ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

❧ • ❧

BÁO CÁO TIẾN ĐỘ

CÁC KỸ THUẬT HỌC SÂU VÀ ỨNG DỤNG

ĐỀ TÀI

Human Protein Atlas Image Classification

Giảng viên hướng dẫn: Th.S Nguyễn Vinh Tiệp

Nhóm sinh viên thực hiện: Nguyễn Minh Dũng - 15520138

Trịnh Hoàng Ngọc - 15520556

Tp. Hồ Chí Minh, ngày 28 tháng 11 năm 2018

1. Mô tả bài toán

1.1. Giới thiệu bài toán

Dự đoán nhãn của mẫu protein có trong tế bào người. Có tổng cộng **28 nhãn khác nhau** có trong tập dữ liệu. Tập dữ liệu được thu thập theo cách được tiêu chuẩn hóa cao bằng cách sử dụng một phương thức hình ảnh (kính hiển vi tiêu điểm). Tuy nhiên, tập dữ liệu bao gồm 27 loại tế bào khác nhau về hình thái rất khác nhau, ảnh hưởng đến phân loại các mẫu protein của các bào quan khác nhau.

Tất cả các mẫu hình ảnh được đại diện bởi **bốn bộ lọc** (được lưu trữ dưới dạng tệp riêng lẻ): protein được quan tâm (bộ lọc màu xanh lá) cộng với ba mốc bộ lọc khác, bao gồm: hạt nhân (xanh dương), vi ống (đỏ), mạng lưới nội chất (vàng). Do đó, bộ lọc màu xanh lá cây nên được sử dụng để dự đoán nhãn và các bộ lọc khác được sử dụng làm tham chiếu.

Một ảnh đầu vào có thể có nhiều nhãn đầu ra (bài toán Multilabel Classification). Kích thước ảnh là 512x512(pixel).

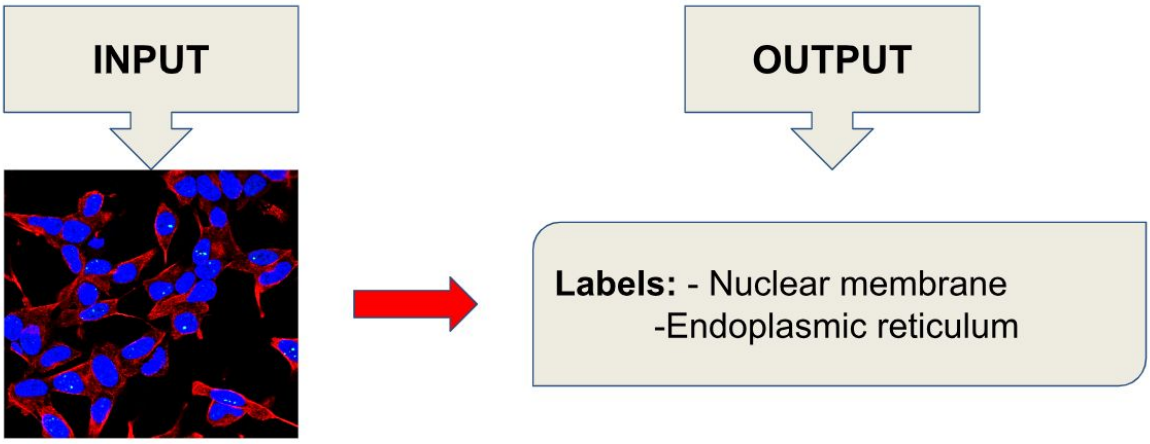
1.2. Mục tiêu bài toán

Xác định nhãn của từng loại protein có trong tế bào người từ hình ảnh tế bào.

1.3. Xác định bài toán

Input: Hình ảnh tế bào

Output: nhãn của loại protein cần quan tâm.



