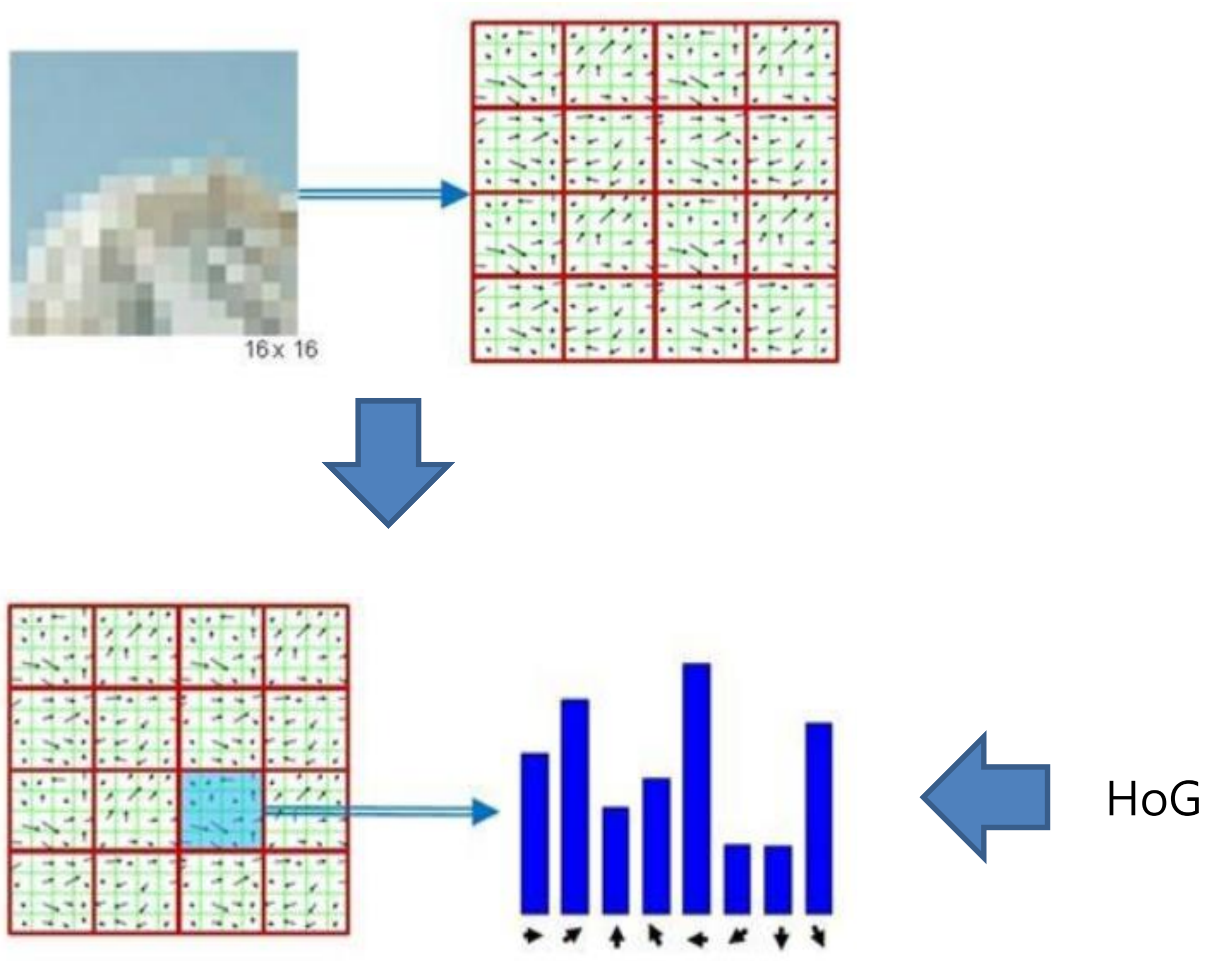
Pedestrian Detection

Sung Soo Hwang

Pedestrian Detection

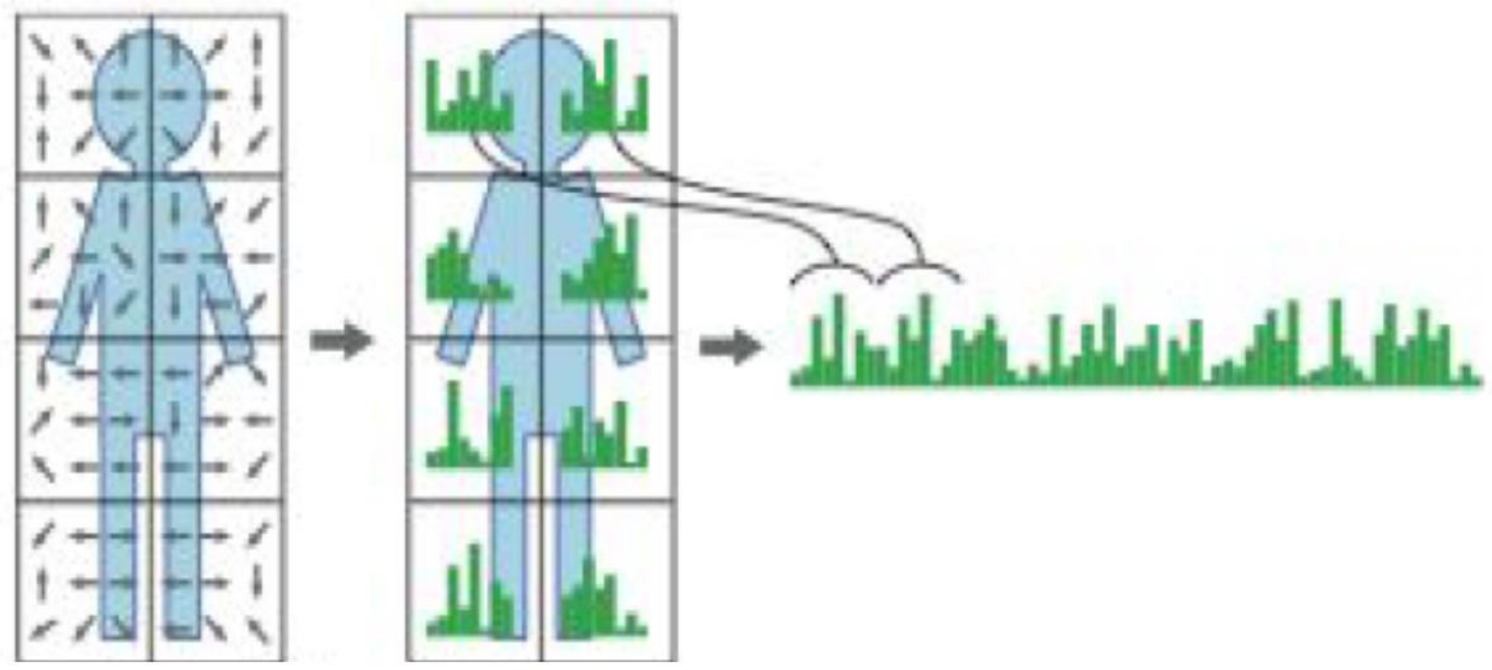
- Feature
  - HoG is used in openCV



http://vision0814.tistory.com/168

