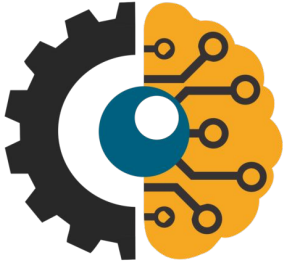
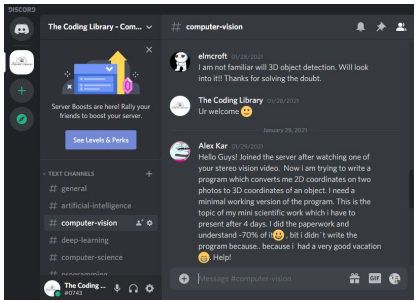


THE CODING LIBRARY

Computer Vision

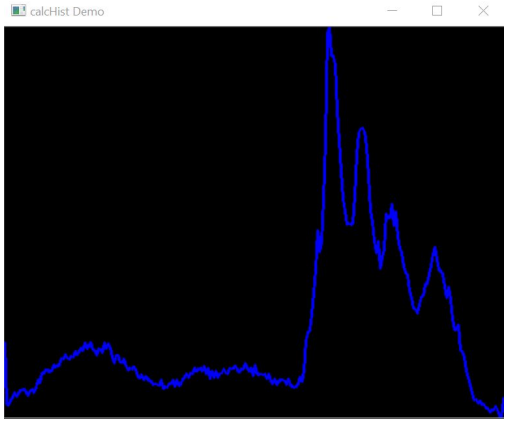
Histograms

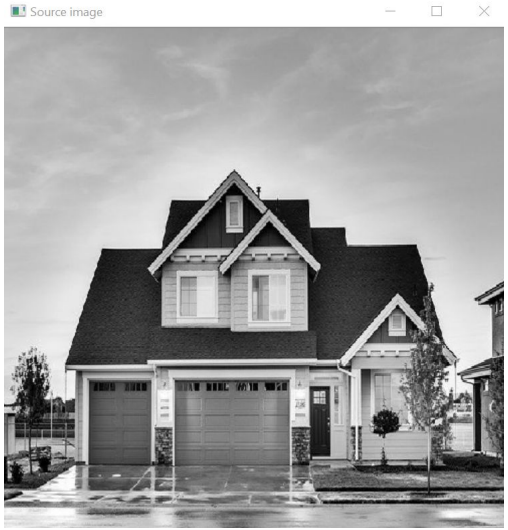


Discord Link in Description

1D Histograms

- Global information
- Not unique
- Could be useful for classification



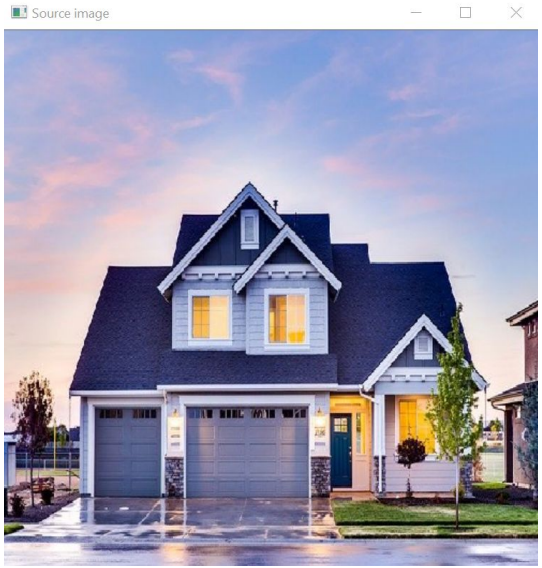


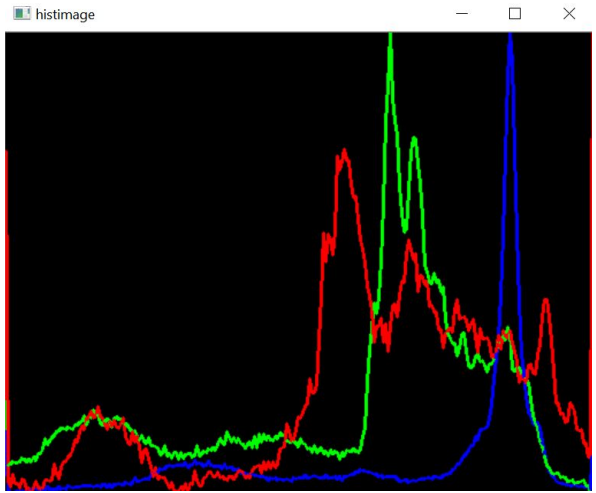
1D Histograms in OpenCV

THE CODING LIBRARY

```
MatND histogram;
int histSize = 256;
const int* channel_numbers = { 0 };
float channel_range[] = { 0.0, 256.0 };
const float* channel_ranges = channel_range;
int number_bins = histSize;

calcHist(&image3, 1, 0, Mat(), histogram, 1, &number_bins, &channel_ranges);
```

