

Script_Poster_Report

April 10, 2024

1 Đặt vấn đề

1.1 Dự đoán sớm tình trạng ung thư tuyến giáp giúp đạt hiệu quả điều trị cao hơn

Ung thư tuyến giáp là loại ung thư phổ biến (đặc biệt với phụ nữ và khi đã qua tuổi 45)

Nghiên cứu từ Viện Ung thư Quốc gia Mỹ (National Cancer Institute) cho thấy rằng việc phát hiện ung thư tuyến giáp ở giai đoạn sớm giúp tăng cơ hội sống sót của bệnh nhân lên đến 98% so với chỉ 66% khi phát hiện ở giai đoạn muộn.

Cũng theo Nghiên cứu từ Viện Ung thư Quốc gia Mỹ (National Cancer Institute):

Việc chẩn đoán chính xác về loại và giai đoạn của ung thư giúp tăng cơ hội cho việc lựa chọn phương pháp điều trị phù hợp. Kết quả là, tỷ lệ sống sót và chất lượng cuộc sống sau điều trị đã được cải thiện đáng kể.

1.2 Phương pháp chọc hút tế bào (FNA) có nhiều ưu điểm trong dự đoán tình trạng bệnh với chuẩn Bethesda

Các phương pháp xác định ung thư tuyến giáp hiện nay có thể kể đến như sờ nắn để xác định khối u. Ngoài ra để xác định rõ hơn cần thực hiện xét nghiệm máu đánh giá các hormone hay nội soi thanh quản để lấy ảnh siêu âm.

Tuy nhiên các cách này vẫn cần được kiểm chứng lại do độ chính xác đặc biệt với các tình trạng bệnh phức tạp.

Bởi vậy có thêm các phương pháp như chọc hút tế bào FNA, cắt mô thùy, hay kiểm tra phân tử. Phương pháp cắt mô thùy sẽ cắt 1 phần mô lớn gây áp lực, đau đớn cho bệnh nhân và có mức độ xâm lấn cao. Trong khi phương pháp kiểm tra phân tử yêu cầu trang thiết bị tân tiến và khó triển khai trong đại trà.

Như vậy phương pháp chọc hút tế bào FNA - dùng kim tiêm nhỏ hút ra 1 lượng nhỏ tế bào - trở nên hiệu quả nổi bật khi ít xâm lấn mà vẫn có độ chính xác trong chẩn đoán tốt.

1.3 Hạn chế của các nghiên cứu hiện nay

Tập trung vào bài toán phân loại nhị phân với 2 nhãn Lành tính và Ác tính. Hoặc Lành tính với FTC - loại ung thư tuyến giáp phổ biến nhất.

Điều này gây khó khăn trong ứng dụng thực tế vì thang chuẩn Bethesda được phân làm nhiều bậc để dự đoán các tình trạng bệnh khác nhau.

2 Mục tiêu nghiên cứu

Áp dụng một số mô hình tiền huấn luyện trong xử lý ảnh để phân loại ảnh vi thể tế bào ung thư tuyến giáp

Tiêu chuẩn phân loại dựa theo chuẩn Bethesda với 3 nhãn (B2, B5, B6) của bộ dữ liệu được cung cấp bởi bệnh viện 108 - thiếu các nhãn còn lại do bác sĩ chưa làm sạch được dữ liệu.

3 Phát biểu bài toán

Để thực hiện được mục tiêu đã đặt ra \rightarrow đặt ra 2 bài toán cần xử lý

Bài toán 1: Tiền xử lý ảnh Đầu vào là ảnh chụp vi thể tế bào

Đầu ra là các ảnh cắt chứa vùng quan tâm sao cho kích thước phù hợp với đầu vào của bài toán 2

Bài toán 2: Huấn luyện mô hình Đầu vào là ảnh cắt thu từ bài toán 1

Đầu ra là vector chứa xác suất ảnh thuộc về các nhãn B2, B5, B6

4 Phương pháp nghiên cứu

Đây là hình thể hiện **luồng hoạt động của hệ thống**.

Bắt đầu từ việc thu thập dữ liệu. 1 kim tiêm sẽ chọc vào cổ và lấy ra 1 lượng nhỏ tế bào tuyến giáp. Sau đó các lớp tế bào được xếp vào dưới kính hiển vi để chụp ảnh.

