

# BÀI THU HOẠCH

## CHỦ ĐỀ: IMAGE SEGMENTATION

### A. Thông tin cá nhân:

MSSV: 19127273

Huỳnh Thị Mỹ Thanh

### B. Bài thu hoạch:

#### *I. Problem statement*

- Input: image
- Output: segmentation map (chia ảnh thành nhiều phần hoặc nhiều vùng thuộc về cùng 1 lớp) hoặc là 1 ảnh chứa các thông tin về classification, localization, segmentation..

#### *II. Challenge*

- Độ phức tạp của đối tượng: hình dạng phức tạp => phân loại các pixel hoặc vùng ảnh trở nên khó khăn hơn
- Số lượng đối tượng: có quá nhiều đối tượng thì phải đảm bảo không có đối tượng nào bị bỏ sót.
- Sự che khuất: đối tượng bị che khuất thì việc segment trở nên khó khăn hơn.
- Sự phân biệt giữa đối tượng và nền: đối tượng và nền có màu sắc tương tự nhau là một thách thức.

#### *III. Methods*

##### **\*PP truyền thống:**

##### 1. Region – based:

Phân vùng (gom những pixel cùng đặc tính như cùng ánh sáng, màu sắc..)

Để phân đoạn ảnh theo vùng cần có những luật mà mình sẽ quy định sẵn, được sd khi ảnh cần phân đoạn là những ảnh nhiễu.

Region – based bao gồm:

- Region growing (lan tỏa vùng): chọn 1 seed pixel (có thể random), khi chạy thì mình sẽ cho nó 1 cái luật bằng cách đặt 1 threshold (ngưỡng), có nghĩa là những điểm lân cận của seed pixel sẽ tuân thủ theo 1 ngưỡng nào đó, nếu lớn hơn ngưỡng đó thì gom vào 1 vùng, nhỏ hơn thì gom vào 1 vùng.

- Splitting and merging: xét toàn bộ 1 ảnh, cũng đặt ra cho nó những luật được định sẵn. Giả sử mình đặt rule đó là q, với điều kiện như sau:
  - $z_{\max} - z_{\min} \leq \text{threshold}$  đk để gộp
  - $z_{\max} - z_{\min} > \text{threshold}$  – điều kiện để tách

## 2. Clustering:

Phương pháp học máy không giám sát, chia image thành những cụm khác nhau

Thường sử dụng K-means clustering, cách thực hiện:

- Chọn ra điểm centroid tùy ý (random), sau khi chọn dc thì tính khoảng cách (euclid) tìm khoảng cách nhỏ nhất, điểm nào gần với centroid nhất thì gom lại thành 1 vùng.
- Sau đó tính toán lại các centroid bằng cluster mean => lặp lại n lần cho đến khi centroid không đổi => tất cả điểm dữ liệu đã được gom cụm.

## 3. Phân đoạn dựa trên edge detection

Dựa vào việc tìm biên cạnh của ảnh bằng cách dùng các kernel (sobel , canny, prewitt..).

## 4. Thresholding:

Từ 1 bức ảnh màu hoặc grayscale biến nó thành binary có 3 loại (global, local và adaptive)

**\*PP hiện nay:**

## 5. Deep learning

### *5.1 Semantic segmentation:*

#### **Sliding windows idea:**

Định nghĩa ra cửa sổ nhỏ => trượt qua toàn bộ ảnh, phân lớp điểm central của nó thuộc class nào.

Phương pháp này khá mất thời gian vì phải trượt cửa sổ qua toàn bộ ảnh, để khắc phục, ý tưởng mới sau đây ra đời.

**Sử dụng convolution trên cả ảnh:** thay vì sử dụng tích chập trên từng cửa sổ trượt, ta sẽ thực hiện tích chập trên cả ảnh => phân lớp.

Tuy nhiên, idea này sẽ xuất hiện vấn đề càng tích chập kích thước ảnh càng nhỏ (mất thông tin) , để khắc phục lại có 1 ý tưởng mới như sau.

**Fully connected:** không giảm kích thước lớp convolution phía sau thay vào đó là fully connected => rơi vào vấn đề số lượng conv rất lớn vì kích thước ảnh lớn.

Cách khắc phục:

### Down sampling, up sampling idea:

Mạng của nó chia thành 2 nửa: nửa đầu thực hiện phép tích chập thông thường để classify (kích thước vẫn giảm qua các lớp conv), sau đó mình up sample nó lên để output trở về dc kích thước ban đầu

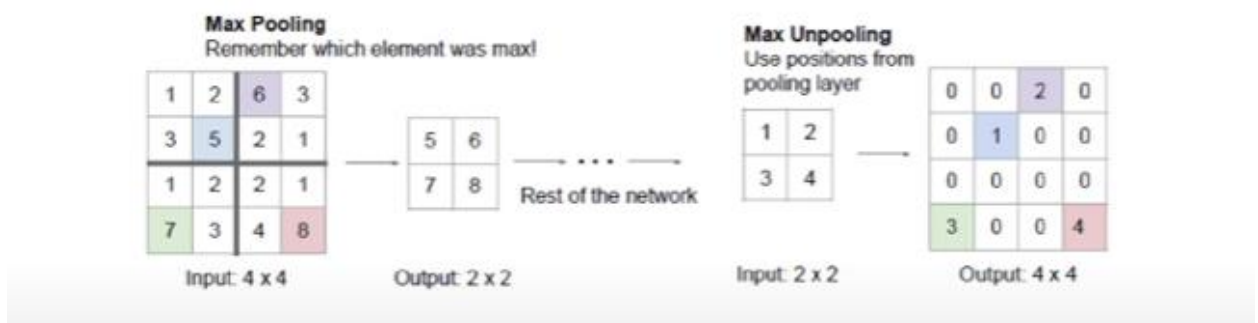
Xuất hiện vấn đề: Làm sao từ feature nhỏ sau nửa đầu để up sampling lên 1 tập lớn để đưa ra output lớn như ban đầu?