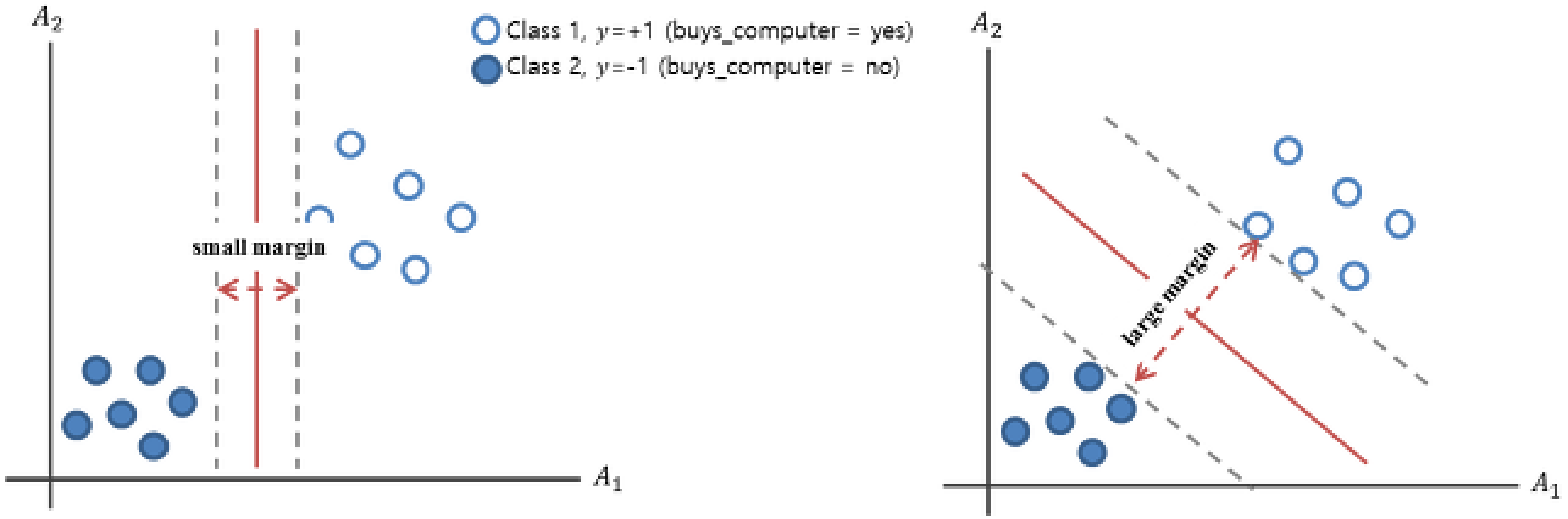
Pedestrian Detection

- Feature
  - Pedestrian representation using HoG



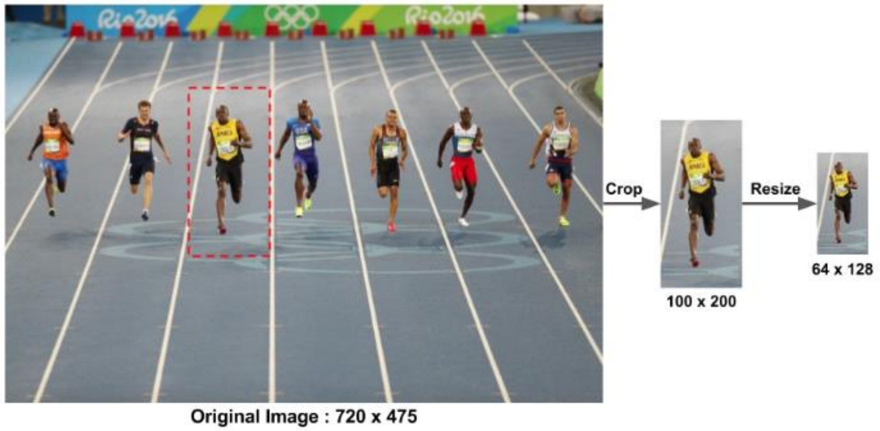
Pedestrian Detection

- Training
  - Support Vector Machine is used in openCV
    - SVM finds maximum marginal hyperplane

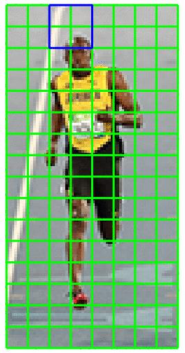

A_2
○ Class 1, y=+1 (buys_computer = yes)
● Class 2, y=-1 (buys_computer = no)
small margin
A_1
A_2
large margin
A_1

