**Inpainting**

1. **Định nghĩa**

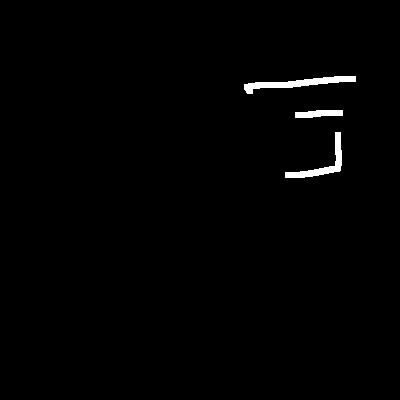
* Inpainting là process xử lý fill những chỗ trống ở trong ảnh để tạo ra một bức ảnh hoàn chỉnh.

1. **Application**

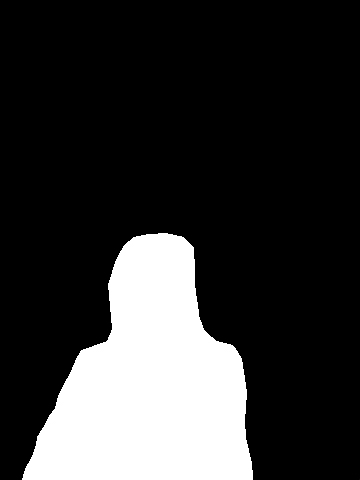
* Chỉnh xửa ảnh (loại bỏ object ở background)
* Xóa watermark
* De-censor ảnh 18+

1. **Experience**

* Small

****

**-**Big



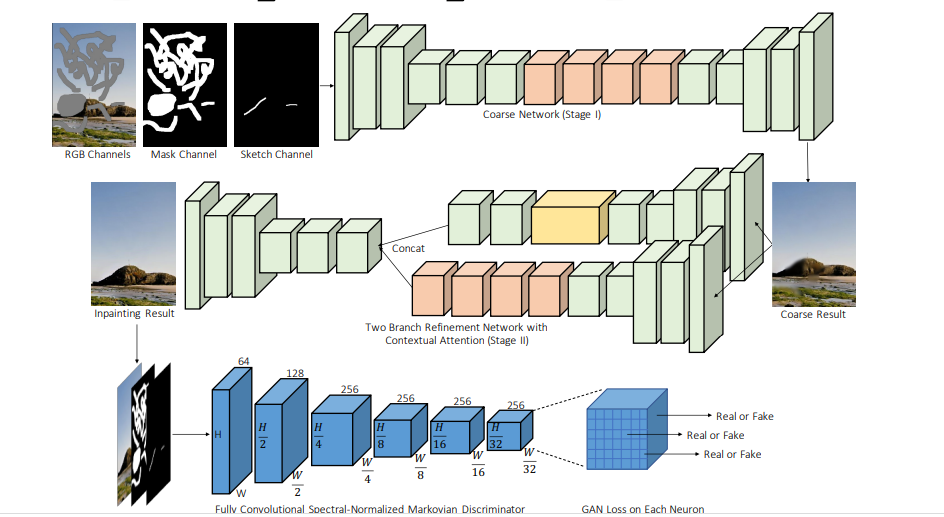
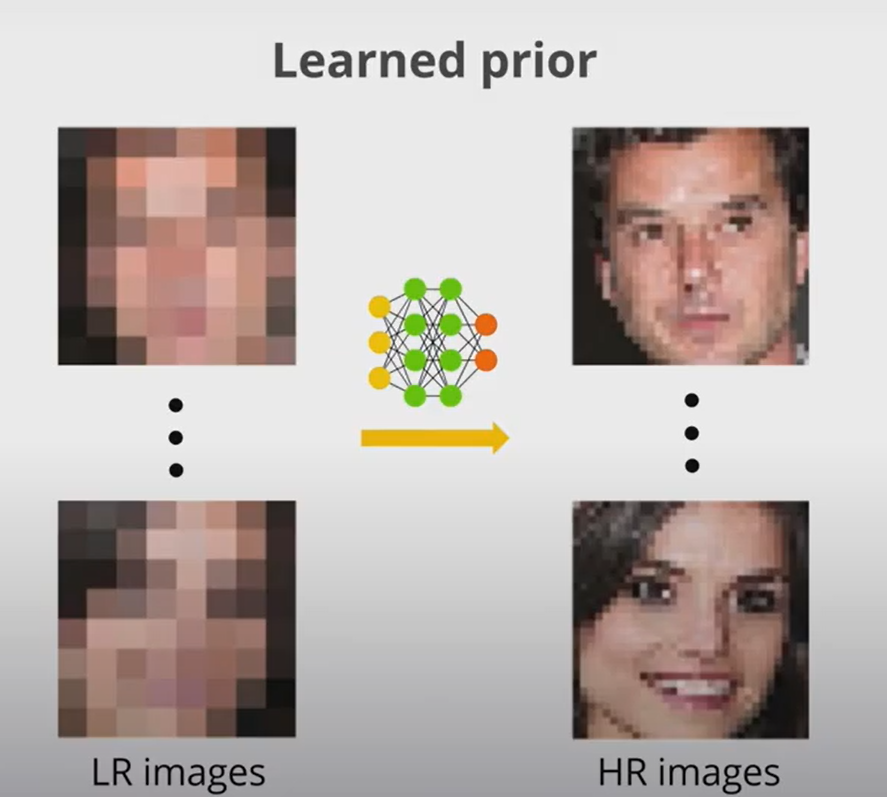
* Water mask:



