

※ Computer Vision Scope

Artificial Intelligence : Algorithm을 하나하나 짜는 것

↓ (차이점 : Rule Base)

Machine Learning : 일정부분 학습 Data를 제공

↓

Deep Learning : Neural Networks

▶ Computer Vision 과 Image Processing이 일부 영역을 차지

※ Segmentation : 경계를 잘 오려 내는 것

- Semantic Segmentation : Image의 Class별로 Pixel를 Classification 하는 것 (Supervised)
 - Image Segmentation : Cluster로 구분 하는 것으로 Unsupervised Segmentation. Feature Vector 유사한 것으로 Clustering
 - Instance Segmentation : 연속된 물체가 겹칠 때 분리하기 위해 사용. Object Detection과 밀접
 - Panoptic[최근 각광 받음] : Instance 와 Segmentation의 중간 형태.. Image 내에서 Size등 처리함
- 참고 - Self Supervised : 한 프레임에 Class(Color로 인지)를 주면 다른 연속된 Image에 자동으로 생성
- Mask r CNN : 물체별로 바운드를 쳐서 물체로 뽑아냄

※ Obstacles of CV (컴퓨터비전의 장애)

- Viewpoint Variation
- Scale
- Illumination
- Occlusion (맞물림, 폐쇄)
- Deformation
- Background Clutter
- Different Resolution
- Intra-Class Variation

※ Applications of CV

- OCR (Optical Character Reader : 광학적 문자 판독장치)
- Biometrics : Fingerprinting Recognition
 - : Face Recognition
 - : Iris Recognition
- Self-Driving Car
- Advanced Driver-Assistance Systems
- Unattended Convenience Store
- Real-Time Facial Recognition
- Annotation(주석) Tool
- Diagnosis(진단) / Prediction
- Artistic Images
- Fake Information

※ Image Filtering

- What is filtering? CNN과 동일 과정으로 생각. 위치에 대한 Function Output. X, y 입력에 대한 Function Output
Local neighborhood(kernel)
- Use Filtering : Denoise, Resize, Texture, Edges, Template Matching등

※ Moving Average Filter

- Matrix에 Filter 크기로 평균값을 주어 Moving Average를 구함
- 참고 : 공식 및 Matrix 참고

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



	0	10	20	30	30	30	20	10	
	0	20	40	60	60	0	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	3	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

※ Shift Filter

- 기준점(중간)
- 오른쪽 값을 왼쪽으로 준다 : Left Shift

0	0	0
0	0	1
0	0	0

Shifted Left by 1 Pixel

※ Edge 도드라지게

0	0	0
0	2	0
0	0	0

- 1/9

1	1	1
1	1	1
1	1	1

= ?

0	0	0
0	1	0
0	0	0

+

0	0	0
0	1	0
0	0	0

- 1/9

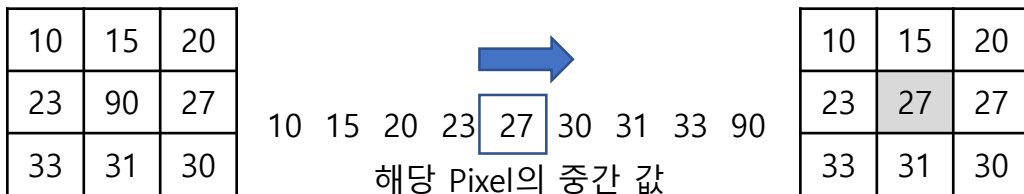
1	1	1
1	1	1
1	1	1

Original Image Filter

Blurring된 Image를 뺀

※ Median Filter

- Non Linear 연산
- Noise가 제거 됨



※ Gaussian Filter

- Gaussian Filter는 Blurring 효과
- 가중평균을 구한다. 평균은 Center Pixel, BandWidth만 조절(분산)