Pedestrian Detection

- Design classifiers or do pre-processing/post-processing for higher performance
  - Image patch resizing



  - Histogram normalization



← For handling overall illumination change

  - Also use image pyramid for multi scale detection

Pedestrian Detection

- openCV function

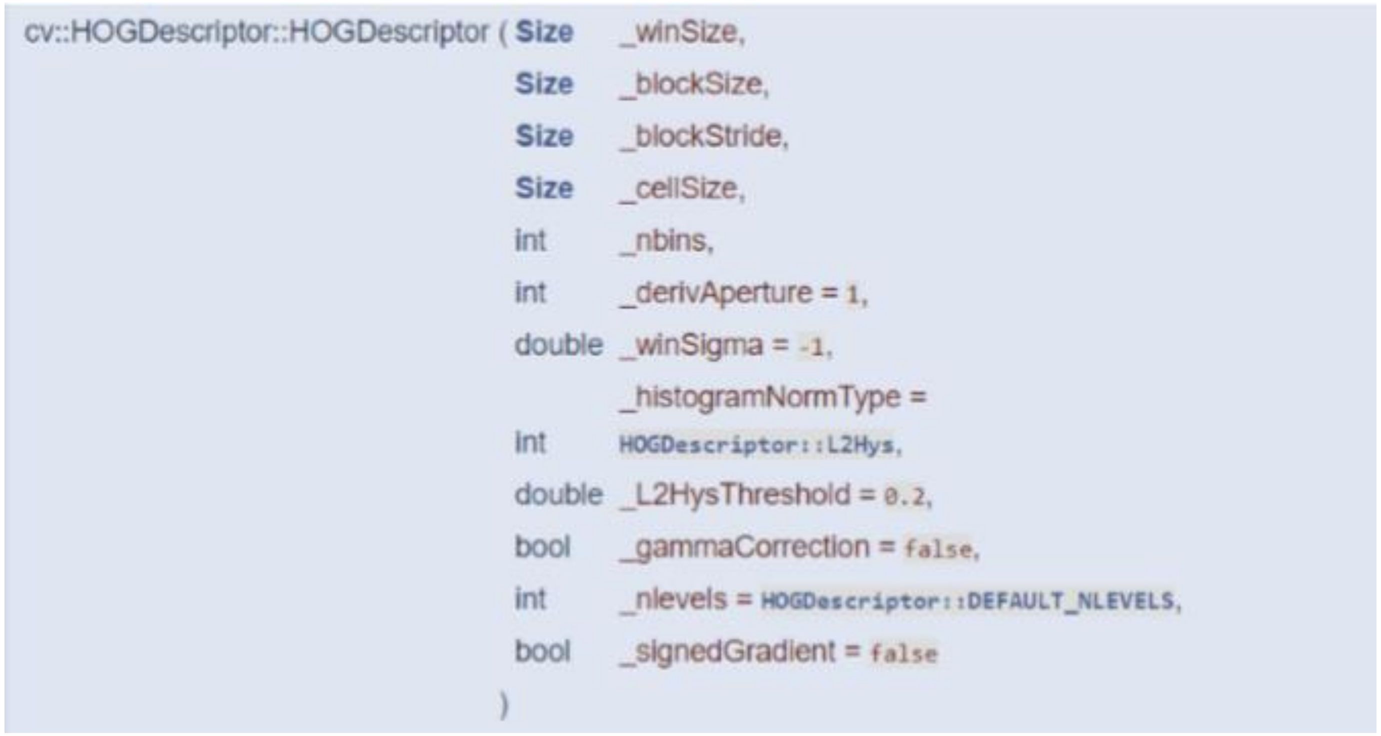
```
HOGDescriptor hog(Size(48, 96), Size(16, 16), Size(8, 8), Size(8, 8), 9);

hog.setSVMDetector(HOGDescriptor::getDaimlerPeopleDetector());
```

