REDUCE GLARE FOR IMAGE PROCESSING

I. ABSTRACT:

Với các ảnh có đèn xe chói/lóa, thường có độ tương phản cao, độ sáng cao ở những vùng ảnh lóa, còn lại đa số điểm ảnh khác có cường độ thấp.

Strategy:

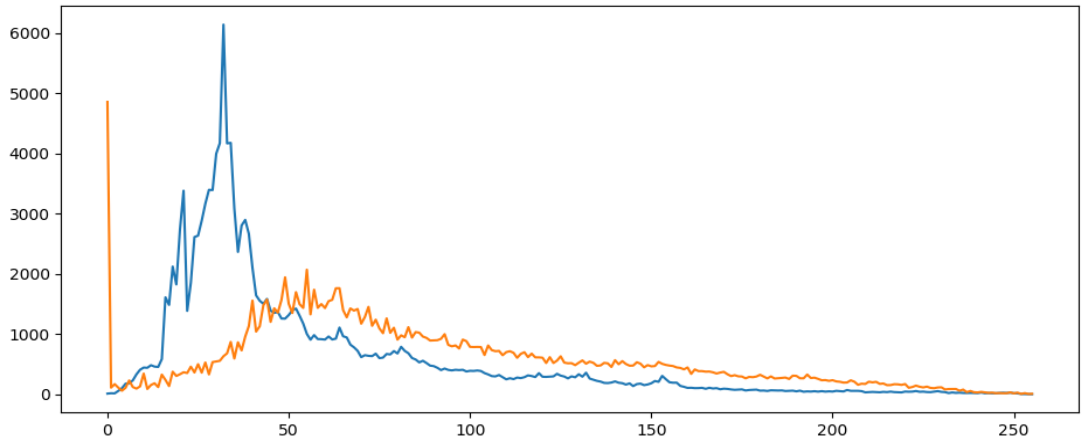
1. Giảm độ lóa của đèn xe.
2. Tăng độ sáng của vật thể ở vùng tối → Dễ nhận diện xe/object hơn.

Method: Sử dụng thuật toán bù sáng như Histogram Equalization, Gamma Correction,…

Giải thuật bù sáng tỏ ra không hiệu quả đối với ảnh có độ tương phản quá chênh lệch.

Dead case of Histogram Equation: Ảnh có tông màu tối nhưng vẫn bị lóa ở một vài chỗ ---(Áp dụng CLAHE)--→ cho ra ảnh bị bù sáng ở những nơi không cần thiết, phá vỡ cấu trúc của ảnh.

Ảnh gốc và ảnh áp dụng CLAHE với contrast limit = 10.0 và title grid = 8x8



Histogram của 2 ảnh trên. Đường xanh của ảnh gốc. Đường cam của ảnh áp dụng CLAHE.

Gamma Correction: Có thể làm rõ vật trong vùng tối nhưng không giảm lóa.
(chi tiết ở file [img-processing] Gamma Correction.docx)