Ảnh này sau đó được thực hiện tiền xử lý tùy theo yêu cầu của từng hệ thống (mục tiêu là giảm nhiễu, chọn ra các vùng đặc trưng nhất của ảnh để huấn luyện). Các phương pháp có thể dùng như cắt ra từng patch nhỏ hoặc chọn ở mức slide level tức 1 phần lớn của ảnh hoặc gần như toàn bộ ảnh gốc. Có thể thấy trong bước này, dự án em đã lựa chọn cắt thành **patch level** thay vì các slide level. Nguyên nhân các patch level sẽ giảm khối lượng tính toán xuống trong quá trình training cũng như giảm gánh nặng phải loại bỏ các vùng nhiễu có trên ảnh.

Tiếp đến là vào 1 mô hình phân loại dựa vào kỹ thuật deep learning. Lý do các phương pháp truyền thống như học máy với **cây phân loại và hồi quy CART**, hay **SVM**, hay sử dụng phân loại xác suất **Naive Bayes** yêu cầu hiểu biết các luật sâu rộng về dữ liệu để chọn ra các trường phân loại phù hợp - điều mà có thể cần đến 1 chuyên gia trong lĩnh vực y tế vào cuộc để làm rõ hơn về các trường thông tin này từ ảnh. Trong khi đó các mô hình học sâu như **CNN** đã được chứng minh đạt hiệu quả tốt cho tác vụ phân loại ảnh và học được nhiều đặc trưng có được từ ảnh.

Đến với bài toán 1.

Để cắt ra được ảnh quan tâm hiện tại có 2 hướng là phân đoạn các tế bào của ảnh gốc. Thực hiện cắt theo phân cụm hoặc theo mật độ rồi tiến hành khử nhiễu trên ảnh được cắt.

Cách thứ 2 chính là tiến hành khử nhiễu trước trên toàn bộ ảnh gốc như làm xám ảnh, phân cụm màu sắc để gom các tế bào cùng quan tâm về cùng 1 nhãn màu rồi tiến hành loại bỏ các nhãn mẫu khác (nhiều). Sau đó làm mờ toàn bộ ảnh gốc, giữ nguyên giá trị màu của các vùng tế bào quan tâm rồi mới thực hiện cắt ảnh ra sau.

Trong đó hướng thứ nhất đã và đang được em sử dụng, còn hướng thứ 2 là cách em mới thêm vào để xác định liệu thay đổi output của bài toán 1 thì hiệu suất output bài toán 2 ảnh hưởng thế nào.

Về **Phân đoạn**:

Các thuật toán phổ biến trong phân đoạn ảnh:

- U-Net: U-Net là một mô hình deep learning được thiết kế đặc biệt cho các tác vụ phân đoạn ảnh y tế, như phân loại tế bào trong hình ảnh y học.

- Mask R-CNN: Mask R-CNN là một mô hình deep learning phổ biến được sử dụng cho các tác vụ phân đoạn ảnh, đặc biệt là trong việc phát hiện và phân đoạn đối tượng trong hình ảnh.
- FCN (Fully Convolutional Networks): FCN là một kiến trúc mạng nơ-ron sử dụng toàn bộ lớp convolutional để thực hiện phân đoạn ảnh.
- DeepLab: DeepLab là một kiến trúc mạng sử dụng các lớp atrous convolution để mở rộng receptive field và cải thiện hiệu suất phân đoạn.

Thử nghiệm dùng celldetection với U-Net, ... tuy nhiên thời gian thực hiện phân đoạn lâu và khi độ chồng lấn các tế bào quá cao thì mô hình bỏ sót nơ.

Vậy nên chuyển sang dùng YOLO segment -> thu được kết quả nhanh hơn.

Đến với bài toán 2.

Hướng giải quyết được sử dụng là sử dụng học chuyển tiếp - transfer learning.

Cụ thể 1 mô hình đã được train sẵn trên 1 tập dữ liệu đủ rộng lớn và đa dạng về các object trong thế giới tự nhiên. Mô hình này sau đó được giữ nguyên trọng số ở các tầng đầu. Loại bỏ đi các tầng Fully Connected cuối cùng và tùy biến bằng các lớp phù hợp. Việc sử dụng học chuyển tiếp từ 1 mạng đã train trước đó giúp thời gian huấn luyện giảm xuống. Lý do vì mô hình trước đó đã học được nhiều đặc trưng của ảnh với bộ dữ liệu lớn dùng để đào tạo.

Các tầng sửa đổi có thể thêm vào là: Global Average Pooling (hoặc Flatten), Dense (còn gọi là Fully Connected), Dropout (để tránh over fit), ... và cuối cùng là lớp Dense đầu ra có số neuron bằng số nhãn để thực hiện việc dự đoán.