1D Histograms - Color Histograms



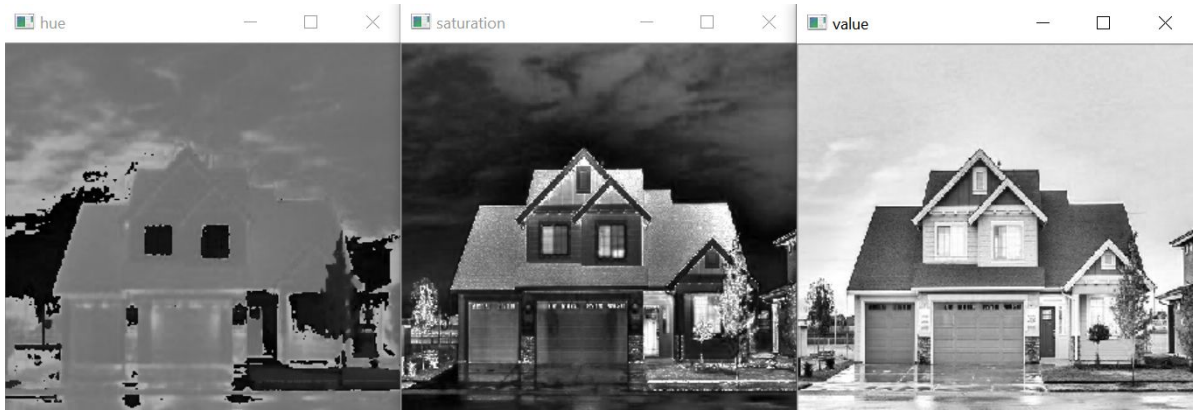


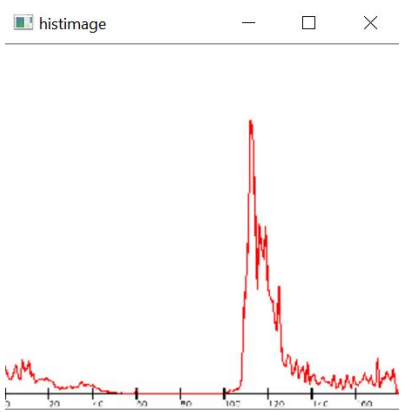
1D Color Histograms in OpenCV

THE CODING LIBRARY

```
vector<Mat> bgr_planes;
split(image3, bgr_planes);

int histSize = 256;
float range[] = { 0, 256 }; //the upper boundary is exclusive
const float* histRange = { range };
bool uniform = true, accumulate = false;

Mat b_hist, g_hist, r_hist;

calcHist(&bgr_planes[0], 1, 0, Mat(), b_hist, 1, &histSize, &histRange, uniform, accumulate);
calcHist(&bgr_planes[1], 1, 0, Mat(), g_hist, 1, &histSize, &histRange, uniform, accumulate);
calcHist(&bgr_planes[2], 1, 0, Mat(), r_hist, 1, &histSize, &histRange, uniform, accumulate);
```

