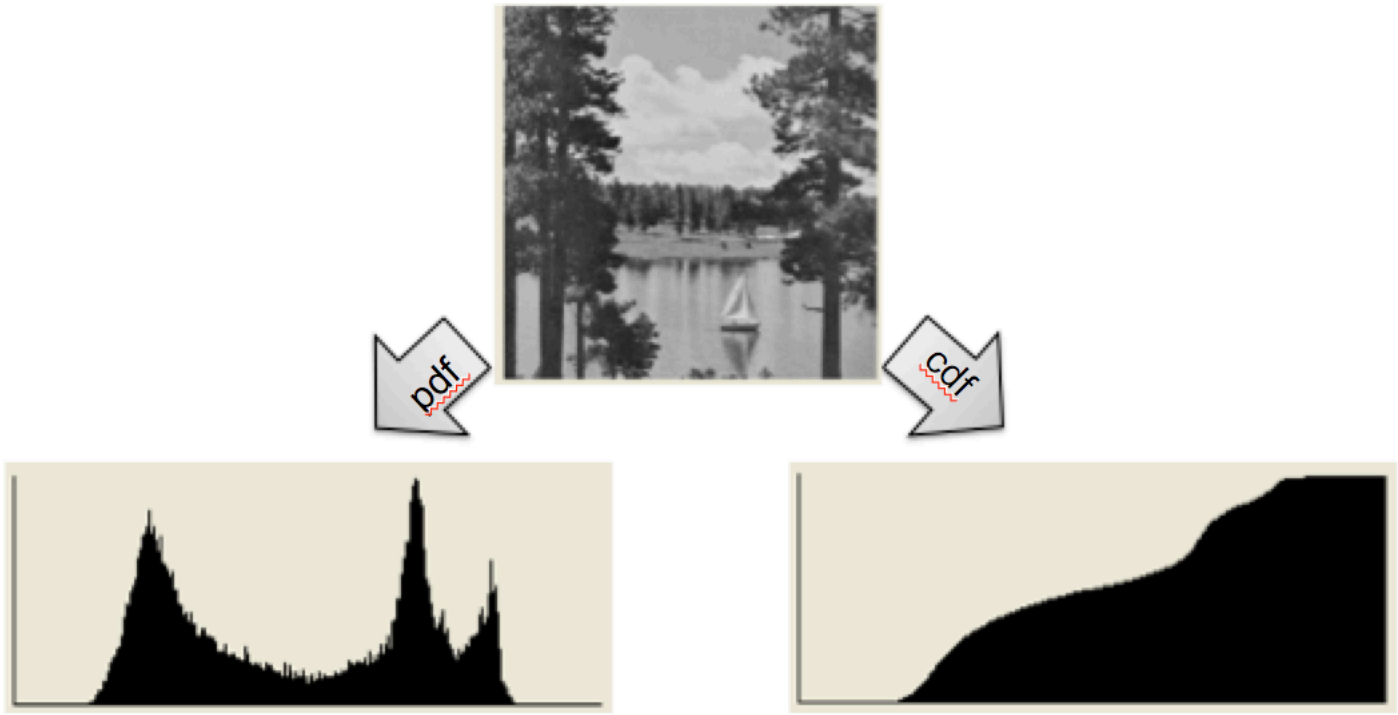
Histogram
Assignment 2

Gwenaelle Cunha Sergio
ABR Lab – BEP
KNU 2014.1

Histogram

- Graphical representation of the distribution of data

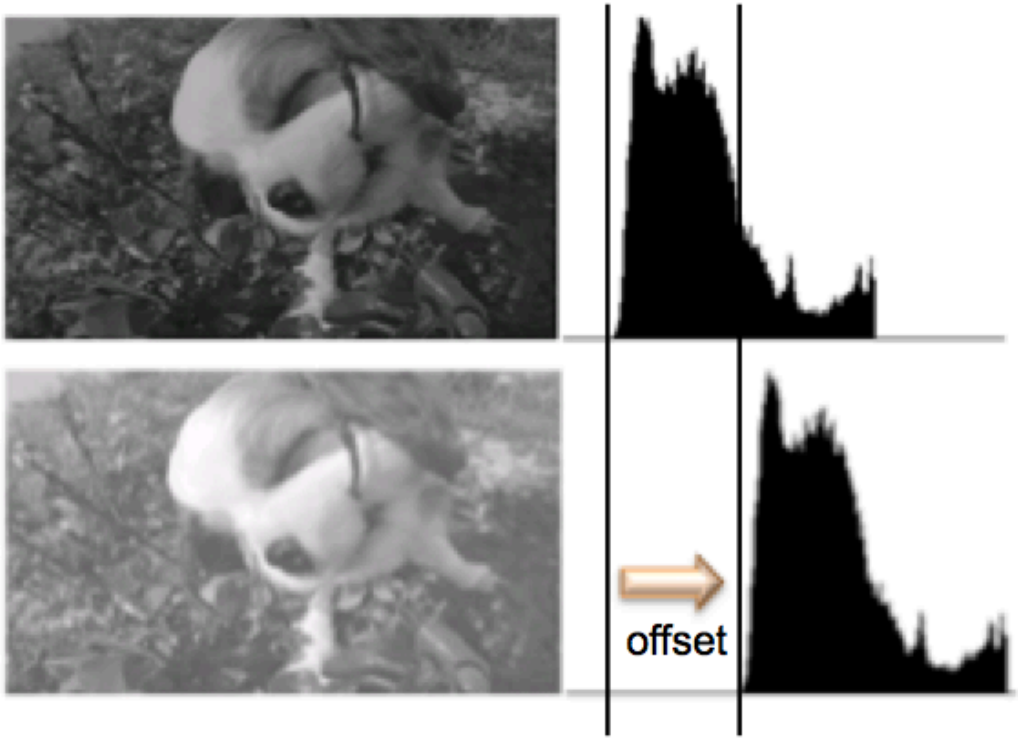


Histogram

- Graphical representation of the distribution of data
- Importance: find properties and pre-process images
- Applications: sliding, stretching, shrink, equalize, quantization, etc.

Sliding