Các lớp trong một mạng nơ-ron có vai trò cụ thể trong quá trình huấn luyện và dự đoán. Dưới đây là vai trò cụ thể và ưu nhược điểm của mỗi lớp trong kiến trúc mạng nơ-ron được mô tả:

Global Average Pooling hoặc Flatten:

- Vai trò: Lớp này được sử dụng để chuyển đổi đầu ra của các lớp convolutional thành một vector 1 chiều, làm phẳng dữ liệu để có thể đưa vào lớp Dense (Fully Connected) tiếp theo.
- Ưu điểm: Giảm số chiều của dữ liệu, giúp tiết kiệm tài nguyên tính toán và giảm độ phức tạp của mạng.
- Nhược điểm: Mất mát thông tin không gian trong dữ liệu hình ảnh, không giữ được thông tin về mối quan hệ giữa các pixel.

Dense (Fully Connected):

- Vai trò: Lớp Dense được sử dụng để tạo ra các kết nối đầy đủ giữa các đơn vị nơ-ron từ lớp trước đó và lớp sau đó. Nó cung cấp khả năng học các mối quan hệ phi tuyến tính trong dữ liệu.
- Ưu điểm: Cung cấp khả năng học mối quan hệ phức tạp giữa các đặc trưng trong dữ liệu.
- Nhược điểm: Tăng số lượng tham số và độ phức tạp của mạng, dễ gây overfitting nếu không sử dụng các kỹ thuật như dropout.

Dropout:

- Vai trò: Dropout là một kỹ thuật regularization được áp dụng trong quá trình huấn luyện mạng nơ-ron để tránh overfitting. Nó ngẫu nhiên loại bỏ một phần của các đơn vị nơ-ron trong quá trình huấn luyện.
- Ưu điểm: Giảm nguy cơ overfitting bằng cách loại bỏ sự phụ thuộc quá mức vào một số đơn vị nơ-ron cụ thể.
- Nhược điểm: Có thể làm giảm hiệu suất của mạng trong quá trình dự đoán nếu áp dụng quá mức hoặc không đúng cách.

Lớp Dense đầu ra:

- Vai trò: Là lớp cuối cùng của mạng nơ-ron, được sử dụng để thực hiện việc dự đoán hoặc phân

loại. Số lượng neutron trong lớp này thường bằng số lớp nhân của tác vụ phân loại.

- Ưu điểm: Cho phép mô hình sản xuất kết quả cuối cùng dưới dạng xác suất hoặc giá trị dự đoán cho mỗi lớp nhân.

- Nhược điểm: Không có nhược điểm nổi bật, nhưng hiệu suất của mô hình có thể phụ thuộc vào cách thiết kế của lớp này và các lớp trước đó.

Khác biệt của lớp Pooling và Flatten:

Lớp Pooling:

- Mục tiêu: Lớp Pooling thường được sử dụng để giảm kích thước không gian của đầu ra từ các lớp convolutional và giúp làm giảm độ phức tạp của mô hình. Nó cũng giúp giảm việc overfitting bằng cách giảm số lượng tham số cần học.

- Phương pháp hoạt động: Lớp Pooling thực hiện việc tổng hợp thông tin từ các vùng nhỏ của ảnh bằng cách áp dụng một toán tử như max-pooling hoặc average-pooling. Max-pooling lấy giá trị lớn nhất từ mỗi cửa sổ, trong khi average-pooling tính trung bình của các giá trị.

Lớp Flatten:

- Mục tiêu: Lớp Flatten chuyển đổi một đầu ra ma trận (2D hoặc 3D) từ lớp trước thành một vector 1D, giúp chuẩn bị dữ liệu cho các lớp Dense (Fully Connected) tiếp theo.

- Phương pháp hoạt động: Lớp Flatten không thực hiện bất kỳ phép biến đổi nào trên dữ liệu, nó chỉ đơn giản là duỗi hoặc làm phẳng ma trận đầu ra từ các lớp trước thành một vector 1 chiều.

Tóm lại, lớp Pooling được sử dụng để giảm kích thước không gian của đầu ra, trong khi lớp Flatten được sử dụng để chuyển đổi đầu ra từ một ma trận thành một vector.

5 Kết quả và thảo luận

Dữ liệu đầu vào

Dữ liệu sau cắt ảnh

Tiến hành cân bằng dữ liệu:

Tăng cường dữ liệu (Data Augmentation):

Tăng cường dữ liệu là quá trình tạo ra các biến thể của các mẫu dữ liệu hiện có bằng cách áp dụng các biến đổi như xoay, phản chiếu, phóng to, thu nhỏ, cắt, hoặc làm mờ ảnh. Điều này giúp tăng cường độ đa dạng của dữ liệu và giảm thiểu overfitting. Cụ thể em sử dụng: xoay, phản chiếu, phóng to, thu nhỏ.

