Report paper: U-Net: Convolutional Networks for Biomedical Image Segmentation

Giáp Ngọc Hiệu

**Tóm tắt nội dung**

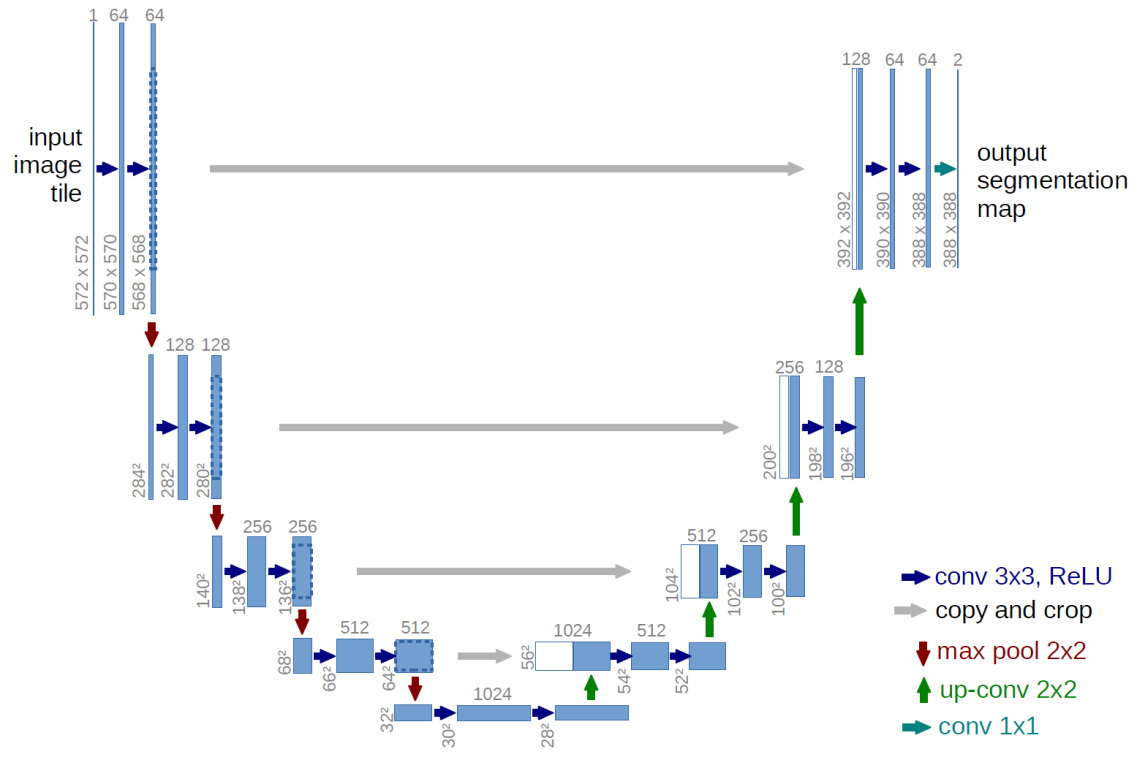
Bài report này là phần tóm tắt lại ý tưởng, nội dung chính, những điều làm được của paper

U-Net: Convolutional Networks for Biomedical Image Segmentation

1. **Ý tưởng**

Thông thường, để đào tạo thành công các mạng sâu cần phải có lượng lớn dữ liệu được gán nhãn. Nhưng trong lĩnh vực y khoa, nguồn dữ liệu được cung cấp khá hạn chế. Tác giả đã đề xuất một mạng mới gọi là U-Net, một mạng thần kinh tích chập được phát triển để phân đoạn hình ảnh y khoa. Kiến trúc bao gồm một phần được gọi là contracting path (encoder) để nắm bắt ngữ cảnh và một phần gọi là expanding path (decoder) cho phép định vị chính xác. Cùng với đó, tác giả cũng thực hiện tăng cường dữ liệu từ dữ liệu có sẵn để việc đào tạo diễn ra hiệu quả hơn. U-Net có thể được đào tạo end-to-end với rất ít hình ảnh và vượt trội so với các phương pháp tốt nhất trước đó.