II. ĐỀ XUẤT GIẢI THUẬT:

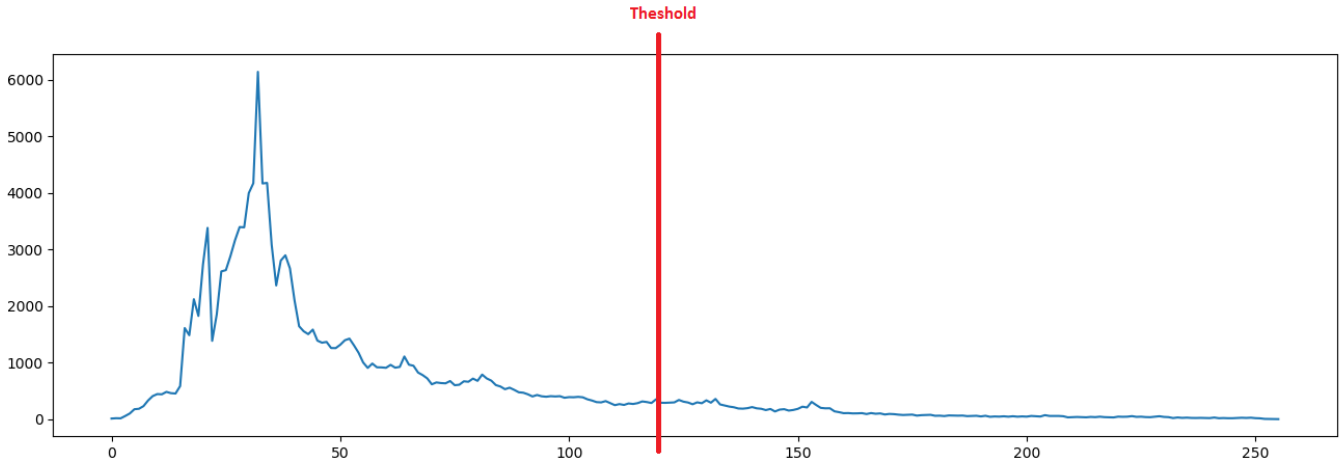
Strategy:

1. Tăng độ sáng của vùng ảnh tối
2. Giảm độ sáng của vùng ảnh lóa

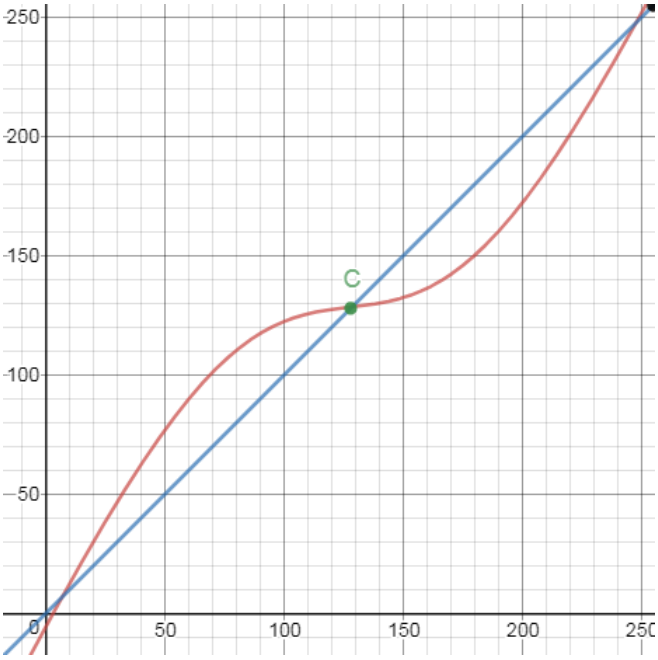
Method:

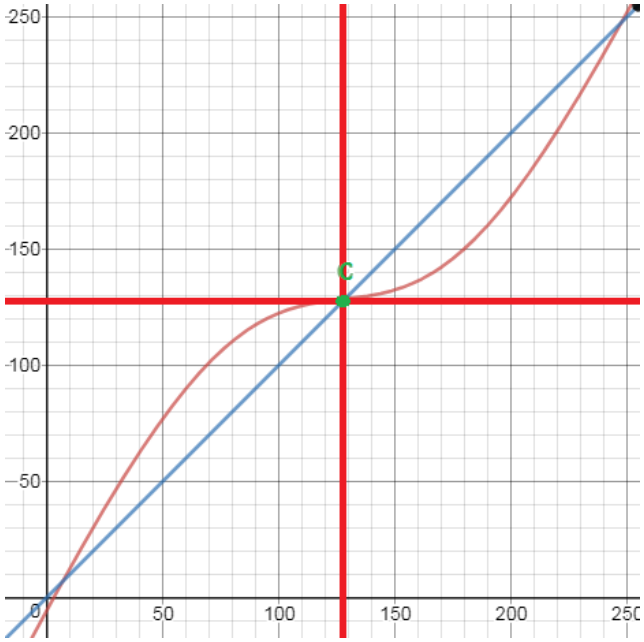
Đặt ra một ***threshold***:

- Vùng ảnh có cường độ dưới threshold: cần tăng cường độ của vùng ảnh đó, để gần threshold hơn. *(Thỏa mãn strategy 1).*
- Vùng ảnh có cường độ trên threshold: cần giảm cường độ của vùng ảnh. *(Thỏa mãn strategy 2).*



CÁCH LÀM 1: Sử dụng hàm non-linear để biến đổi.





Đường màu xanh: x cường độ gốc, y cường độ gốc

Đường màu cam: x cường độ gốc, y cường độ sau khi áp dụng hàm non-linear

Đường màu đỏ: threshold tương ứng

Hàm non-linear đề xuất để có đồ thị gần giống với hình trên:

$$f(x) = 2x - \frac{255}{1 + \frac{31}{30}^{threshold - x}} \quad (*)$$

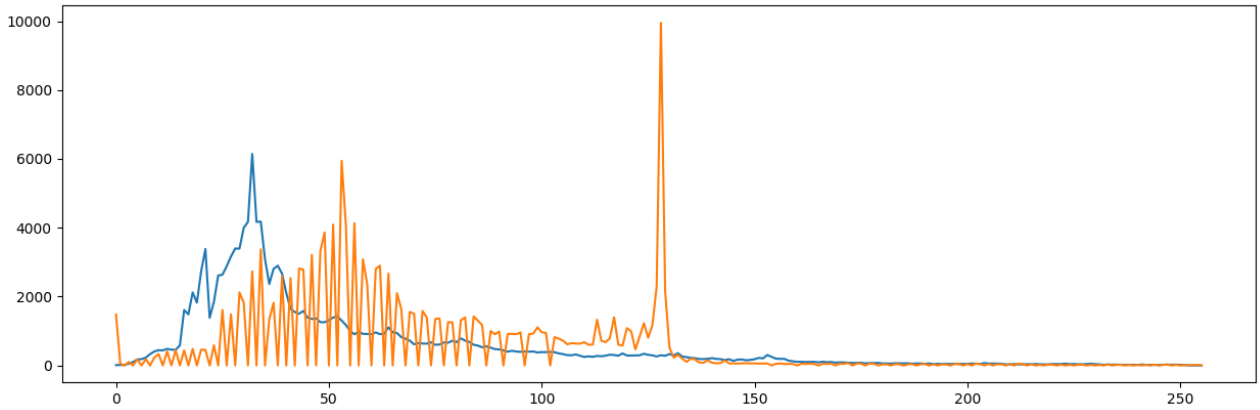
Implementation:

```python
def adjust_ins(image, threshold = 128, alpha = 1.033):

    table = np.array([2*i - 255.0/(1 + alpha ** (threshold - i))
        for i in np.arange(0, 256)]).astype("uint8")

    return cv2.LUT(image, table)
```

Implementation in python

Evaluation:

Ảnh gốc và ảnh áp dụng hàm non-linear.