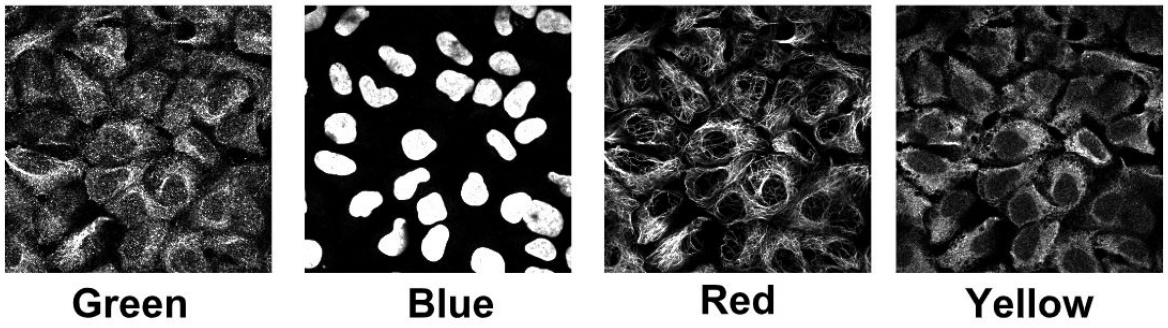
2. Dữ liệu

2.1. Phân tích dữ liệu

Bộ dữ liệu bài toán bao gồm 42774 samples (trong đó 31072 train samples, 11702 test samples). Trong đó, mỗi một sample được tách thành 4 file ảnh khác nhau tương ứng với 4 filter:

- **Green**: thể hiện cấu trúc protein cần quan tâm
- **Blue**: landmark filter for the **nucleus** - biểu diễn phần hạt nhân của tế bào
- **Red**: landmark filter for **microtubules** - vi ống
- **Yellow**: landmark filter for the **endoplasmatic reticulum** - mạng lưới nội chất tế bào

  Định dạng mỗi filter: [filename]_[filter color].png



Bảng: Danh sách các nhãn và số lượng của nó

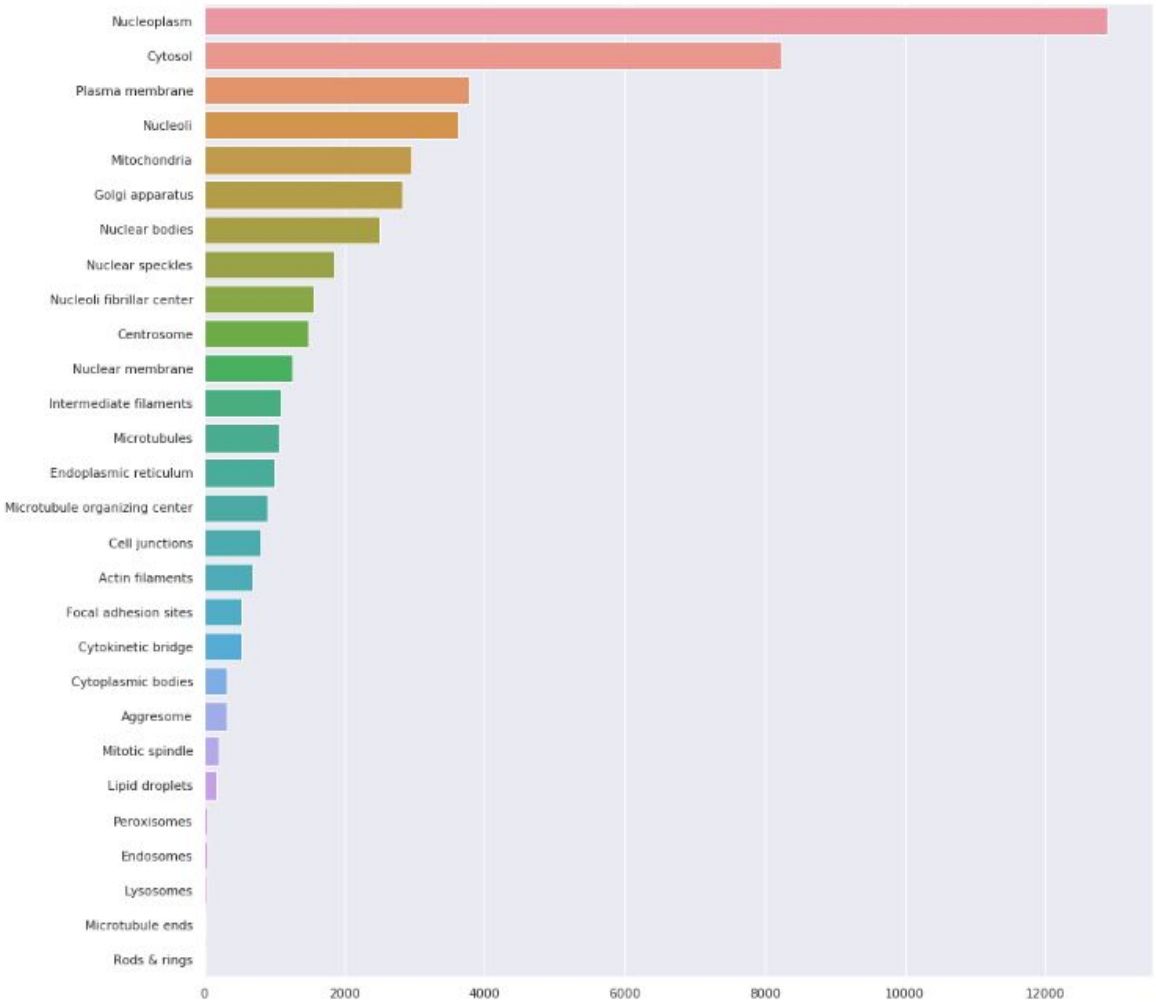
Class	Total	Class	Total	Class	Total	Class	Total
0. Nucleoplasm	12885	7. Golgi apparatus	2822	14. Microtubules	1066	21. Plasma membrane	3777
1. Nuclear membrane	1254	8. Peroxisomes	53	15. Microtubule ends	21	22. Cell junctions	802
2. Nucleoli	3621	9. Endosomes	45	16. Cytokinetic bridge	530	23. Mitochondria	2965
3. Nucleoli fibrillar center	1561	10. Lysosomes	28	17. Mitotic spindle	210	24. Aggresome	322