=> Có 3 yếu tố để thực hiện upsampling (nửa đầu ra heatmap => unpooling để trả về kích thước):

1. Convolution: vai trò để trích xuất đặc trưng.
2. Pooling: giảm kích thước feature map => giảm số lượng phép toán qua từng layer. Có 2 loại: Max pooling (với từng kernel tìm cái có giá trị lớn nhất) và avg (tính trung bình).
3. Unpooling: mục đích từ (2x2) => (4x4)

Để thực hiện được có 2 phương pháp cơ bản sau:

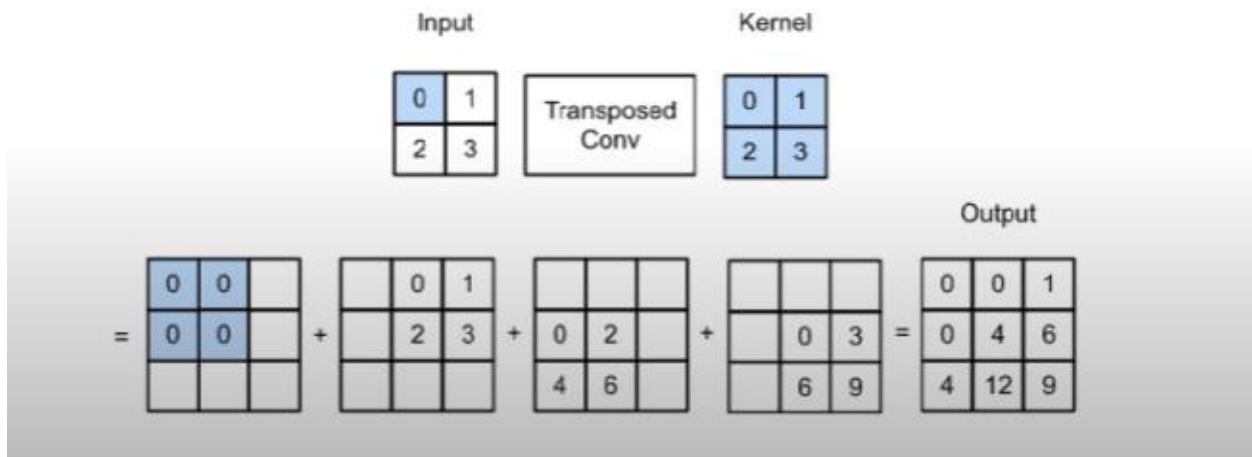
- Nearest neighbor: gán giá trị của input cho các điểm lân cận ở output, số lượng lân cận quyết định dựa vào tỉ lệ output với input.
- Bed of nails: dựa vào tỉ lệ input output để chia thành những vùng, mỗi vùng tương ứng sẽ lấy thẳng input tương ứng gán vào góc trái của vùng, những điểm còn lại sẽ bằng 0.
- Cải tiến của bed of nails: Mỗi lần mình down mỗi lần nó sẽ giữ lại thông tin về vị trí như ảnh minh họa sau:



- Ngoài ra, còn có 1 phương pháp khác là transposed convolution:

Với 1 ảnh input (2x2) => (3x3)

Có 1 cái input và kernel, duyệt qua từng pixel trong input như ảnh minh họa sau



#### Sử dụng Model: unet

Chia mạng đó ra thành 2 nửa, nhánh đầu sd các mô hình trích xuất đặc nhưng như vgg, gg net (mạng conv bình thường) sau đó nó sẽ up sampling (thêm 1 bước skip )

#### 5.2 Instance segmentation:

Output: gán nhãn từng object

Kết hợp giữa object detect + image segmentation

#### Mask RCNN

Tận dụng lại kiến trúc của RCNN để đưa ra được từng box của object nó sẽ áp thêm một mask prediction (mạng convolution để segment).

#### IV. Conclusion

Tổng kết lại, image segmentation là một quá trình quan trọng trong xử lý ảnh và nhận dạng đối tượng. Việc phân loại các pixel hoặc vùng ảnh trong hình ảnh giúp chúng ta trích xuất thông tin hữu ích và giải quyết nhiều vấn đề trong nhiều lĩnh vực khác nhau như y tế, tự động hóa, an ninh, và nhận dạng khuôn mặt. Tuy nhiên, việc thực hiện image

segmentation đòi hỏi một số thách thức như độ phức tạp của đối tượng, số lượng đối tượng, sự che khuất, sự phân biệt giữa đối tượng và nền, và tốc độ xử lý. Vì vậy, việc áp dụng các phương pháp và công nghệ mới nhất để giải quyết các thách thức này là rất cần thiết để đạt được kết quả tốt nhất trong image segmentation.