Đánh giá bài toán 1

Đánh giá bài toán 2

ResNet50:

- Kiến trúc: ResNet50 là một trong các biến thể của mạng ResNet (Residual Network), nổi tiếng với việc sử dụng các khối còn gọi là “residual blocks” để tránh việc mất thông tin trong quá trình huấn luyện mạng sâu.

- Vấn đề cụ thể: ResNet được xây dựng để giải quyết vấn đề đánh giá kém hiệu suất của các mạng nơ-ron sâu khi chúng trở nên quá sâu, dẫn đến hiện tượng vanishing gradient và degradation nên không cập nhật hệ số của các tầng đầu tiên được.

- Nhược điểm: Một trong những nhược điểm của ResNet là cấu trúc khá phức tạp, đòi hỏi nhiều tài nguyên tính toán và thời gian huấn luyện.

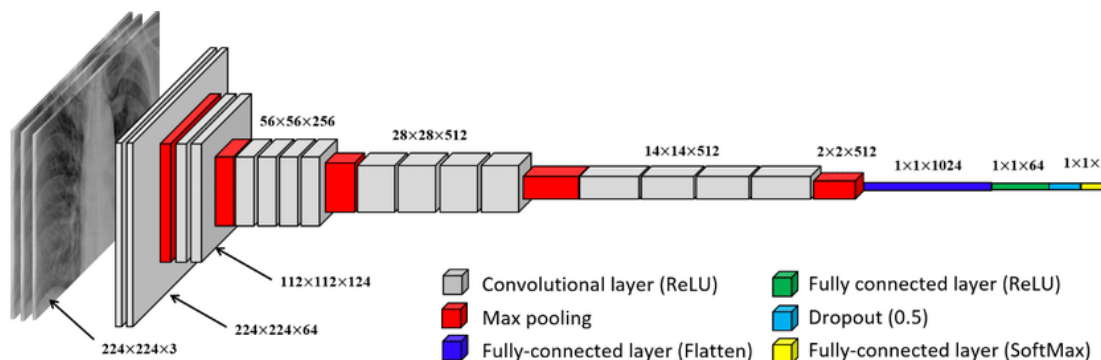
EfficientNetB0:

- Kiến trúc: EfficientNetB0 là một trong các phiên bản của dòng mô hình EfficientNet, được thiết kế để tối ưu hóa hiệu suất và chi phí tính toán của mạng nơ-ron sâu thông qua việc tinh chỉnh cùng lúc ba yếu tố quan trọng: chiều sâu, chiều rộng và độ phân giải.
- Vấn đề cụ thể: EfficientNet được xây dựng để giải quyết vấn đề về hiệu suất và tính hiệu quả của mạng nơ-ron sâu, bằng cách tối ưu hóa các yếu tố kiến trúc của mô hình.
- Nhược điểm: Mặc dù có hiệu suất và hiệu quả tính toán tốt hơn so với nhiều mô hình khác, nhưng phiên bản B0 của EfficientNet có thể bị hạn chế về khả năng học các đặc trưng phức tạp hơn do có kích thước nhỏ.

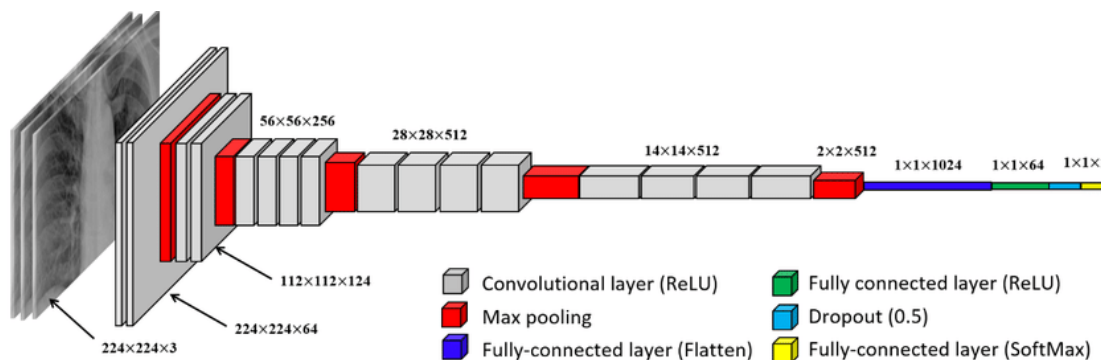
VGG19:

- Kiến trúc: VGG19 là một trong những mô hình CNN cổ điển, với kiến trúc đơn giản bao gồm một loạt các lớp convolutional và lớp pooling, kết thúc bằng các lớp fully connected.
- Vấn đề cụ thể: VGG19 được xây dựng để tạo ra một mô hình CNN cơ bản và dễ hiểu, giúp nghiên cứu cũng như triển khai thực tế một cách dễ dàng.
- Nhược điểm: Mặc dù đơn giản và dễ triển khai, VGG19 có khả năng học các đặc trưng cơ bản, nhưng không hiệu quả bằng các mô hình tiên tiến hơn trong việc học các đặc trưng phức tạp và tự động học các phụ thuộc bậc cao hơn giữa các đặc trưng.