4. Nuclear speckles	1858	11. Intermediate filaments	1093	18. Microtubule organizing center	902	25. Cytosol	8228
5. Nuclear bodies	2513	12. Actin filaments	688	19. Centrosome	1482	26. Cytoplasmic bodies	328
6. Endoplasmic reticulum	1008	13. Focal adhesion sites	537	20. Lipid droplets	172	**27. Rods & rings**	**11**

Theo như bảng thống kê, ta thấy protein loại **Nucleoplasm** chiếm đa số (12885 samples) và **Rod & rings** chiếm số lượng ít nhất (11 samples).

Dữ liệu bài toán có sự chênh lệch lớn giữa nhãn chiếm đa số và nhãn chiếm thiểu số, vì vậy, cần phương pháp thích hợp để đạt được độ chính xác nhãn đầu ra.

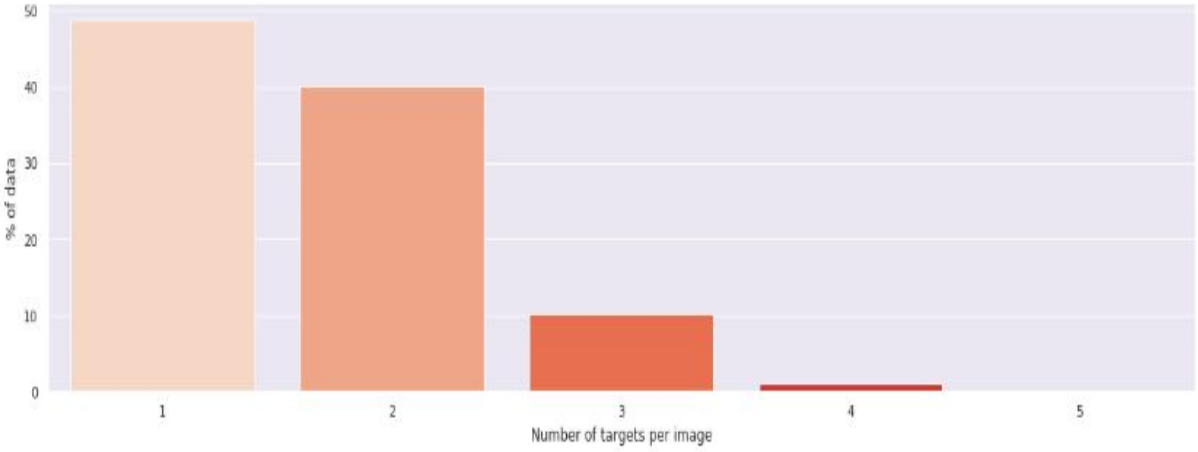
2.2. Mô hình hoá dữ liệu



Dựa theo biểu đồ, ta thấy **Nucleoplasm** và **Cytosol** chiếm nhiều nhất (gần 20.000 sample - 66%), còn **Peroxisomes, Endosomes, Lysosomes, Microtubate ends, Rod & rings** chiếm ít nhất (cả 5 loại chỉ chiếm ~0.33% tổng số mẫu)

2.3. Nhãn

Như nhận định ban đầu, mỗi một ảnh đầu vào có thể có nhiều hơn 1 nhãn đầu ra, mô hình sau phản ảnh số lượng nhãn được gán trên mỗi bức ảnh:



Nhận xét:

- Gần 90% các ảnh có từ 1 đến 2 nhãn phân loại
- Hơn 10% còn lại thuộc về lớp có 3,4 hoặc 5 nhãn.

