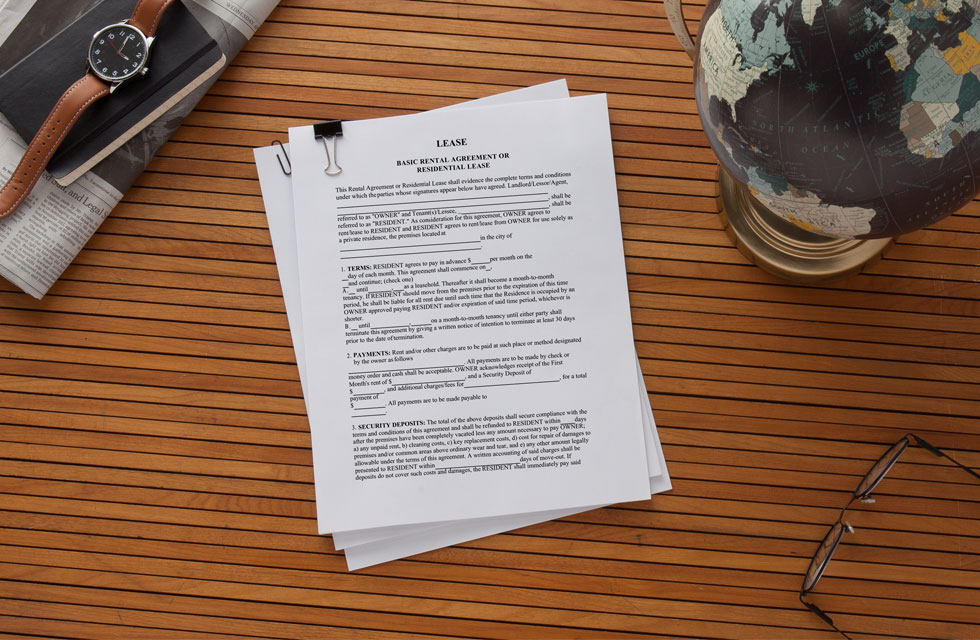
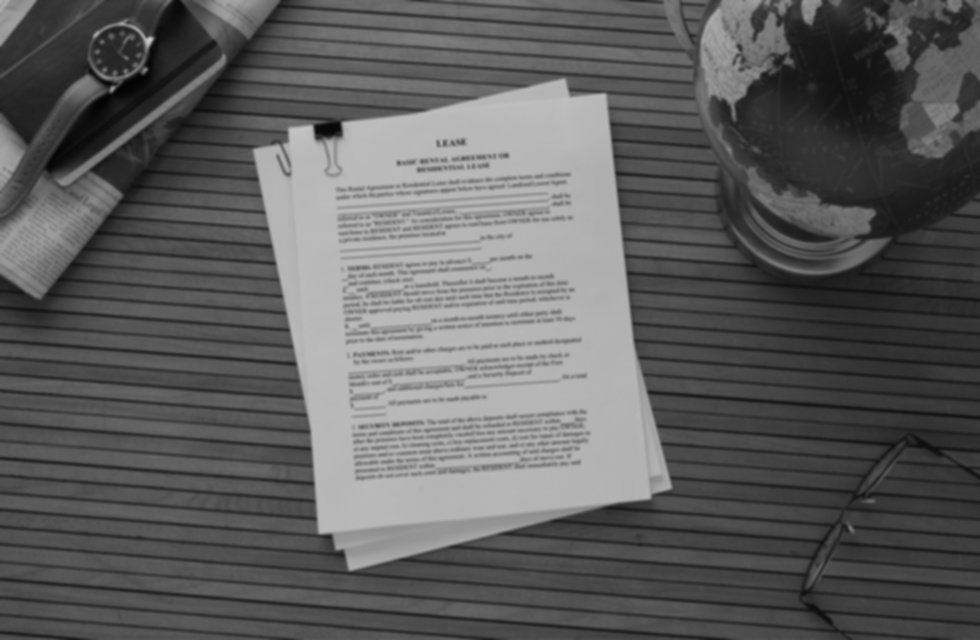
**Tiền xử lý ảnh cho scan văn bản**

**(Image preprocessing for document scanning)**

Phạm Vũ Anh Quân – 19000470

1. **Bộ lọc Gauss (Gaussian Filter)**

* Là bộ lọc hữu ích nhất, tuy nhiên không phải bộ lọc nhanh nhất.
* Được dùng để làm mờ ảnh và giảm nhiễu, sử dụng lý thuyết hàm Gauss.
* Việc sử dụng bộ lọc Gauss thường là bước đầu tiên để phát hiện cạnh trong ảnh.
* Khi làm việc với hình ảnh, ta cần sử dụng hàm Gaussian hai chiều: với là độ lệch chuẩn.
* có vai trò kiểm soát sự biến thiên xung quanh giá trị trung bình của nó.
* 1 số đặc điểm của bộ lọc Gauss:
  + Hệ số của bộ lọc giảm dần khi khoảng cách tới trung tâm bộ lọc tăng.
  + Với giá trị lớn, ảnh sẽ được làm mờ trên một bán kính rất rộng vì vậy lớn cần đi kèm với kích thước bộ lọc lớn.
* Nhân tích chập ảnh ban đầu với bộ lọc sẽ cho ra ảnh kết quả.
* Ví dụ:

***Ảnh gốc*** 

***Ảnh kết quả với kernel\_size=5 và sigma=1.4***

* Mã giả (Pseudocode):

**GaussianFilter**(image, kernel\_size, sigma)

size = int(size/2);

x, y = mesh\_grid[-size:size+1, -size:size+1];

a = 1.0/(2.0 \* pi \* power(sigma, 2));

kernel = a \* exp(-((power(x, 2) + power(y, 2))/(2.0 \* power(sigma, 2)));

blurred\_image = convolution(image, kernel);

return blurred\_image;

1. **Unsharp masking**

* Unsharp masking là kỹ thuật làm nét ảnh phổ biến.
* Các bước của kỹ thuật Unsharp masking:
  + Áp dụng bộ lọc Gauss cho ảnh gốc thu được ảnh .
  + Thực hiện phép thu được mặt nạ cạnh .
  + Nhân với trọng số rồi cộng với , ta thu được ảnh kết quả .
* Trọng số là mức độ tương phản được thêm vào các cạnh, nó sẽ quyết định mức độ nét của ảnh kết quả.
  + nhỏ thì các cạnh sẽ khá phù hợp.
  + lớn thì ảnh kết quả sẽ có các cạnh rất sáng và rất tối.
* Tổng quát lại:
* Ví dụ:

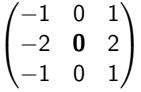
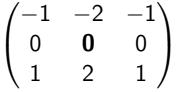
A picture containing text, table

Description automatically generated

***Ảnh kết quả với a=1 và bộ lọc Gauss kích thước 5x5, sigma=1.4***

1. **Toán tử Sobel (Sobel Operator)**

* Toán tử phát hiện biên cạnh dựa trên Gradient bậc nhất
* Sử dụng 2 ma trận lọc và có kích thước 3x3, với là chuyển vị của .

* Nhân tích chập ảnh với 2 ma trận lọc trên sẽ cho ra 2 ảnh tương ứng là và là đạo hàm bậc nhất theo hướng ngang và dọc.
* Tính toán Gradient: và hướng của Gradient: .
* luôn vuông góc với các cạnh. Nó được làm tròn thành 1 trong 4 góc đại diện cho các hướng dọc, ngang và 2 hướng chéo, tương ứng là 0, 90, 45 và 135 độ.
* Ví dụ: Text

  Description automatically generated
* Mã giả (Pseudocode):

**Sobel**(image)

Kx = [[-1,0,1], [-2,0,2], [-1,0,1]];

Ky = transpose(Kx);

Gx = convolution(image, Kx);

Gy = convolution(image, Ky);

gradient = hypot(Gx, Gy);

theta = arctan2(Gy, Gx));

angle = theta \* 180.0 / pi;

return (gradient, angle);

1. **Phương pháp phát hiện biên cạnh Canny (Canny Edge Detection)**

* Phương pháp Canny là một phương pháp phát hiện biên cạnh phổ biến và có hiệu quả rất cao tuy nhiên nó khá phức tạp do là thuật toán nhiều bước.
* Các bước của phương pháp Canny:
  + Bước 1: Ta sử dụng bộ lọc Gauss với kích thước 5x5 để làm mịn ảnh .
  + Bước 2: Tính toán Gradient để phát hiện cường độ và hướng của cạnh.
    - Do các cạnh tương ứng với sự thay đổi của cường độ pixel nên cách dễ nhất để phát hiện nó là áp dụng các bộ lọc làm nổi bật sự thay đổi cường độ theo cả 2 hướng ngang và dọc.
    - Sử dụng bộ lọc Sobel để làm công việc trên và thu về được và .
  + Bước 3: Loại bỏ non-maximum để làm mỏng các cạnh.
    - Với mỗi pixel trên ảnh cường độ gradient, ta kiểm tra xem nó có phải là cực đại cục bộ trong vùng lân cận của nó theo hướng của gradient hay không. Nếu nó là lớn nhất, ta bỏ qua, còn nếu không phải thì đặt giá trị của nó thành 0.
  + Bước 4: Hysteresis Thresholding
    - Sử dụng 2 mức ngưỡng cao và thấp.
      * Mức ngưỡng cao dùng để tìm điểm bắt đầu của cạnh.
      * Mức ngưỡng thấp dùng để xác định hướng đi của cạnh dựa vào các điểm ảnh liên tiếp có giá trị lớn hơn mức ngưỡng này.
      * Mọi điểm ảnh có giá trị nhỏ hơn mức ngưỡng thấp sẽ bị loại bỏ.
* Ví dụ:

Text

Description automatically generated

1. **Khoanh vùng**

* Từ Edge map thu được nhờ phương pháp Canny, ta tìm ra được các đường viền (contour).
* Ta tìm contour của văn bản dựa theo:
  + Diện tích của contour là lớn nhất.
  + Contour Approximation có số đỉnh bằng 4.
* Ví dụ: A piece of paper on a table

  Description automatically generated with medium confidence

Todo:

* Hough transform
* Perspective transform

Tài liệu tham khảo

* [Smoothing Images — OpenCV 2.4.13.7 documentation](https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html#gaussian-filter)
* [OpenCV: Canny Edge Detection](https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html)
* So sánh một số phương pháp phát hiện biên - Nguyễn Vĩnh An, Tạp chí Khoa học ĐHQGHN: Khoa học Tự nhiên và Công nghệ, Tập 31, Số 2 (2015) 1-7