- Moves the histogram by an offset value
- Lighter picture

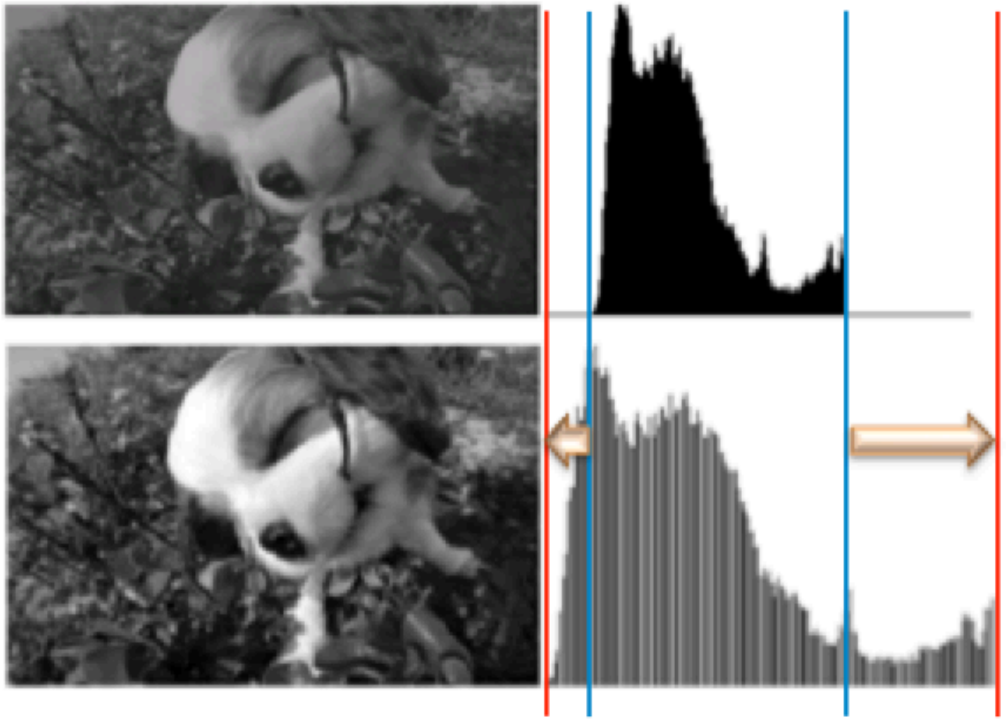


Sliding - Code

```
int offset = 30;
int sliding = 0;
for (int i = 0; i < mat.rows; i++){
    for (int j = 0; j < mat.cols; j++){
        sliding = mat.at<uchar>(i,j) + offset;
        if (sliding < 0) sliding = 0;
        else if (sliding > 255) sliding = 255;
        slideImage.at<uchar>(i,j) = sliding;
    }
}
```

