta đã đoán được từ trước, template matching chỉ có thể tìm một vùng cụ thể trong một bức ảnh lớn và đưa ra nhãn của vùng đó. Tuy nhiên nếu như hai bước ảnh tuy có kích thước khác nhau, nhưng vẫn dung chung một hình dáng template ta có thể tìm được vùng thông qua phiên bản multiscale:



ở bức hình bên trái: kích thước ảnh là 724x 1024

còn bức hình bên phải: có kích thước là: 1275x 1650

( Có thể trong tờ báo này, ta không thấy được size gốc của hai bản nhạc, nhưng nếu mọi người lên github của chúng tôi, thì có thể trực mắt kiểm định phần này )

Về cơ bản, tất cả các feature của sheet bên phải không thể dùng cho sheet bên trái và ngược lại nếu như không có một vài kỹ thuật liên quan đến multiscale feature matching.

Phương pháp giải quyết:

Ta sẽ chạy for loop từ 0.2 -> 2 biểu thị cho hệ số size hình sẽ resize. Trước hết, ta resized hình rồi template matching với template. Sau đó ta lưu kết quả lớn nhất khi template matching trả về. Và cuối cùng mỗi vòng for, ta lưu lại giá trị lớn nhất đó để so sánh với các kết quả sau này. Cụ thể là phần code như sau:

savedScale =0

found = None

for scale in np.linspace(0.2, 20, 20):

imageResize = img.resize(scale)

result = cv2.matchTemplate(template, imageResize) [1]

maxVal = cv2.minMaxLoc(result) [2]

if found is None or maxVal> found:

found = maxVal

savedScale = scale

Mục đích của phần code trên là giúp tìm ra được hệ số scale phù hợp cho template để áp dụng vào hình. Và bước tiếp theo là sẽ resized hình theo hệ số scale đã tìm được và dung nó để template matching với template.

resizedImg = img.resized(savedScale)

startLoc = cv2.matchTemplate(template, resizedImg)

# startLoc sẽ trả về cặp tọa độ (x,y) của template có trong hình.

Tại đây mỗi template ta đã có 2 cặp tọa độ [ startLoc, endLoc], endLoc = startLoc + template.shape

startLoc

endLoc

Bước cuối cùng là trả về tọa độ của bức hình gốc: Bức hình của mình được resized ở một hệ số savedScale nên ta có cặp tọa độ startLoc và endLoc được biến đổi thành:

startLoc\_new = startLoc \* 1/savedScale

endLoc\_new = endLoc \* 1/savedScale.



Result.

References:

[1] : <https://docs.opencv.org/4.5.2/d4/dc6/tutorial_py_template_matching.html>

[2] : <https://docs.opencv.org/4.5.2/de/da9/tutorial_template_matching.html>

[3] : <https://www.pyimagesearch.com/2015/01/26/multi-scale-template-matching-using-python-opencv/>