1. **Phương pháp**
   1. **Network Architecture**



**Contracting path (Encoder)**

Phần này bao gồm các 3x3 Conv layers và 2x2 MaxPooling thông thường. Đi từ trên xuống dưới width x height giảm, còn depth tăng. Mục đích để nắm bắt các đặc trưng quan trọng và giảm kích thước của feature map cũng như thời gian tính toán

**Expanding path (Decoder)**

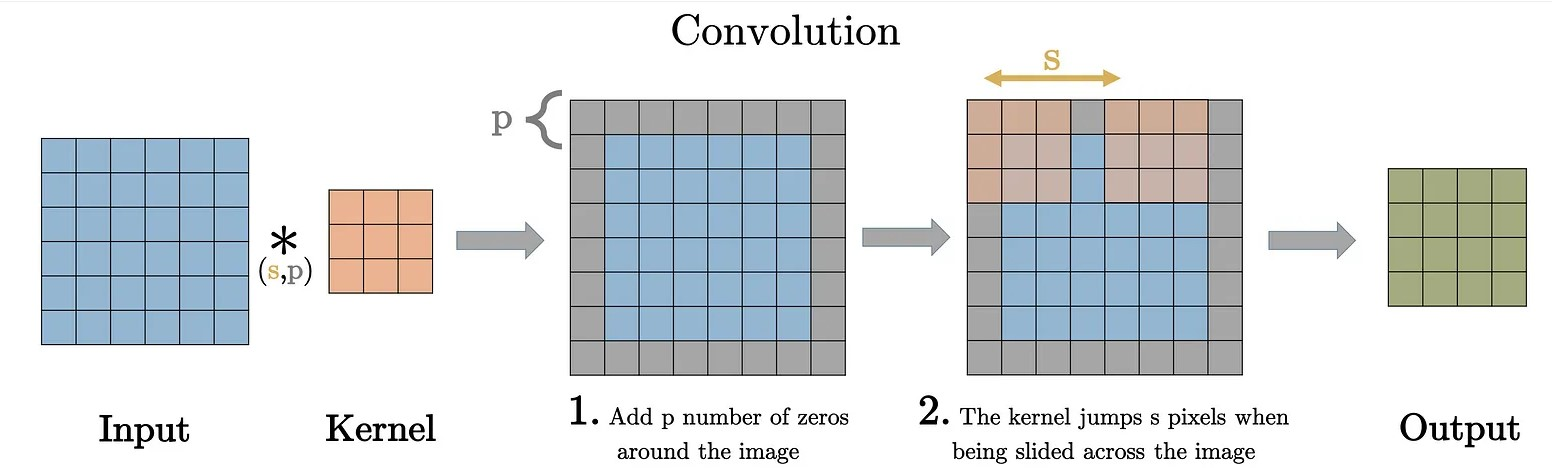
Phần này ngược lại với encoder, làm tăng width x height và giảm depth bằng cách áp dụng transposed convolution. Mỗi giai đoạn của decoder lại lấy layer phía đối xứng của encoder crop rồi concatenate lại (cách kết nối này tương tự với cách kết nối trong một số kiến trúc như ResNet hay DenseNet giúp cải thiện performance của model).

Nếu ảnh đầu vào là ảnh màu có kích thước h x w x 3 thì số channels đầu tiên là 3. Ở phần decoder chỗ áp dụng Conv layer 1x1, số filters chính là số classes cần segment. Output sẽ có dạng h x w classes. Nếu lấy argmax theo chiều của channels chúng ta sẽ biết được từng pixel thuộc về class nào. Nhóm các pixel chung class vào một nhóm và gán màu cho chúng. Từ đây ta có thể tạo mask cho ảnh ban đầu với các phần segment của ảnh ban đầu.

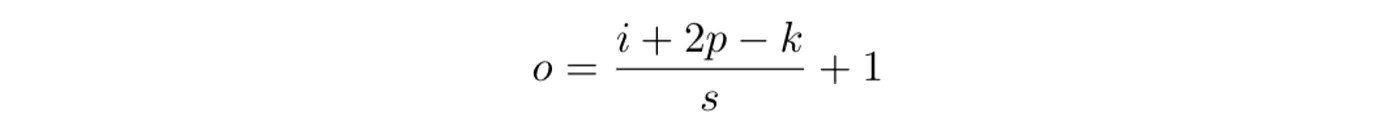
Lí do sử dụng skip connection ở đây là nếu chỉ sử dụng feature map ở phía bên dưới để xây dựng lại feature map có spatial dimension cao hơn thì rất nhiều thông tin trên spatial dimension bị mất. Chính việc thêm skip connection ở phần encoder chúng ta sẽ tái tạo lại vị trí của các pixels tốt hơn.

