

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Lớp: CS2225.CH1501

Môn: NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

GV: PGS.TS Lê Đình Duy
Trường ĐH Công Nghệ Thông Tin, ĐHQG-HCM



ỨNG DỤNG TẠO ẢNH THẺ

Nguyễn Hoàng Thịnh - CH2001016

Nguyễn Thanh Phong - CH2001012

Nguyễn Quan Duy Tùng - CH2001019

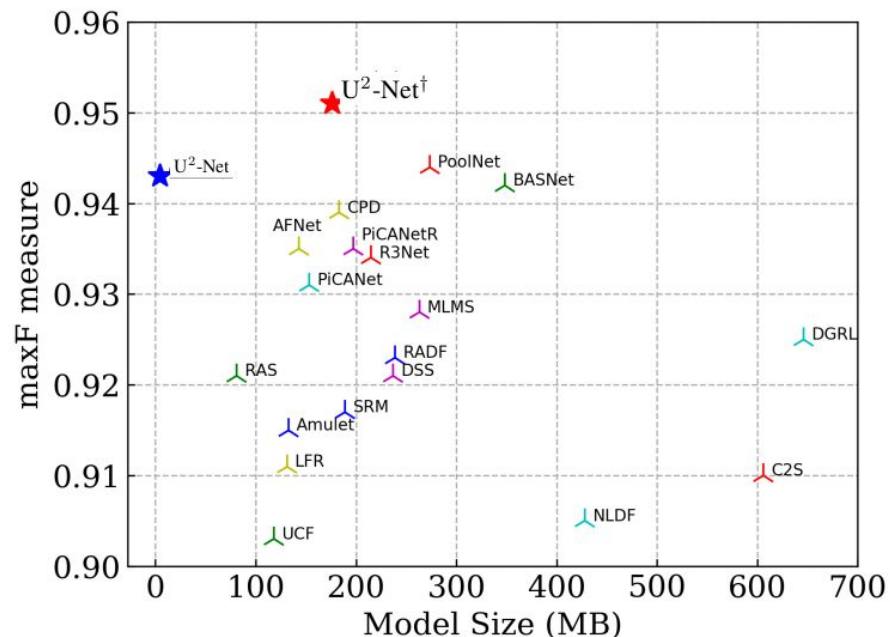
Link Github: <https://github.com/nhthinh/CS2225.CH1501>

Tóm tắt

- Tên đề tài: Phát hiện cây lúa bệnh dựa vào hình ảnh
- Tóm tắt về đề án và kết quả đạt được
 - Salient Object Detection (phát hiện đối tượng nổi bật) (SOD) với model U2-net+, nhằm mục đích bóc tách các đối tượng trong ảnh.
 - Sử dụng Deep Convolution Neural Networks (CNN), Fully Convolutional Networks (FCN), Residual U-blocks (RSU)

Tóm tắt

Hiệu quả hơn so với các mô hình sử dụng backbones, chẳng hạn như Alexnet, VGG, ResNet, DenseNet,... vì mục đích ban đầu của backbones được thiết kế để phân loại, trích xuất tính năng đại diện để định danh hơn là segmentation toàn bộ đối tượng.