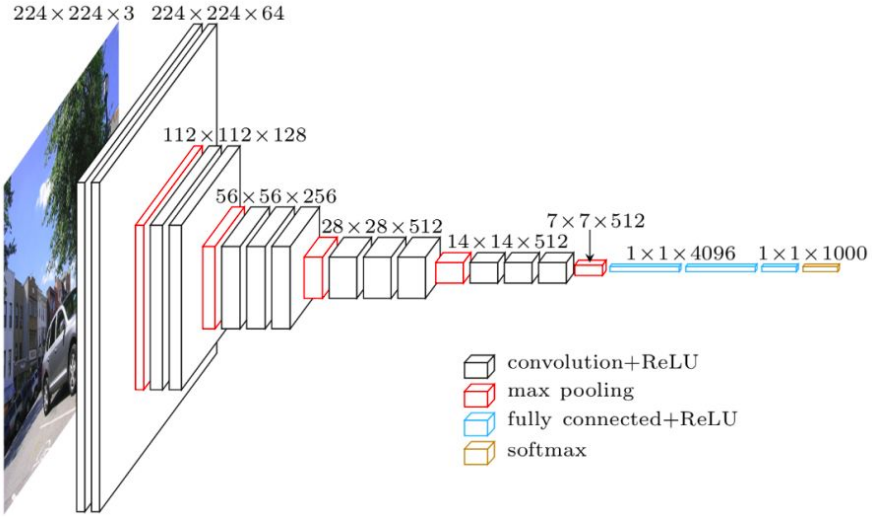
3. Phương pháp thực hiện

3.1. Xử lý dữ liệu

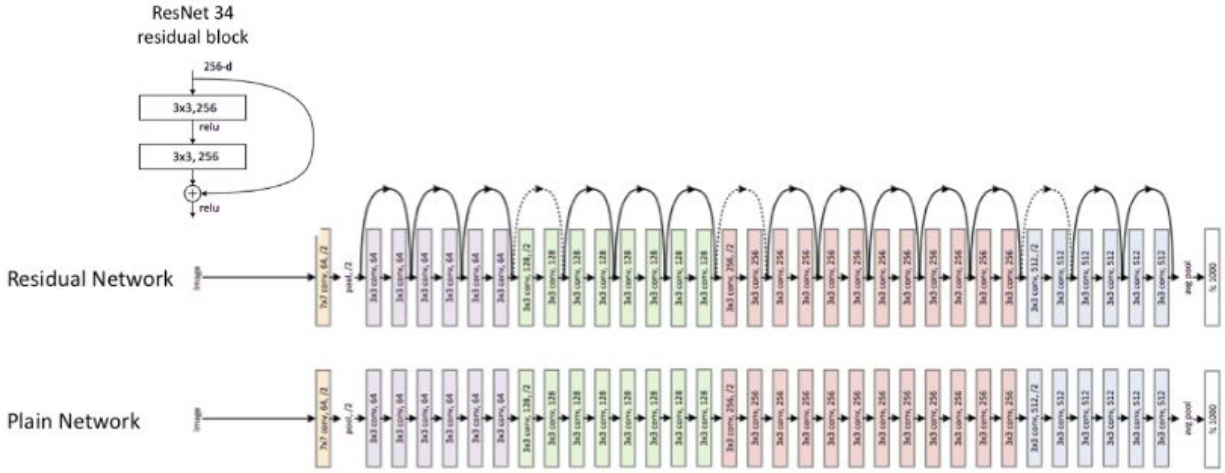
Tách 31072 samples thành 24857 samples làm tập train và 6215 sample làm tập validation - (80% train, 20% val).