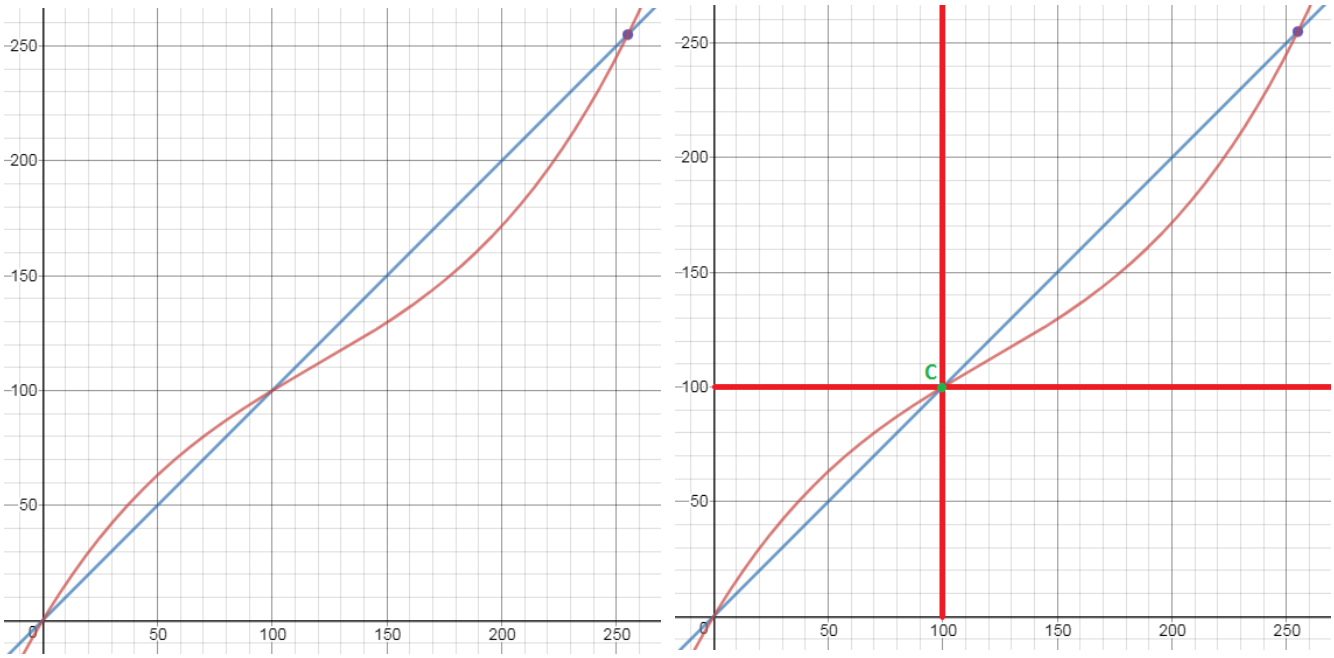


Histogram của 2 ảnh trên. ***Đường xanh của ảnh gốc.*** ***Đường cam của hàm non-linear.***

Hàm non-linear (*) biến đổi những điểm xung quanh threshold thành rất sát với threshold.

⇨ **Cần tìm 1 hàm thích hợp hơn.**

<u>CÁCH LÀM 2:</u> Sử dụng hàm polynomial để biến đổi.



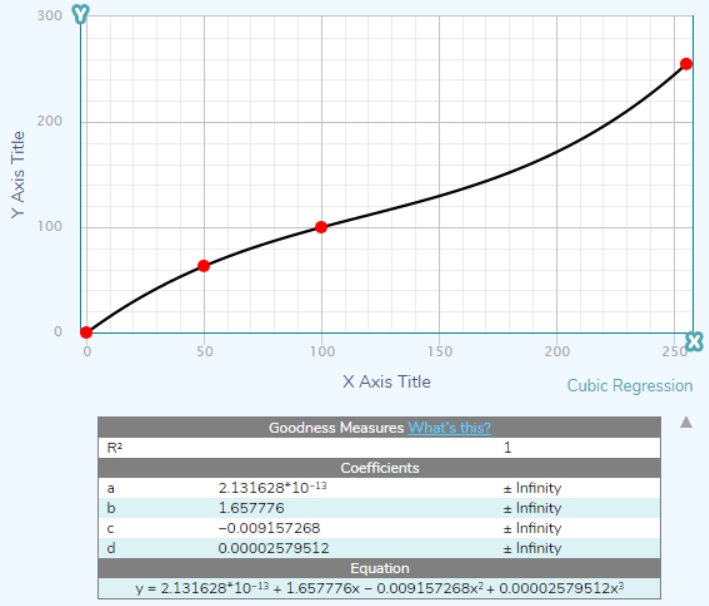
Đường màu xanh: x cường độ gốc, y cường độ gốc
Đường màu cam: x cường độ gốc, y cường độ sau khi áp dụng hàm polynomial
Đường màu đỏ: threshold tương ứng