**Standard Convolution & Transposed Convolution**

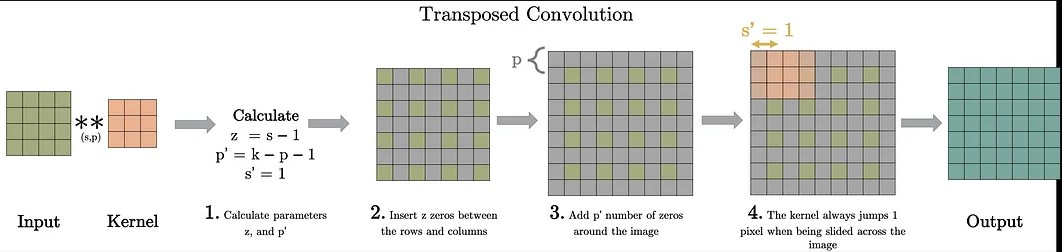
**\* Standard Convolution**



Với size của input (i), kernel (k), padding (p), và stride (s), size của feature map (o) được tạo ra tính bằng:



**\* Transposed Convolution**



Có thể hiểu hoạt động của transposed convolution bao gồm 4 bước:

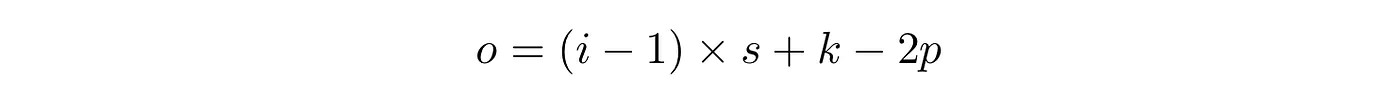
Bước 1: Tính z và p’

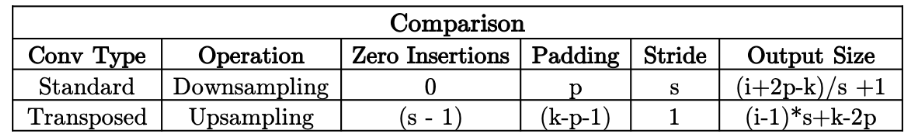
Bước 2: Giữa mỗi hàng và cột của đầu vào, chèn z số 0. Điều này làm tăng kích thước của đầu vào ( 2 \* i - 1 ) x ( 2 \* i - 1 )

Bước 3: Đặt hình ảnh đầu vào đã sửa đổi với số lượng số 0 p’

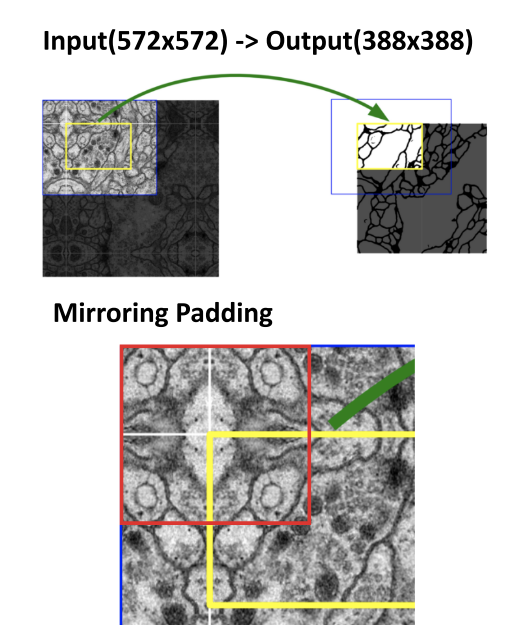
Bước 4: Thực hiện tích chập bình thường trên hình ảnh được tạo từ bước 3 với stride bằng 1.

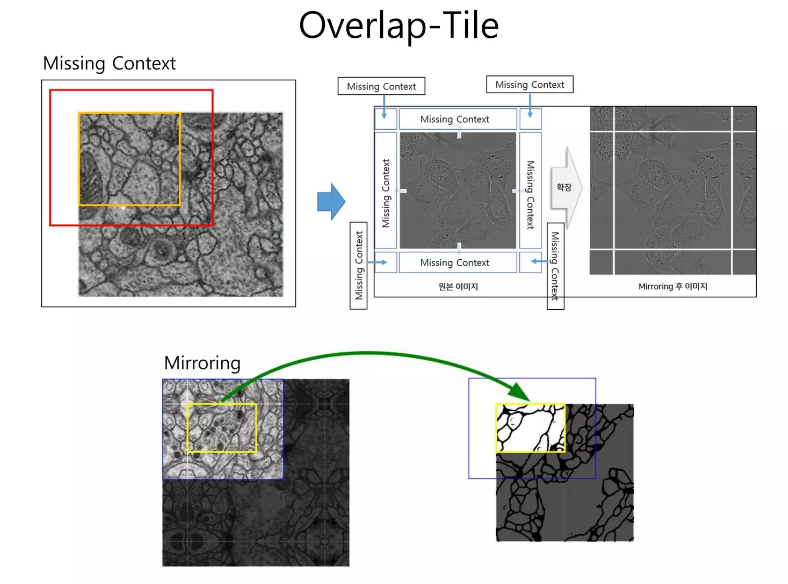
Size của output được tính như sau:



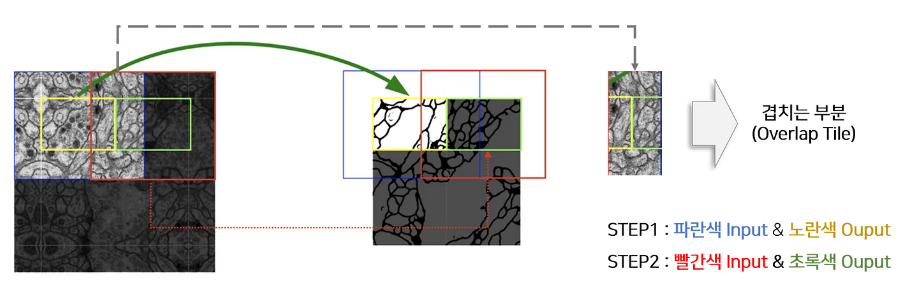


* 1. **Overlap-tile strategy**

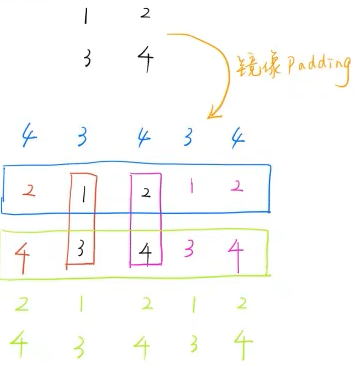




Với "overlap tile strategy", ta chia hình ảnh gốc thành các phần nhỏ hơn được gọi là "tiles" có kích thước mà mạng U-Net có thể xử lý được. Ta chia các tiles này thành các vùng giao nhau, cho phép chúng overlap lên nhau. Điều này giúp đảm bảo rằng các phần biên của các tiles cũng được xử lý và không bị bỏ sót.



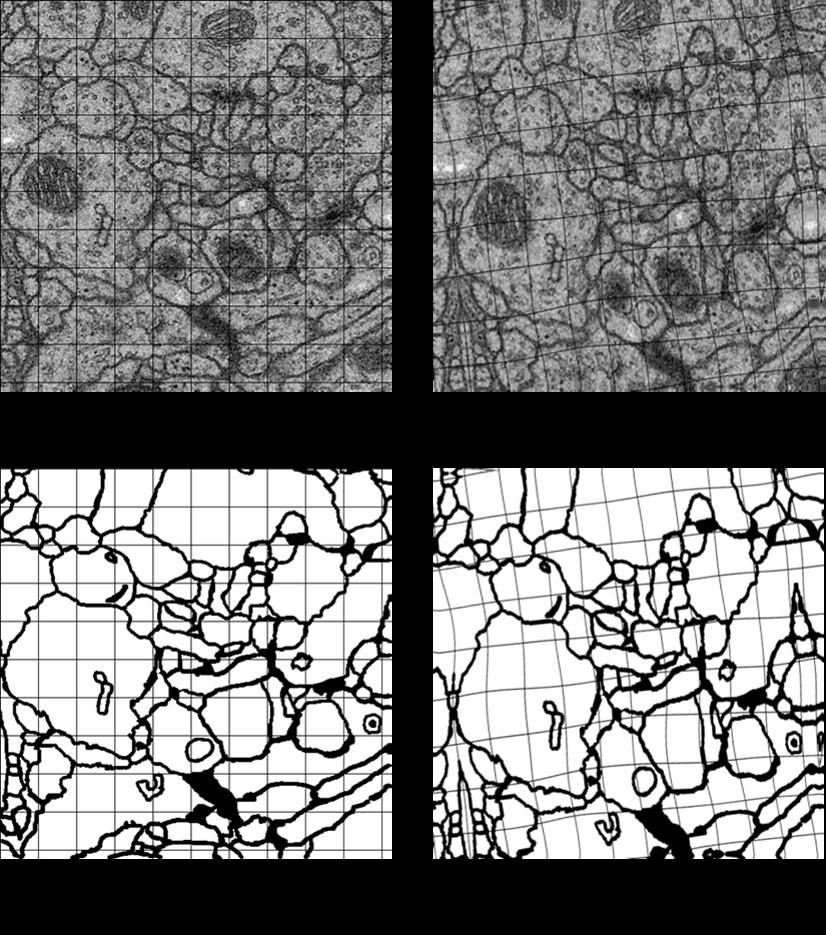
Vì trong U-Net, kích thước của đầu vào lớn hơn đầu ra nên ta cần lấy đầu vào lớn hơn bằng cách sử dụng reflect padding ở các phần cạnh của ảnh.