Ảnh của các thành viên của nhóm



Nguyễn Hoàng Thịnh



Nguyễn Thanh Phong



Nguyễn Quan Duy Tùng

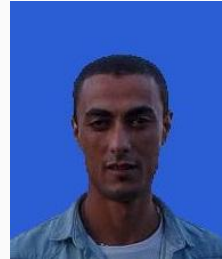
Mô tả bài toán

Task: Tạo ảnh thẻ chân dung

Input: hình ảnh có gương mặt người



Output: hình thẻ chân dung

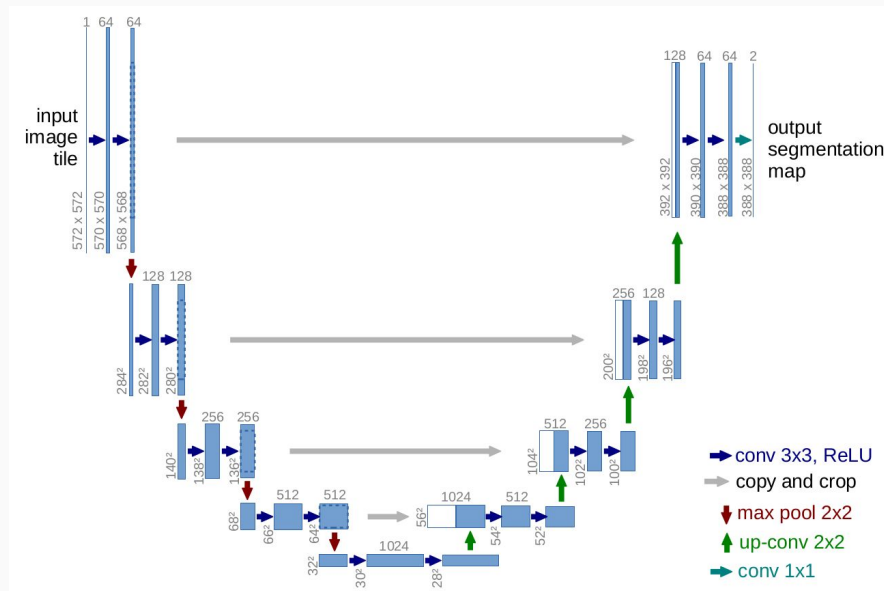


Loại bài toán ML

- Segmentation

- U-Net model

- Không sử dụng fully connected do đó có thể chấp nhận input với kích thước bất kì.
 - Padding method giúp phân đoạn hình ảnh hoàn toàn mà không bị hạn chế bởi dung lượng bộ nhớ GPU.



Loại bài toán ML

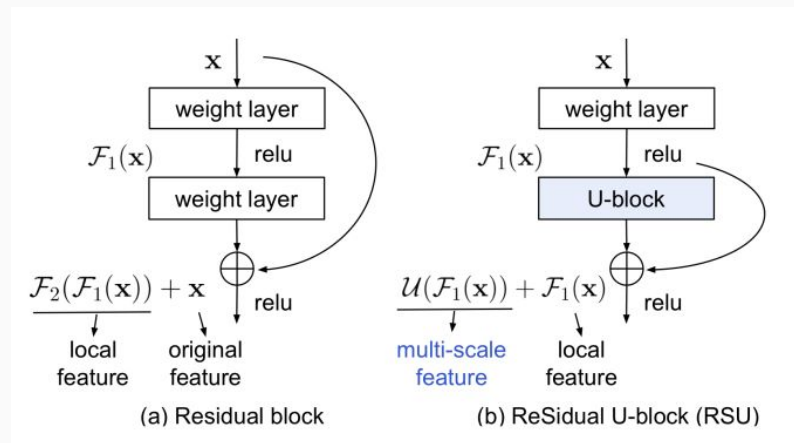
- Segmentation

- ReSidual U-Blocks

- Được thiết kế nhằm biến đổi các đối tượng $x(H \times W \times C_{in})$ thành các feature map $F_1(x)$ với C_{out} trích xuất local features

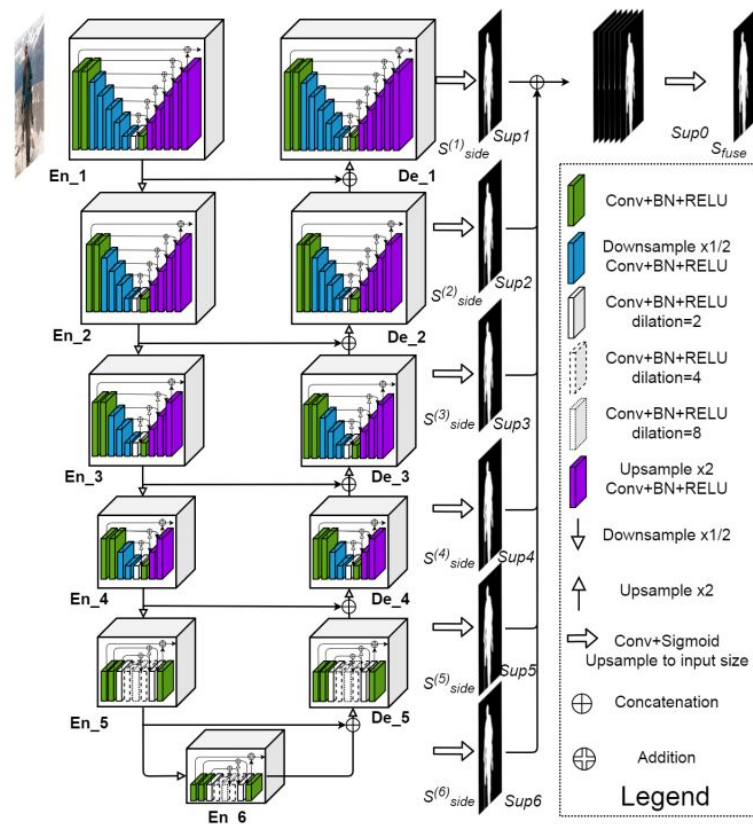
- Nhúng $F_1(x)$ vào U-Net tạo liên kết các local features và multi-scale features