Sử dụng trang web: https://mycurvefit.com/

Input data:

X Axis Title	Y Axis Title
255	255
100	100
0	0
50	63.22

Hàm gợi ý:



Goodness Measures What's this?		
R^2		1
Coefficients		
a	$2.131628*10^{-13}$	± Infinity
b	1.657776	± Infinity
c	−0.009157268	± Infinity
d	0.00002579512	± Infinity
Equation		
$y = 2.131628*10^{-13} + 1.657776x - 0.009157268x^2 + 0.00002579512x^3$		

Hàm non-linear đề xuất để có đồ thị gần giống với hình trên:

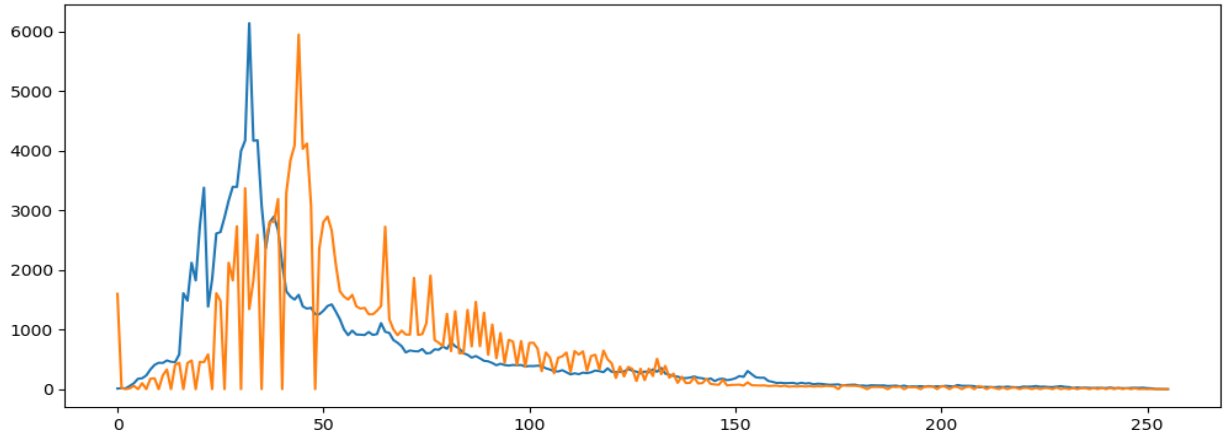
$$f(x) = 1.657766x - 0.009157128x^2 + 0.00002579473x^3$$

Implementation:

```
def adjust_ins2(image):
    table = np.array([
        1.657766*i-0.009157128*(i**2) + 0.00002579473*(i**3)
        for i in np.arange(0, 256)]).astype("uint8")
    return cv2.LUT(image, table)
```

Evaluation:

Ảnh gốc và ảnh áp dụng hàm polynomial.



Histogram của 2 ảnh trên. ***Đường xanh của ảnh gốc.*** ***Đường cam của hàm polynomial.***

Hàm polynomial thực hiện đúng như kỳ vọng: giảm cường độ ở vùng sáng và tăng ở vùng tối.

Ảnh sau khi áp dụng hàm polynomial có ***độ tương phản thấp***, do cường độ bị phân bố lại gần theshold.

