**BÀI TOÁN TRÍCH XUẤT GAN, U GAN TRÊN MRI 3D**

1. **Dữ liệu:**

Dữ liệu MRI được chia sẻ từ thầy Phạm Thế Bảo, dữ liệu này đã từng được sử dụng bởi thầy Lê Trọng Ngọc trong luận án cách đây 4 năm trước, cũng đề tài tương tự.

Nguồn của bộ dữ liệu này được lấy từ các trung tâm y tế cả trong và ngoài nước.

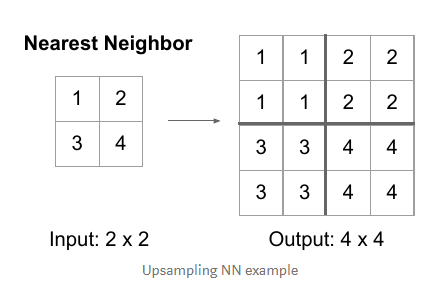
Bao gồm: 27 bộ tương ứng với 27 bệnh nhân khác nhau. Trong đó 16 bộ được chụp bằng công nghệ MRI 1.5T (Hệ thống: Sigma HDxt/HDx; Achieva, Philips medical systems, Cleveland,OH; andGEmedical systems, Milwaukee, WI) từ đại học trung tâm y tế Chicago. Và 11 bộ còn lại được chụp tại Việt Nam bằng công nghệ MRI (Avanto, Siemens). Bộ dữ liệu được nhận từ thầy Bảo đủ 27 bộ, trung bình mỗi bộ có số slice từ 54 đến 56, kích thước ma trận hay kích thước ảnh là 320x230 pixel.

Các phần thông tin khác của dữ liệu được nêu trong bài báo [1]. Vì hiện tại kiến thức của bọn em không đủ để sử dụng hết tất cả các thông tin trên dữ liệu nên em chỉ nêu những thứ em biết.

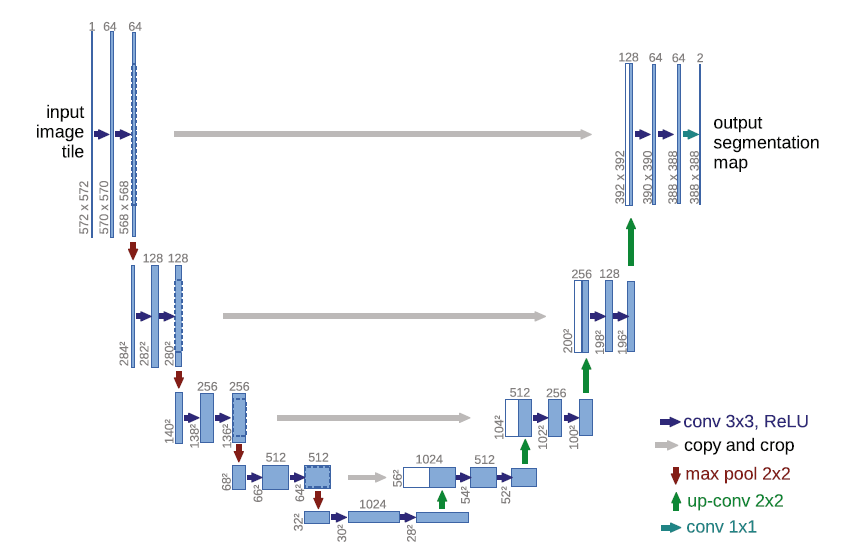
1. **Phương pháp tiếp cận:**
   1. **Các vấn đề:**
      1. Nhiễu từ các cơ quan nội tạng khác có cùng mức xám/ cùng mức năng lượng/ cùng độ tương phản với gan.
      2. Mức độ tương phản (intensity) của MRI trên vùng gan không đồng đều hoặc không cao.
      3. Imbalance data với u gan.
   2. **Phương pháp xử lý dữ liệu:**
      1. Cắt bớt các nội tạng dư thừa (Chỉ áp dụng trên tập dữ liệu train, để huấn luyện model): dựa vào ground truth có sẵn, lấy giá trị max, min của dòng và cột trên mỗi slice, để tính chiều dài và chiều rộng của gan, sau đó set up một khung hình tối thiểu (min-width,min-height), dựa vào khung hình này để cho ra kết quả tương ứng, nếu kích thước gan lớn hơn khung hình tối thiểu thì phải padding ảnh ra sao cho kích thước ảnh sau khi cắt bằng với kích thước khung hình tối thiểu. Việc set up khung hình tối thiểu để tránh trường hợp bị vỡ khung của ground truth sau khi cắt (các ground truth khi bị vỡ các giá trị 1 sẽ loang ra khắp nơi khiến vị trí khối u, gan không đúng).
      2. Trong thời gian đầu nghiên cứu, có sử dụng thuật toán N4ITK để xử lý intensity của MRI, nhưng sau đó không nắm chắc được kiến thức nên đã loại bỏ phương án này (mặc dù vẫn áp dụng thực tế nhưng khống viết trong báo cáo nghiên cứu khoa học).
      3. Với u gan, phải trích xuất gan trước để lấy vùng gan, loại bỏ các background không cần thiết, chỉ giữ lấy background chứa gan, loại bỏ được nhiều độ nhiễu từ các cơ quan nội tạng khác. Để xử lý imbalance trước khi train model, nhóm nghiên cứu đã loại bỏ toàn bộ tất cả các slice không chứa gan trên tập train, sau đó dùng các phương pháp augmentation: thêm noise (gaussian), flip, rotate, kể cả dùng lại phương pháp cắt gan trong phương án 1.
      4. Thêm metric cho cả 2 model gan, u gan: VOE (volume overlap error, mean IOU, RVD).
   3. **Thuật toán FCN – kiến trúc Unet 2D:**