Sau khi thu được các kết quả dự đoán cho từng tile, ta kết hợp các kết quả dự đoán từ các tiles để tái tạo kết quả cho toàn bộ hình ảnh gốc.

Theo tác giả, overlap tile strategy rất quan trọng để áp dụng mạng cho hình ảnh lớn mà không cần quá nhiều tài nguyên tính toán.

* 1. **Elastic Deformation for Data Augmentation**



Tác giả tạo biến dạng mềm bằng các vector dịch chuyển ngẫu nhiên trên lưới 3x3. Các giá trị của các vector dịch chuyển được lấy mẫu từ một phân phối Gaussian với độ lệch chuẩn là 10 pixel. Điều này đảm bảo rằng các biến đổi sẽ là những biến đổi nhỏ, không gây ra biến dạng quá lớn. Để tính toán giá trị dịch chuyển tại mỗi pixel trên hình ảnh, tác giả sử dụng phép nội suy bicubic.

Ngoài ra, tác giả còn sử dụng Dropout ở phần encoder để mạng có thể học được các đặc trưng tốt hơn cũng như tổng quát hóa.

* 1. **The loss function**

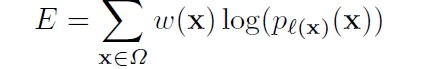
Đầu ra của mạng được tính toán bằng pixel-wise soft-max trên feature map cuối cùng, sau đó loss được thực hiện với hàm cross entropy.

**Softmax function**

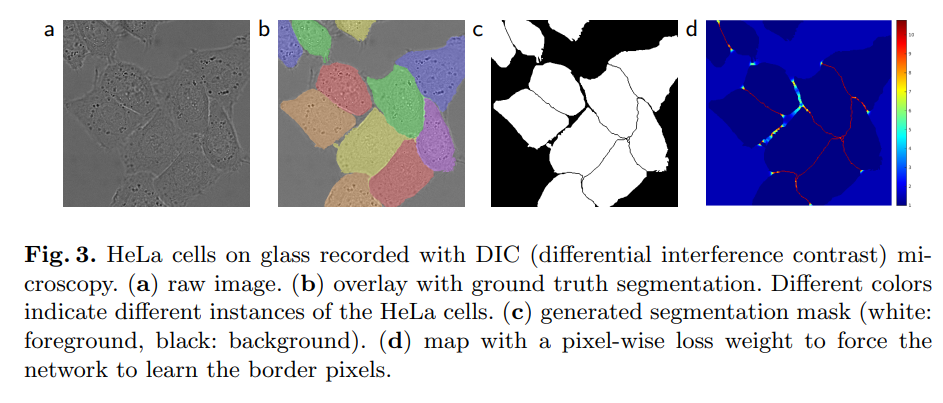


trong đó ak(x) biểu thị kích hoạt trong kênh đặc trưng k tại vị trí pixel x ∈ Ω, K là số lớp.

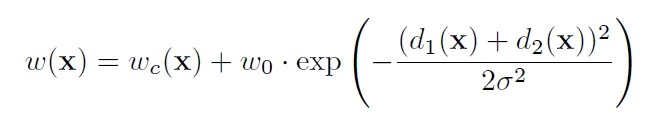
**Cross entropy loss function**



trong đó l : Ω → {1, . . . , K} là nhãn thực của từng pixel và w : Ω → R là weight map. Vì để tách các đối tượng tiếp xúc của cùng một lớp, tác giả đề xuất gán trọng số cho các pixel. Các nhãn nền phân tách giữa các ô chạm nhau sẽ có được trọng số lớn trong hàm loss. Điều này làm tăng tầm quan trọng của một số pixel trong quá trình đào tạo.