III. TỐI ƯU HÓA HIỆU SUẤT

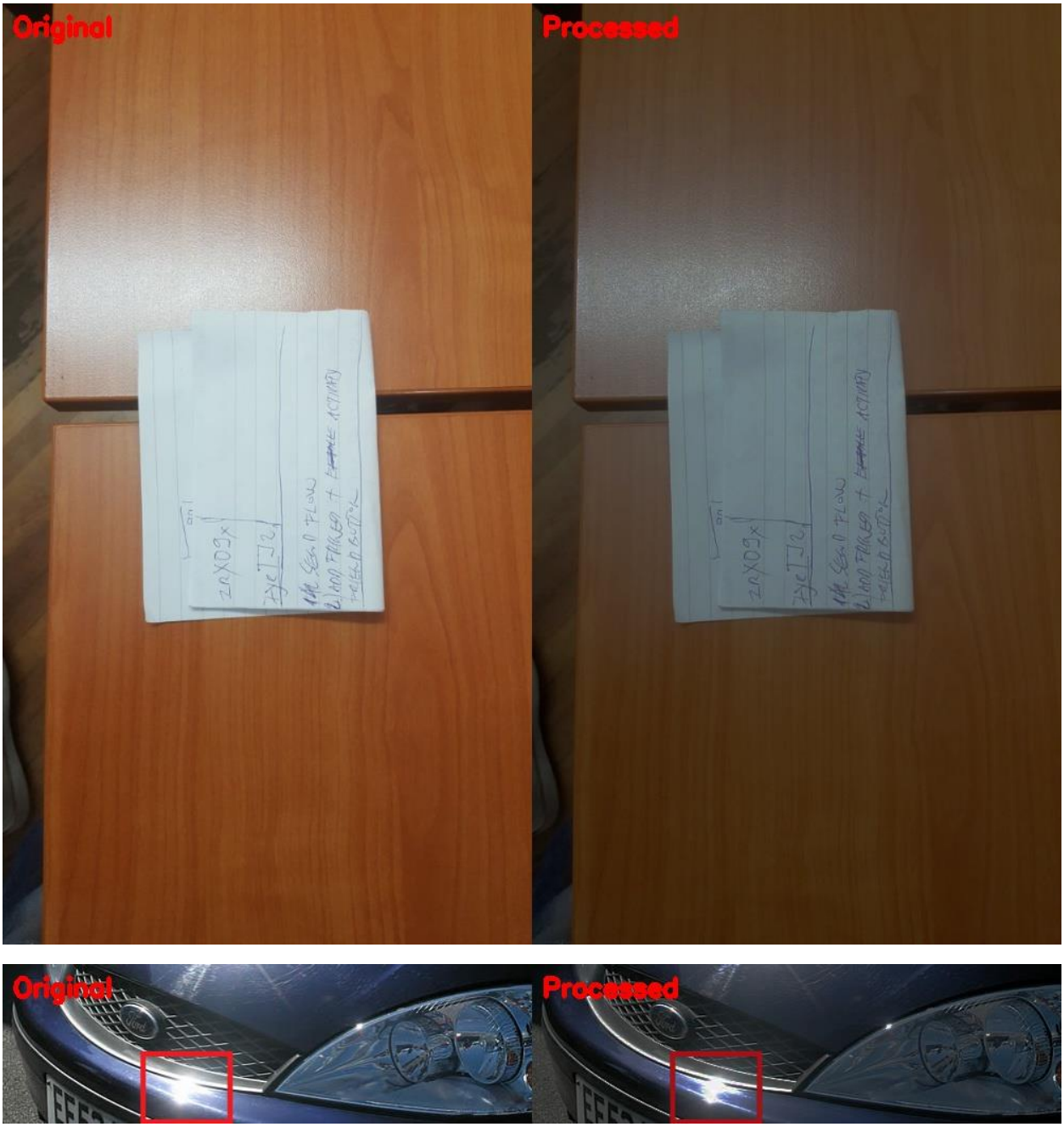
Sử dụng 4 filter từ 2 hàm polynomial và gamma correction:

1. **Polynomial filter (phần II, cách 2)**
   ⇨ Tái phân phối cường độ ánh sáng cho gần threshold = 100.
   ⇨ Độ tương phản thấp
2. **Gamma Correction**: gamma < 1
   ⇨ Tăng độ tương phản
   ⇨ Bù sáng
3. **Lặp lại 2 bước trên.**