1. **Phương pháp có sẵn trên Opencv**

* Fast Marching Method của paper \*\*"An Image Inpainting Technique Based on the Fast Marching Method"\*\* by Alexandru Telea in 2004. Ý tưởng của thuật toán là đi từ rìa của phần chưa được tô vào trong. Mỗi pixel được chỏ đến sẽ được tính bằng tổng normalized của những pixel neighbors. Xong sẽ tìm đến pixel tiếp theo để chạy tiếp. Flag là cv.INPAINT\_TELEA
  + Kq:
    - Small****
    - Big****
* Dựa trên paper \*\*"Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting"\*\* by Bertalmio, Marcelo, Andrea L. Bertozzi, and Guillermo Sapiro in 2001. Method chạy quay rìa của phần cần inpainted (chuyển line sang point cùng intensity) sau đó dùng gradient vectors để tô áp dụng fluid dynamics. Flag là  [**cv.INPAINT\_NS**](https://docs.opencv.org/4.x/d7/d8b/group__photo__inpaint.html#gga8002a65f5a3328fbf15df81b842d3c3ca05e763003a805e6c11c673a9f4ba7d07)
  + Kq
    - Small ****
    - Big****

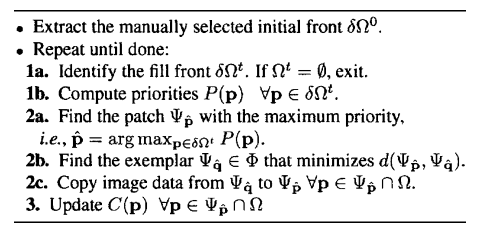
1. **Deep learning method**

* Tất nhiên là phải có deep learning, phương pháp rất là phổ biến và có thể dùng được dùng cho rất nhiều bài toán một cách không não.
* Đơn giản là tạo ra một cái mạng convolution hoặc transformer hay Gans j đó. Xong cứ tống cho training với một bộ dataset đủ lớn (input là ảnh cần inpainting với output là ảnh đã fill hết) là xong. Training xong thì chỉ cần nhét input ảnh cần qua model là nó tự trả về ảnh đã inpaint.
* Ở đây dùng mạng Deepfillv2
* ****
* Trên Sota thì đang có mạng Palette đang đứng top.
* Vấn đề của phương pháp deep learning luôn là data nhưng data càng ngon thì tỉ lệ thuận với kết quả.
* 
* KQ:
  + Small:****
  + Big:****

1. **Deep Image Prior**

* Deep Image Prior của Dmitry Ulyanov - 2018
* Tiếp đến là một phương pháp cũng dùng deep learning nhưng lại theo hướng khác.
* Ý tưởng của phương pháp này là sử dụng việc pior về ảnh của bản than model đã tốt sẵn r. Và thay vì sử dụng learn prior như bình thường thì cái này sử dụng explicit prior method. Bài toán nó trở thành bài toán tối ưu công thức này.
* Với:
  + x là ảnh clean
  + là ảnh corrupted
  + E là hàm quan hệ giữa ảnh clean và ảnh corrupted của inpaint là như sau:
    - với m là ma trận mask.
  + R là hàm chuẩn hóa
* Phương pháp này modify công thức trên như sau:
  + Đầu tiên là chuyển không gian search từ ảnh sang tìm kiếm parameter cho model để tạo ra được ảnh gần với ảnh gốc (corrupted) từ một ảnh noise.
  + Để hệ hàm hóa thành 0.
* Với z là ảnh noise và là model của mình.
* Bài toán trở về thành bài toán optimizer tìm parameter cho model bằng gradient accent.
* Bài này thì ko cần data chỉ cần ảnh corrupted với mask thôi, nhưng tất nhiên kq ko bằng deep learning thường được.
* Phương pháp này cần chú ý là phải dừng việc training tìm parameter sớm nếu ko sẽ bị overfit (giống với ảnh gốc quá nhiều).
* Tóm lại là có những bước sau:
  + Với input là x (corrupt image)
  + Khởi tạo input z uniform noise hay lấy bừa ảnh.
  + Optimize function với gradient
  + Tìm được parameter r thì cho ảnh đi qua model với parameter đó là xong.
* KQ:

1. **Exampler-Based**

* Region Filling and Object Removal by Exemplar-Based Image Inpainting bởi Antonio Criminisi - 2004
* Ý tưởng của method này là đầu tiên chọn ra một patch (block) có priority cao nhất trong chô chưa được tô, rồi tìm patch (block) nào phù hợp nhất đã được tô r copy vào.
* Base line của thuât toán
* ****
* Vấn đề là chậm khủng kiếp do phải tìm tất cả vùng source. Với cả ảnh ko mượt do là copy block.

1. **Future work**

* Áp dụng nhưng mạng mới transformer gans.
* Cái Exampler based thì tìm trong neighbor thôi.