PDF_CDF_technical_report

introduction

Grayscale 이미지의 PDF와 CDF를 계산하고, plot한다

코드 구현

PDF_CDF.cpp

main 함수

```
1)
    Mat input = imread("input.jpg", IMREAD_COLOR);
    Mat input_gray;

    cvtColor(input, input_gray, COLOR_RGB2GRAY);    // convert RGB to Graysca

    // PDF, CDF txt files
    FILE *f_PDF, *f_CDF;

    fopen_s(&f_PDF, "PDF.txt", "w+");
    fopen_s(&f_CDF, "CDF.txt", "w+");

2)
    // each histogram
    float *PDF = cal_PDF(input_gray);       // PDF of Input image(Grayscale) : [L]
    float *CDF = cal_CDF(input_gray);       // CDF of Input image(Grayscale) : [L]

3)
    for (int i = 0; i < L; i++) {
        // write PDF, CDF
        fprintf(f_PDF, "%d\t%f\n", i, PDF[i]);
        fprintf(f_CDF, "%d\t%f\n", i, CDF[i]);
    }
```

1. input 이미지 로드
2. cal_PDF, cal_CDF 함수를 사용해서 PDF와 CDF 계산
3. fprintf 사용해 txt 파일로 저장

hist_func.h

PDF 계산 함수

```
float *cal_PDF(Mat &input) {

    int count[L] = { 0 };
    float *PDF = (float*)calloc(L, sizeof(float));

1)
    // Count
    for (int i = 0; i < input.rows; i++)
        for (int j = 0; j < input.cols; j++)
            count[input.at<G>(i, j)]++;
2)
    // Compute PDF
    for (int i = 0; i < L; i++)
        PDF[i] = (float)count[i] / (float)(input.rows * input.cols);

    return PDF;
}
```

1. 전체 픽셀의 개수를 count
2. PDF 공식을 통하여 모든 픽셀에 대하여 계산

$$- PDF(r) = n_r/N \ (n_r\text{: \# of pixel with intensity } r)$$

CDF 계산 함수

```
float *cal_CDF(Mat &input) {
```

```
    int count[L] = { 0 };
    float *CDF = (float*)calloc(L, sizeof(float));

1)
    // Count
    for (int i = 0; i < input.rows; i++)
        for (int j = 0; j < input.cols; j++)
            count[input.at<G>(i, j)]++;
2)
    // Compute CDF
    for (int i = 0; i < L; i++) {
        CDF[i] = (float)count[i] / (float)(input.rows * input.cols);

        if (i != 0)
            CDF[i] += CDF[i - 1];
    }

    return CDF;
}
```

1. 전체 픽셀의 개수를 센다
2. CDF 공식을 통하여 누적 확률 분포 계산

$$- CDF(r) = \sum_{k=0}^{r} PDF(k) = \sum_{k=0}^{r} n_k / N$$

pdf_cdf_transfer.py

```
import matplotlib.pyplot as plt
import numpy as np

pdf = np.loadtxt("PDF.txt", delimiter="\t", usecols=1)
cdf = np.loadtxt("CDF.txt", delimiter="\t", usecols=1)

x = np.arange(256)

#pdf 그리기
```

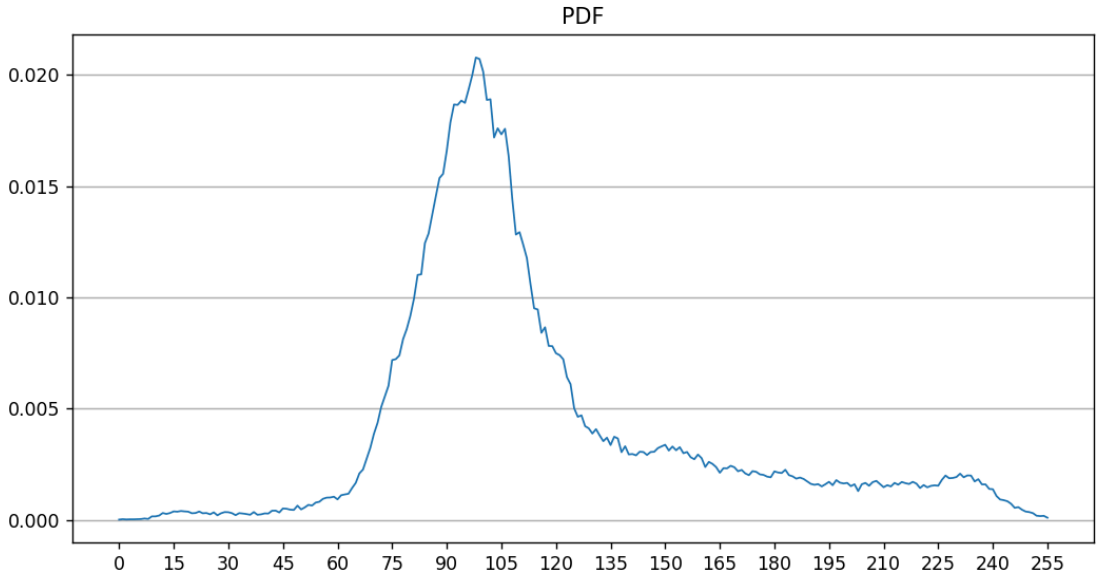
```
plt.figure(figsize=(10,5))
plt.plot(x, pdf, linewidth=1)
plt.xticks(range(0, 256, 15))
plt.grid(True, axis='y')
plt.title("PDF")
plt.show()

#cdf 그리기
plt.figure(figsize=(10, 5))
plt.plot(x, cdf, linewidth=1)
plt.xticks(range(0,256,15))
plt.grid(True, axis='y')
plt.title("CDF")
plt.show()
```