**Weight map**



trong đó wc : Ω → R là weight map để cân bằng tần suất lớp, d1 : Ω → R biểu thị khoảng cách đến đường viền của ô gần nhất và d2 : Ω → R là khoảng cách đến đường viền của ô gần thứ hai. Trong các thử nghiệm của mình, tác giả đặt w0 = 10 và σ ≈ 5 pixel.

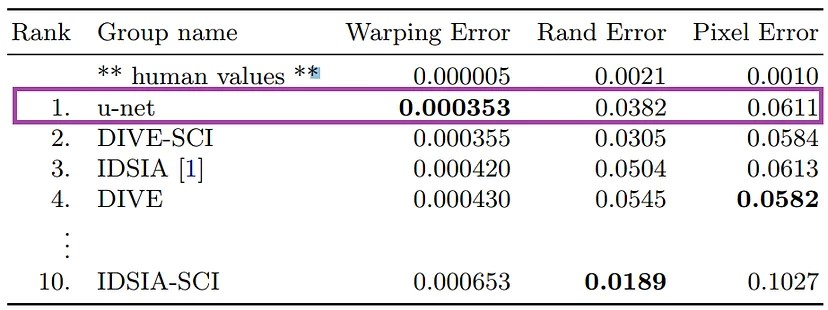
Ngoài ra, tác giả còn thực hiện khởi tạo trọng số trong mạng bằng cách khởi tạo các trọng số ban đầu từ phân phối Gaussian với độ lệch chuẩn là √2/N , trong đó N biểu thị số node đến của một nơ-ron. Ví dụ. đối với tích chập 3x3 và 64 kênh đặc trưng ở lớp trước

N = 9 \* 64 = 576

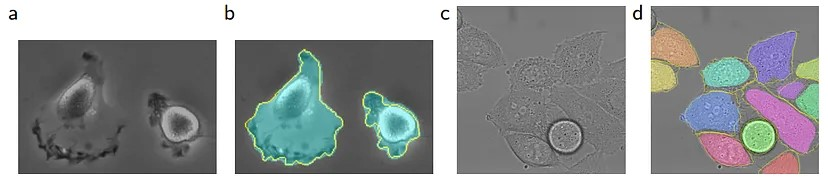
Mục đích là các trọng số ban đầu nên được điều chỉnh sao cho mỗi feature map trong mạng có phương sai xấp xỉ 1. Tránh các phần của mạng có thể kích hoạt quá mức, trong khi các phần khác không bao giờ đóng góp, đảm bảo mức đóng góp của các phần trong mạng là như nhau.

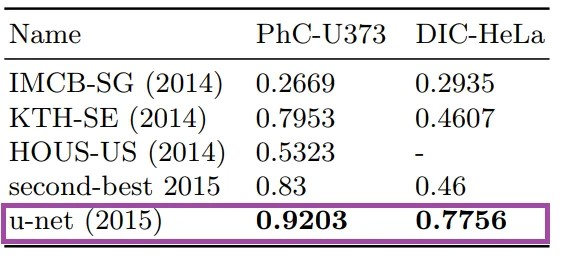
1. **Kết quả**

Hiệu suất của U-Net thực hiện trên phân đoạn ảnh EM, so với các phương pháp khác:



Kết quả từ bộ dữ liệu PhC-U373 và DIC-HeLa và so sánh với các nghiên cứu trước đây:





1. **Note**

Ưu điểm của U-Net:

+ Hiệu suất cao

+ Xử lý tốt các tác vụ phân đoạn ảnh nhiều lớp, vì nó có thể xử lý một số lượng lớn các lớp và tạo ra bản đồ phân đoạn mức pixel cho mỗi lớp.

+ Sử dụng hiệu quả dữ liệu đào tạo: U-Net sử dụng các skip connection, cho phép mô hình kết hợp các tính năng cấp cao và cấp thấp từ hình ảnh đầu vào.

Nhược điểm:

+ Một số lượng lớn các tham số: U-Net có nhiều tham số do skip connection và các layer ở phần decoder.

+ Có thể khó khăn trong việc xử lí các hình ảnh có kích thước lớn

- Phân đoạn ngữ nghĩa, còn được gọi là phân loại dựa trên pixel, là một nhiệm vụ quan trọng trong đó ta phân loại từng pixel của hình ảnh thuộc về một lớp cụ thể.