3.2. Mô hình các model đã thử nghiệm

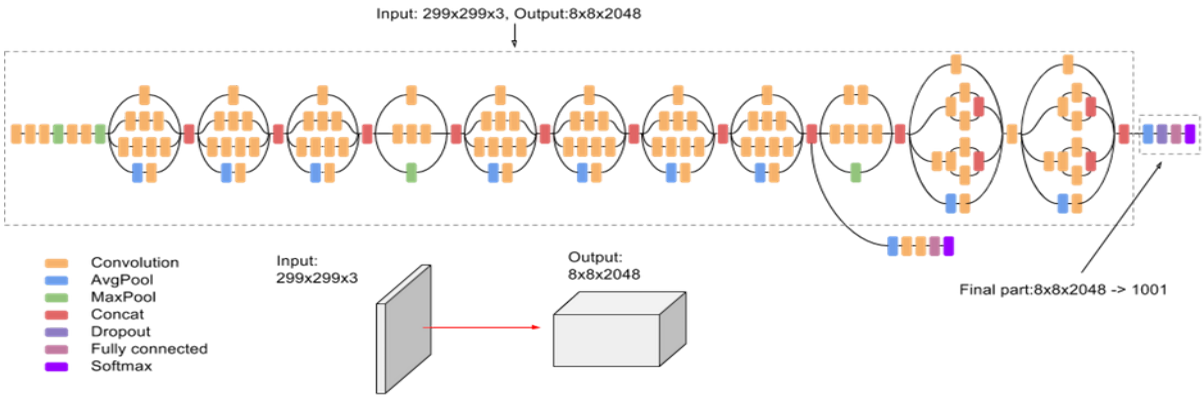
- **VGG16**



- **Resnet**



- **InceptionResnet**



3.3. Thông số retrain

- **Input:** Image with size 224x224x3 (VGG, Inception), 256x256x3 (ResNet34)
- **Pre-train:** VGG16, ResNet34, InceptionResNetV2
- **Data augmentation:** Flip, Rotate, Lighting.
- **Optimize:** AdamOptimizer
- **Loss function:** Facal Loss. Vì dữ liệu bài toán không cân bằng.
- **Ngưỡng để phân lớp (lấy trên kernel kaggle):**
  th_t = np.array([0.565,0.39,0.55,0.345,0.33,0.39,0.33,0.45,0.38,0.39,
  0.34,0.42,0.31,0.38,0.49,0.50,0.38,0.43,0.46,0.40,
  0.39,0.505,0.37,0.47,0.41,0.545,0.32,0.1])
- **Output:** Sử dụng Sigmoid function. Vì đây là bài toán 1 lớp có thể có nhiều nhãn nên ta không dùng Softmax.
- **Độ đo đánh giá:** Marco-F1 và Kaggle LB

3.4. Các thử nghiệm

- **Thử nghiệm 1:** Gộp 3 kênh màu R,G,B thành 1 ảnh sử dụng làm input cho retrain.
  Kết quả:

R+G+B	InceptionResnetV2	Resnet34	VGG16
Validation(marco-F1 with th = 0.5)	**0.396**	**0.495**	**0.51**
Validation(marco-F1 with th_t)	0.377	0.476	0.49
Test (kaggle LB with thres = 0.5)	0.318	0.359	0.375
Test (kaggle LB with thres = th_t)	**0.334**	**0.414**	**0.431**

Nhận xét: Model VGG16 cho kết quả tốt hơn so với 2 model còn lại. LB cao nhất hiện tại đang là **0.588**.

- **Thử nghiệm 2:** Gộp 4 kênh màu R,G,B,Y theo tỷ lệ: (R/2+Y/2,G,B/2+Y/2)
  Kết quả:

	Resnet34	VGG16
Validation(F1-score with th_t)	0.474	0.5
Test (kaggle LB with thres = th_t)	**0.418**	**0.418**

Nhận xét: Kết quả Resnet cao hơn so với thử nghiệm 1. Tuy nhiên vẫn thấp hơn so với VGG16 ở thử nghiệm 1.

- **Thử nghiệm 3:** Kết hợp output của 2 model Resnet34 và VGG16 ở thử nghiệm 1. Tính output predict theo tỷ lệ 0.3*Resnet + 0.7VGG

Kết quả:

0.3Resnet+0.7VGG	Acc
Validation(F1-score with th_t)	0.5
Test (kaggle LB with thres = th_t)	**0.439**

Nhận xét: Độ chính xác được cải thiện nhẹ. Tuy nhiên cần thử nghiệm thêm nhiều phương pháp khác nữa để cải thiện độ chính xác.