  - getDaimlerPeopleDetector()
  - getDefaultPeopleDetector()

Pedestrian Detection

- openCV function



```
cv::HOGDescriptor::HOGDescriptor ( Size    _winSize,
                                   Size    _blockSize,
                                   Size    _blockStride,
                                   Size    _cellSize,
                                   int     _nbins,
                                   int     _derivAperture = 1,
                                   double  _winSigma = -1,
                                           _histogramNormType =
                                   int     HOGDescriptor::L2Hys,
                                   double  _L2HysThreshold = 0.2,
                                   bool    _gammaCorrection = false,
                                   int     _nlevels = HOGDescriptor::DEFAULT_NLEVELS,
                                   bool    _signedGradient = false
                                 )
```

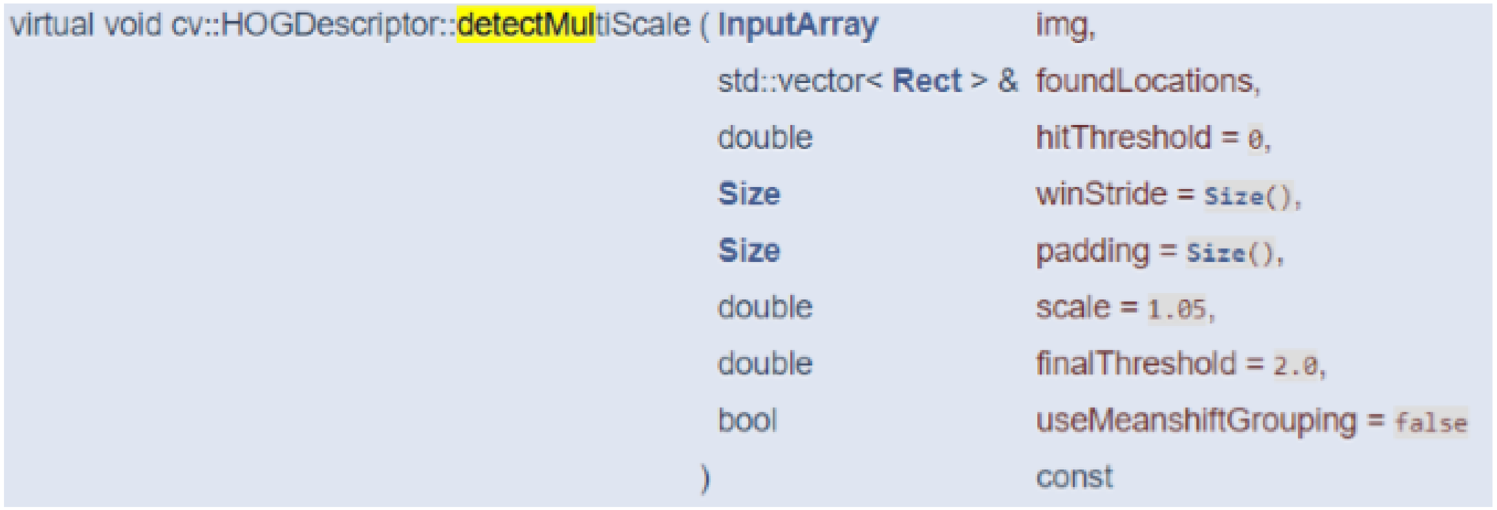
Pedestrian Detection

- openCV function
  - **win_size** – Detection window size. Align to block size and block stride.
  - **block_size** – Block size in pixels. Align to cell size. Only (16,16) is supported for now.
  - **block_stride** – Block stride. It must be a multiple of cell size.
  - **cell_size** – Cell size. Only (8, 8) is supported for now.
  - **nbins** – Number of bins. Only 9 bins per cell are supported for now.
  - **win_sigma** – Gaussian smoothing window parameter.
  - **threshold_L2hys** – L2-Hys normalization method shrinkage.
  - **gamma_correction** – Flag to specify whether the gamma correction preprocessing is required or not.
  - **nlevels** – Maximum number of detection window increases.