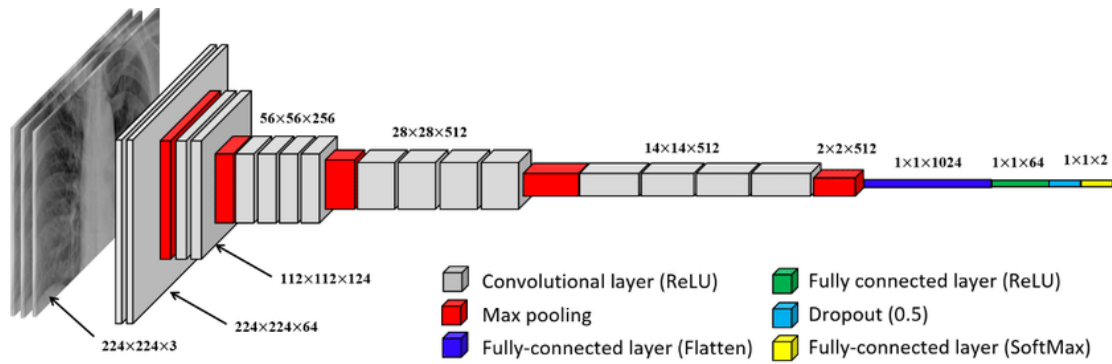
Với ResNet50



Với EfficientNetB0



VGG19



Kết quả ma trận nhầm lẫn ...

Kết quả Accuracy và F1 Score

```
[ ]: import numpy as np

# Confusion matrix
conf_matrix = np.array([[1787, 980, 614], [960, 2279, 142], [634, 122, 2625]])

# Chỉ số cho từng lớp
for i in range(conf_matrix.shape[0]):
    TP = conf_matrix[i, i]
    FP = np.sum(conf_matrix[:, i]) - TP
    FN = np.sum(conf_matrix[i, :]) - TP
    TN = np.sum(conf_matrix) - (TP + FP + FN)

    accuracy = (TP + TN) / (TP + TN + FP + FN)
    precision = TP / (TP + FP)
    recall = TP / (TP + FN)
    f1_score = 2 * (precision * recall) / (precision + recall)

    print("Class:", i+1)
    print("Accuracy:", accuracy)
    print("Precision:", precision)
    print("Recall:", recall)
    print("F1 Score:", f1_score)
    print()
```

```
Class: 1
Accuracy: 0.6856945676821453
Precision: 0.528541851523218
Recall: 0.528541851523218
F1 Score: 0.528541851523218
```

```
Class: 2
Accuracy: 0.7827072858128759
Precision: 0.6740609287193138
```

Recall: 0.6740609287193138
F1 Score: 0.6740609287193138

Class: 3
Accuracy: 0.8509316770186336
Precision: 0.7763975155279503
Recall: 0.7763975155279503
F1 Score: 0.7763975155279503

```
[ ]: import numpy as np

# Confusion matrix
conf_matrix = np.array([[1787, 980, 614], [960, 2279, 142], [634, 122, 2625]])

# True Positives (TP), True Negatives (TN), False Positives (FP), False
↪Negatives (FN)
TP = np.diag(conf_matrix)
FP = np.sum(conf_matrix, axis=0) - TP
FN = np.sum(conf_matrix, axis=1) - TP
TN = np.sum(conf_matrix) - (TP + FP + FN)

# Accuracy
accuracy = np.sum(TP) / np.sum(conf_matrix)
print("Accuracy:", accuracy)
```

Accuracy: 0.6596667652568273

6 Kết luận và kiến nghị

Với các nghiên cứu trước đây [2] kết hợp kết quả nghiên cứu hiện tại cho thấy tiềm năng cải tiến deep learning phân loại ảnh FNA đa nhân theo chuẩn Bethesda

Hướng nghiên cứu tương lai cần mở rộng bài toán cho các dữ liệu từ nhiều trung tâm cũng như phát triển tính giải thích được nguyên nhân phân loại để gây dựng niềm tin với các bác sĩ