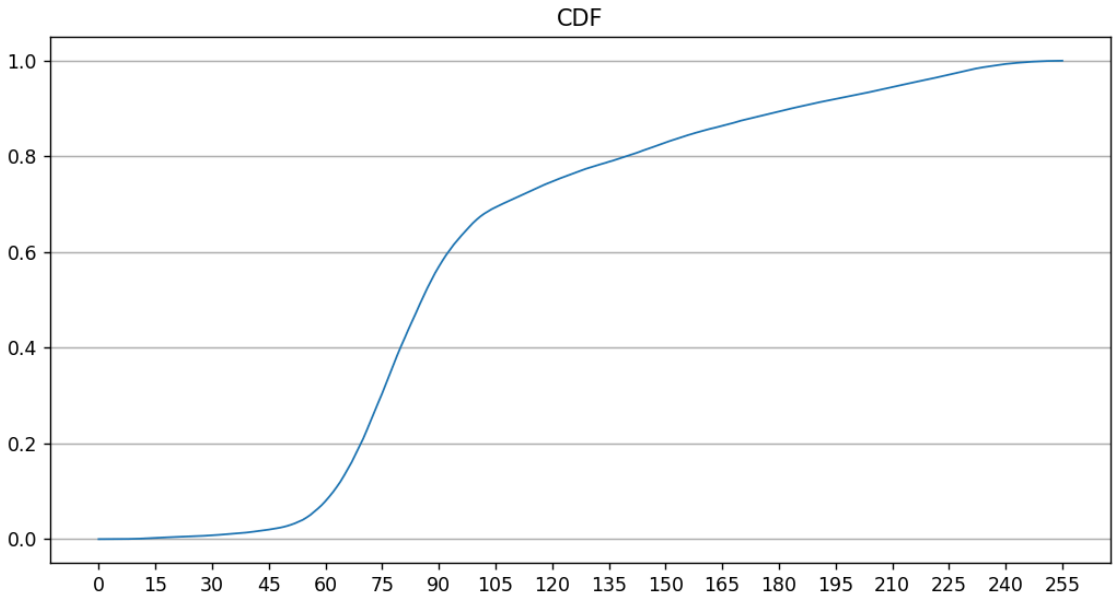
python의 matplotlib을 사용해서 plot했다.

x축은 intensity 값, y축은 각 intensity의 확률을 나타냈다.

Result

plot한 결과는 다음과 같다.





PDF는 이미지의 intensity의 비율을 알 수 있다.

PDF 분석 결과, 대부분의 intensity가 90과 105 사이에 몰려있다. 그렇기 때문에 전체적인 이미지의 대비가 낮다고 볼 수 있다.

CDF란, 밝기값 i 이하의 픽셀이 전체에서 차지하는 누적 비율을 말한다.

CDF 분석 결과, 75부터 갑자기 급상승 하는 것을 볼 수 있다. 거의 수직 구조이기 때문에 이미지 밝기 분포의 구조를 알 수 있는데, 상대적으로 어두운 영역보다는 밝은 영역이 많다고 할 수 있다.

결론

이미지 stretching, 이미지 equalization, 이미지 matching을 통해 이미지의 대비를 개선 시켜야 한다!