Để xử lý bài toán segmentation, nhóm nghiên cứu đã tìm hiểu về thuật toán FCN trong deep learning, sau đó nhóm tìm hiểu thêm về kiến trúc Unet 2D vì đây là kiến trúc cơ bản và nổi tiếng trong các bài toán segmentation liên quan đến ảnh y tế.

Về thuật toán FCN, cấu trúc cơ bản sẽ như các thuật toán CNN thông thường gồm các lớp convolution và các lớp max pooling. Tuy nhiên lớp cuối cùng Fully connected sẽ được thay bằng lớp up-convolution. Ba kiến trúc FCN được giới thiệu đầu tiên là: FCN-32s, FCN-16s, FCN-8s [2]. Ký hiệu –ns (với n=8,16,32) trong mỗi tên của kiến trúc tương ứng với tham số stride của lớp up-convolution. Kích thước ảnh sẽ được nhân lên tương ứng với số stride được set up. Kích thước nhân lên bằng phương pháp duplicate nearest neighbor (hình 1)



Hình 1: Mô tả hoạt động của lớp up-convolution. Kích thước đầu vào là 2 x 2. Với stride = (2 x 2), lớp up-convolution sẽ lặp 4 lần mỗi đặc trưng của ảnh đầu vào và cho ra kết quả với kích thước 4 x 4.

Kiến trúc Unet (hình 5) được xây dựng dựa trên thuật toán FCN, tuy nhiên kiến trúc sẽ được chia làm 2 phần: contracting, expansive. Đặc điểm chung của hai phẩn là đều có cùng số lớp convolution và số kernel. Phần contracting sẽ nằm bên trái của kiển trúc và có cấu trúc như các mô hình CNN thông thường, với các lớp convolution và các lớp pooling. Phần expansive sẽ nằm bên phải và thay các lớp pooling thành các lớp up-convolution. Tại mỗi bước của phần contracting, các kích thước width – height của ảnh sẽ giảm và tăng depth. Ngược lại, phẩn expansive sẽ tăng kích thước width-height của ảnh và giảm depth. Ngoài ra kiến trúc còn có phần concatenate để kết hợp các vùng thông tin của 2 phần contracting và expansive với nhau để cho ra kết quả tốt nhất.

Hình 5: Kiến trúc U-net được giới thiệu bởi tác giả Ronnerberger[3]. Các khối màu xanh miêu tả đầu ra của mỗi bước, với số kernel được ký hiệu phía trên và kích thước dữ liệu được ký hiệu phía dưới bên trái mỗi khối. Các mũi tên được quy định và ký hiệu cho mỗi lớp khác nhau.

Tuy nhiên, để cho ra kết quả tốt nhất, nhóm nghiên cứu có tinh chỉnh mô hình để phù hợp với dữ liệu, bao gồm:

Thêm các lớp batch normalization sau mỗi lớp convolution, thêm dropout (dropout rate = 0.5).

Batch Normalization (BN): Tạo hiện tượng Zero-mean với độ lệch chuẩn là 1, giúp cho các feature phân phối đều với nhau, giúp tối ưu các mạng neural với số layer sâu. Từ đó ta có thể tăng learning-rate cao hơn trong quá trình training. Ngoài ra BN giúp cho model không rơi vào trạng thái bão hòa mỗi khi đi qua hàm kích hoạt. Trạng thái bão hòa là khi giá trị đầu vào thay đổi liên tục nhưng giá trị đầu ra vẫn giữ nguyên. Thêm nữa BN có tác dụng như một dạng của regularization giúp giảm hiện tượng overfiting.

Dropout: Có tác dụng như regularization, ngăn chặn overfiting bằng cách thêm yếu tố “phạt” vào loss function.

Các tham số mô hình Unet mà nhóm nghiên cứu sử dụng:

* Input shape: (n,256,256,1).
* Convolutions: Depth/Kernels (contracting: 32,64,128,256; expansive: 256,128,64,32; deepest layer: 512), kernel size = 3 x 3.
* Max-pool: stride = 2 x 2
* Up-convo: stride = 2 x 2
* Learning rate: 1e – 6 (trong trường hợp không tinh chỉnh model), 1e-4(Trường hợp có tinh chỉnh model).
* Activation: ReLu.
* Optimizer: Adam.