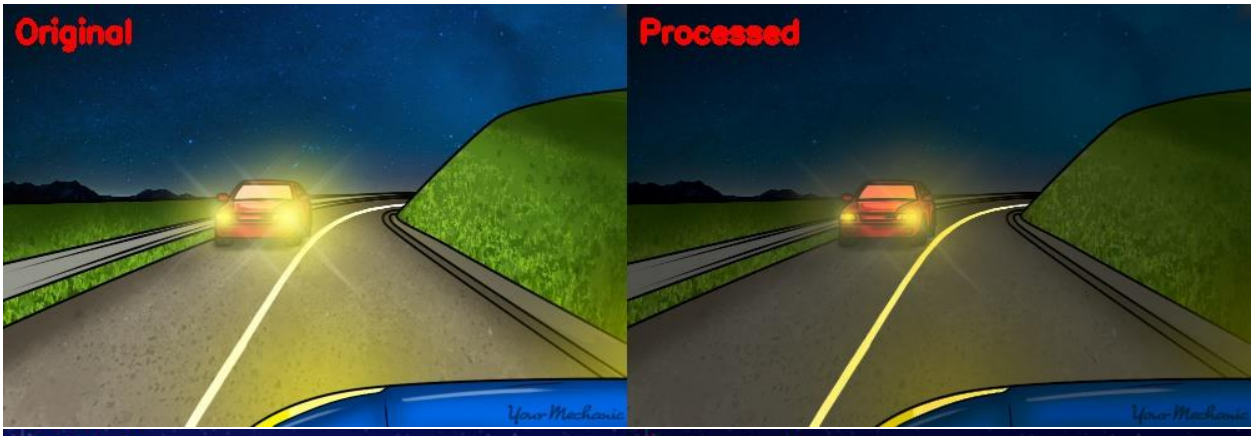
Áp dụng:

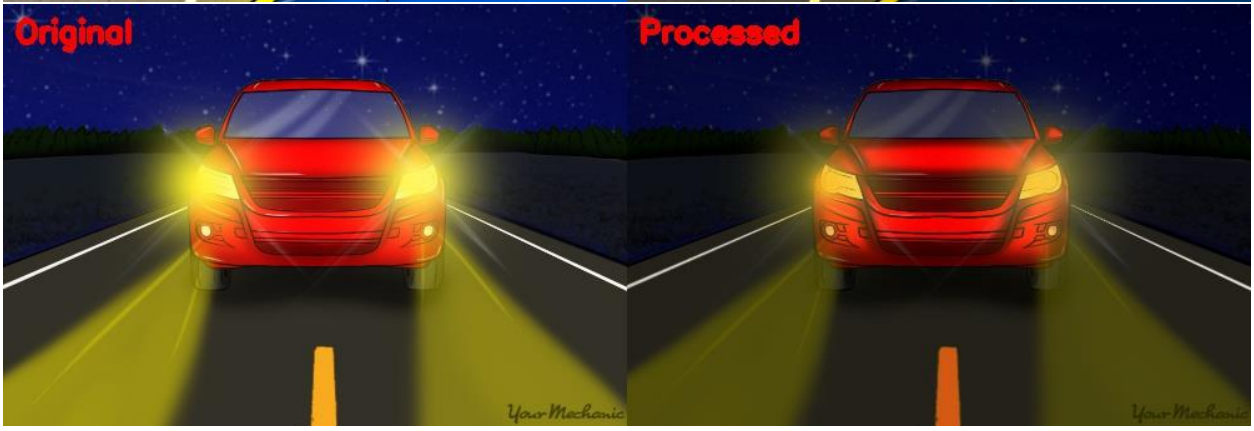
Áp dụng 4 filter vào ảnh gốc cho ra hình giảm độ chói trong môi trường tương phản cao.

Showcase:




Original
Processed


Original
Processed


Original
Processed