HSV Histogram

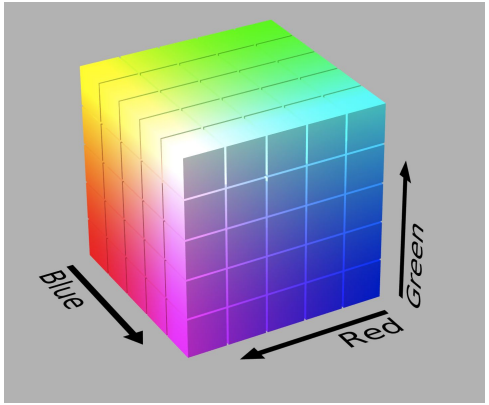




Histogram Application

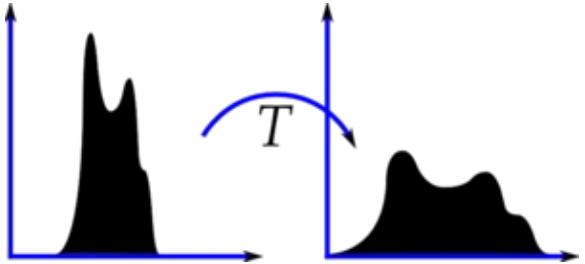
3D Histograms

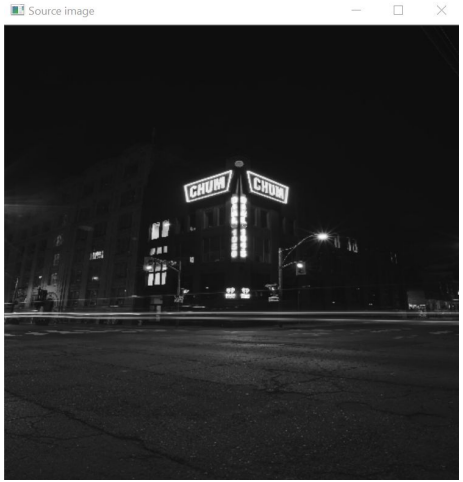
- Channels are not independent
- Better discrimination comes from considiring all channels simultaneously
- Reduce Quantisation
  - 6 bits
  - 4 bits
  - 2 bits

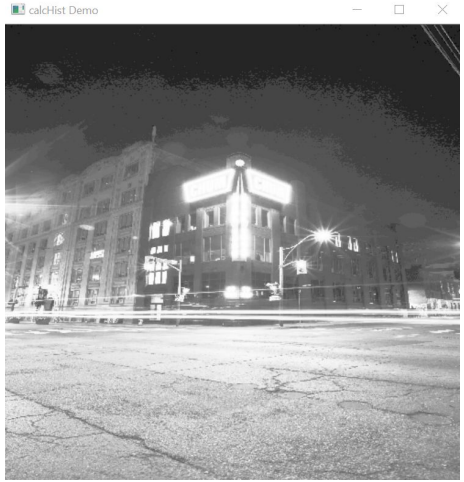


Equalisation

- If an image has insufficient contrast
- Human can distinguish 700-900 greyscales
- Evenly distribute the greyscales
- Normally equalise only the greyscales / luminance







Equalisation in OpenCV

THE CODING LIBRARY

```
//equalize the histogram
Mat histEqualized;
equalizeHist(image4, histEqualized);

imshow("Source image", image4);
imshow("calcHist Demo", histEqualized);
waitKey();
```

Histogram Comparison

THE CODING LIBRARY

```
Correlation: 0.606325
ChiSquare: 88411.6
Intersection: 21492.8
Bhattacharyya: 0.23998
```

Histogram Comparison - Metrics

THE CODING LIBRARY

- To compare two histograms (H_1 and H_2), first we have to choose a *metric* ($d(H_1, H_2)$) to express how well both histograms match.
- OpenCV implements the function **cv::compareHist** to perform a comparison. It also offers 4 different metrics to compute the matching:
  1. **Correlation (CV_COMP_CORREL)**

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}}$$

     where

$$\bar{H_k} = \frac{1}{N} \sum_J H_k(J)$$

     and N is the total number of histogram bins.

  2. **Chi-Square (CV_COMP_CHISQR)**

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

  3. **Intersection (method=CV_COMP_INTERSECT)**

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

  4. **Bhattacharyya distance (CV_COMP_BHATTACHARYYA)**

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H_1} \bar{H_2} N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

Histogram Comparison in OpenCV

THE CODING LIBRARY

```
double histMatchingCorrelation = compareHist(histogram, histogram2, HISTCMP_CORREL);
double histMatchingChiSquare = compareHist(histogram, histogram2, HISTCMP_CHISQR);
double histMatchingIntersect = compareHist(histogram, histogram2, HISTCMP_INTERSECT);
double histMatchingBhattacharyya = compareHist(histogram, histogram2, HISTCMP_BHATTACHARYYA);

cout << "Correlation: " << histMatchingCorrelation << endl;
cout << "ChiSquare: " << histMatchingChiSquare << endl;
cout << "Intersection: " << histMatchingIntersect << endl;
cout << "Bhattacharyya: " << histMatchingBhattacharyya << endl;
```

Histogram Back Projection

- Better approach to selecting colours (Based on samples)
- Histogram the samples
- Normalize the histogram
- Back project the normalized histogram onto an image
- Results in a probability image which indicated the similarity between the image and the sample set

```
Mat backproj;
calcBackProject(&hue, 1, 0, histogram, backproj, &ranges, 1, true);
```