National Tsing Hua University
Department of Electrical Engineering
EE4292 IC Design Laboratory (積體電路設計實驗)
Fall 2021

Implement Edge Detection and Object Detection on Image using HOG (Histogram of Oriented Gradient) Algorithm With Some Famous Filters

利用方向梯度直方圖演算法實現圖像邊緣檢測及物件識別

Team Members: [1]林暄傑、劉亦傑、謝霖泳

Motivation and Objective:

After the department store closes, some thieves will sneak into the store. So, we develop a system that can detect the edges in an image. When a thief sneaks into a store, the edges in the monitor screen will change drastically, indicating that something unusual has happened in the store.

Functionality:

The function we want to implement in our hardware is below:

Normal mode:

1. Extract feature vectors: by calculating gradient, we can obtain the feature vectors.
2. Obtain theta to derive histogram: we can detect the number of objects with histogram.
3. Object detection: detect the target objects, such as people, animals or boxs.

Advanced mode:

Detection with noise figures

method1: median filter

method2: Gaussian denoiser

method3: Non-local means denoiser

Detection with low-resolution figures

method: Image super-resolution via sparse representation

There are normal mode and advanced mode in our circuit. We can compare the result between noise figures, low resolution figures and normal figures.

Specification:

- image: 640*480 pixels
- input: 36 pixels per cycle (each pixel is 8-bit)
- output: 36 pixels pixel per cycle
- timing : 3ns
- area : 750,000 um^2
- cycle count : 10000

Implementation:

1. Survey the algorithms on Internet, and decide to implement classical HOG (Histogram of Oriented Gradient) algorithm
2. image preprocessing
    a. denoise
    b. enhance resolution
3. Compute gradients (cell / block)
4. Use weighted vote to build orientation cells, so that it could construct the histogram.
5. Collect HOG's over detection window
6. The combined vectors are fed to a linear SVM for object/non-object classification (Python based)

Verification:

1. Verify the calculated value with the golden file.
2. *$fwrite* the value into a log file (processed output result).
3. Visualize the result after whole processed data is written. We expect to see the edges being highlighted.

Reference:

[1] ***Histograms of Oriented Gradients for Human Detection***, by Navneet Dalal and Bill Triggs, CVPR 2005
[2] ***Image super-resolution as sparse representation of raw image patches,*** by J. Yang et al, CVPR 2008.
[3] ***A non-local algorithm for image denoising***, CVPR 2005
[4] https://en.wikipedia.org/wiki/Median_filter
[5] https://en.wikipedia.org/wiki/Gaussian_filter