$$F_1(x) + U(F_1(x))$$



Loại bài toán ML

- Segmentation
 - U2-Net model
 - Không sử dụng pre-trained backbones
 - Với cấu trúc Encoder-Decoder với mỗi khối là một RSU-block được nhúng U-Net modules nhằm khai thác multi-scale và multi-level features .



Các bước thực hiện

Bước 1. Load hình ảnh

Sử dụng file.uploads từ google.colab



Bước 2. Xử lý hình ảnh với model u2netp

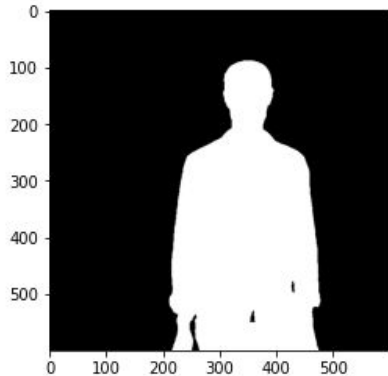
Sử dụng model u2netp để bóc tách đối tượng



Các bước thực hiện

Bước 3. Xét quá ngưỡng để remove các phần bị mờ (ThreHold = 0.9)

```
# define the cutoff threshold below which,  
# background will be removed.  
THRESHOLD = 0.9  
  
# refine the output  
out_img[out_img > THRESHOLD] = 1  
out_img[out_img <= THRESHOLD] = 0  
plt.imshow(out_img)  
shape = out_img.shape
```



Bước 4. Merge 2 layers để loại bỏ background



Các bước thực hiện

Bước 5. Đổi màu nền



Bước 6. Sử dụng model u2net_portrait và Haar Cascades để detect face. Sau đó crop vị trí face.



Dữ liệu

- Data-train: **DUTS-TR** là một phần của tập dữ liệu **DUTS**, có 10553 ảnh. Một tập dữ liệu thường dùng để SOD.

<http://saliencydetection.net/duts/download/DUTS-TR.zip>

- Data-test: **DUT-OMRON** có 5168 ảnh có chứa 1 hoặc 2 đối tượng; và **ECSSD** có 1000 ảnh phức tạp chứa các đối tượng lớn trong ảnh.

<http://saliencydetection.net/dut-omron/download/DUT-OMRON-image.zip>

<https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/data/ECSSD/images.zip>

Phương pháp đánh giá

$$F_{\beta}^w = (1 + \beta^2) \frac{\text{Precision}^w \cdot \text{Recall}^w}{\beta^2 \cdot \text{Precision}^w + \text{Recall}^w}$$

$$MAE = \frac{1}{H \times W} \sum_{c=1}^H \sum_{c=1}^W |P(r, c) - G(r, c)|$$

	DUT-OMRON		ECSSD		
Configuration	maxF _β	MAE	maxF _β	MAE	Time(ms)
Baseline U-Net	0.725	0.082	0.896	0.066	14
RES U-Net	0.781	0.065	0.933	0.042	19
VGG-16 backbone	0.808	0.063	0.942	0.038	23
ResNet-50	0.813	0.058	0.937	0.041	41
RSU U ² -Net	0.823	0.054	0.951	0.033	33
RSU U ² -Net ⁺	0.813	0.060	0.943	0.041	25

Hạn chế và hướng phát triển

- Do bộ dữ liệu train chứa đối tượng đa dạng, có thể thay thế một bộ dữ liệu mới phù hợp hơn với đối tượng người dùng.
- Khai thác kết quả để sử dụng trong các hệ thống khác (web, mobile app,...)