Stretching

- Normalizes the histogram of the image to values between 0 and 255
- Brighter picture



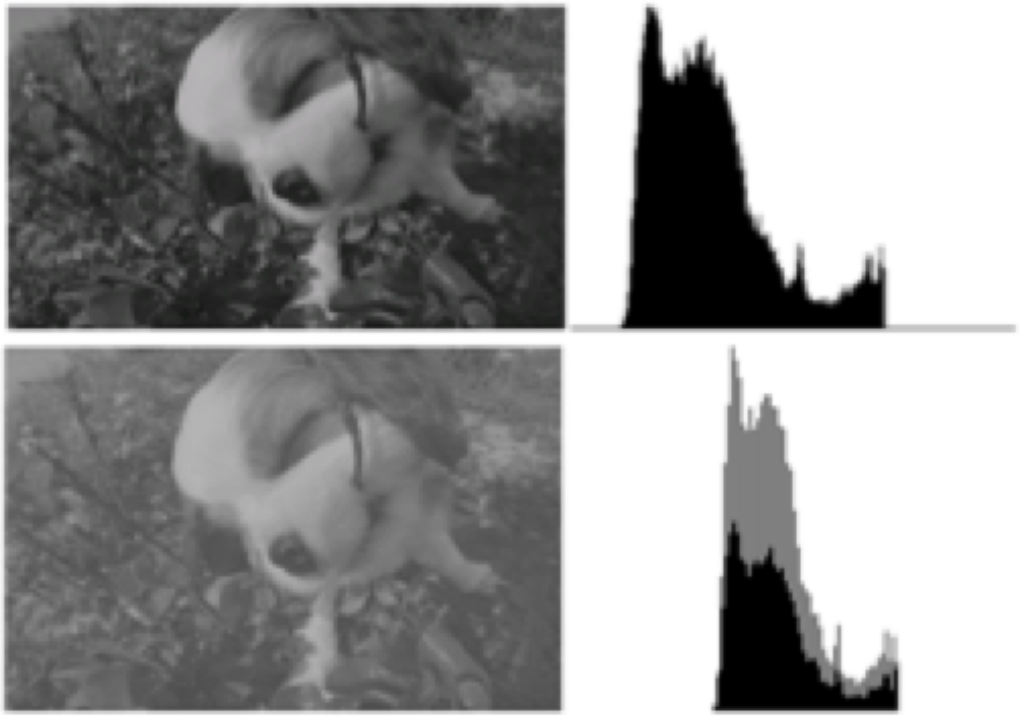
Stretching - Code

```
int min = 0, max = 255;
int maxGray = 0, minGray = 0;
for (int i = 0; i < mat.rows; i++){
        for (int j = 0; j < mat.cols; j++){
                if (mat.at<uchar>(i,j) > maxGray) maxGray = mat.at<uchar>(i,j);
                if (mat.at<uchar>(i,j) < minGray) minGray = mat.at<uchar>(i,j);
        }
}

for (int i = 0; i < mat.rows; i++){
        for (int j = 0; j < mat.cols; j++){
                stretchImage.at<uchar>(i,j) =
                        cvRound((max-min)*(mat.at<uchar>(i,j) - minGray)/
                                                        (maxGray - minGray));
        }
}
```

Shrink

- Similar to stretch, but the minimum and maximum values are customizable



Shrink - Code

```
int min = 80, max = 230;
int maxGray = 0, minGray = 0;
for (int i = 0; i < mat.rows; i++){
        for (int j = 0; j < mat.cols; j++){
                if (mat.at<uchar>(i,j) > maxGray) maxGray = mat.at<uchar>(i,j);
                if (mat.at<uchar>(i,j) < minGray) minGray = mat.at<uchar>(i,j);
        }
}

for (int i = 0; i < mat.rows; i++){
        for (int j = 0; j < mat.cols; j++){
                stretchImage.at<uchar>(i,j) =
                        cvRound((max-min)*(mat.at<uchar>(i,j) - minGray)/
                                                        (maxGray - minGray));
        }
}
```