Pedestrian Detection

- openCV function

```
hog.detectMultiScale(frame, found, 1.2, Size(8, 8), Size(32, 32), 1.05, 6);
```



```
virtual void cv::HOGDescriptor::detectMultiScale ( InputArray            img,
                                                   std::vector< Rect > & foundLocations,
                                                   double                hitThreshold = 0,
                                                   Size                  winStride = Size(),
                                                   Size                  padding = Size(),
                                                   double                scale = 1.05,
                                                   double                finalThreshold = 2.0,
                                                   bool                  useMeanshiftGrouping = false
                                                 )                       const
```

Pedestrian Detection

- openCV function
  - **img** – Source image.
  - **found_locations** – Detected objects boundaries.
  - **hit_threshold** – Threshold for the distance between features and SVM classifying plane.
  - **win_stride** – Window stride. It indicates the "step size" in both the x and y location of the window
  - **padding** – It indicates the number of pixels in both the x and y direction in which the sliding window ROI is "padded" prior to HoG feature extraction. (8,8), (16,16), (24,24), (32,32)
  - **scale0** – Coefficient of the detection window increase.
  - **group_threshold** – Coefficient to regulate the similarity threshold. When detected, some objects can be covered by many rectangles. 0 means not to perform grouping.

Pedestrian Detection

- Example code

```
Mat frame;
vector<Rect> found;
int i;
char ch;

// open the video file
VideoCapture cap("pedestrian.avi");

if (!cap.isOpened()) {
		cout << "can't open video file" << endl;
		return 0;
}

// detector (48x96 template)
HOGDescriptor hog(
		Size(48, 96),
		Size(16, 16),
		Size(8, 8),
		Size(8, 8),
		9);

hog.setSVMDetector(HOGDescriptor::getDaimlerPeopleDetector());
```

Pedestrian Detection

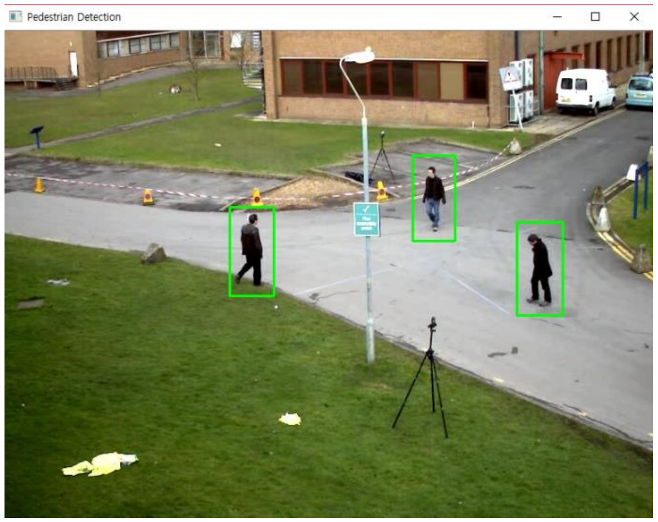
- Example code

```
while (1) {
        // input image
        cap >> frame;
        if (frame.empty()) break;

        // detect
        hog.detectMultiScale(
                frame,
                found,
                1.2,
                Size(8, 8),
                Size(32, 32),
                1.05,
                6);

        // draw results (bounding boxes)
        for (i = 0; i < (int)found.size(); i++)
                rectangle(frame, found[i], Scalar(0, 255, 0), 2);

        // display
        imshow("Pedestrian Detection", frame);
        ch = waitKey(10);
        if (ch == 27) break;                    // ESC Key
        else if (ch == 32)                      // SPACE Key
        {
                while ((ch = waitKey(10)) != 32 && ch != 27);
                if (ch == 27) break;
        }
}
```


Pedestrian Detection