※ Gaussian Pyramid

- Image를 Size를 줄여 작은 Size을 얻어 피라미드로 쌓는 것
- Blurring(Gaussian) + Resize(Sub Sampling)

※ SIFT (Scale Invariant Feature Transform)

- Image에서 Feature Vector를 뽑아 줌. Key Point 를 추출 (ex. 책의 경우 각 꼭지점)
- 유사한 Feature를 Pair로 묶어서 Matching

1) Making Scale Space – Gaussian Pyramid Resizing + Bandwidth

2) Computing DoG Image : BandWidth별 Gaussian 함수를 Merge

DoG : Difference of Gaussians – Edge Detector로의 역할

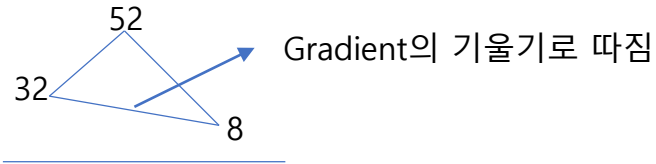
3) Finding Key Point : 가운데 Point가 Max나 Min을 찾음

- Gaussian Pyramid Resizing 후 DoG Output의 Center로 Key Point를 찾음

4) Removing Useless Keypoints

- Key Point 좌표를 Original Image에 Matching
- x, y 축 Gradient가 크면 꼭지점 임

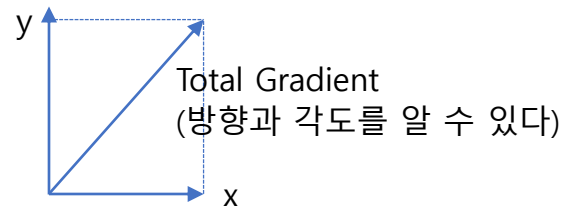
	65	
30	52	8
	10	



$$X\text{축은 } (8-30)/2 = -22/2 \text{ Gradient}$$

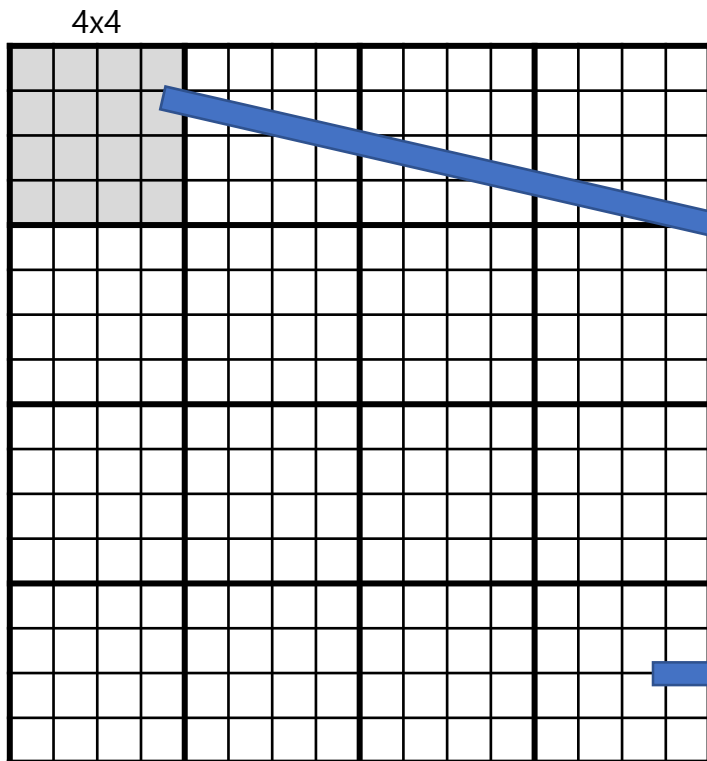
$$Y\text{축은 } (65-10)/2 = 55/2 \text{ Gradient}$$

- 5) Key Point Orientations : Gradient의 기준이 되는 방향
- 방향을 Align 해주면 회전에 Variation 한다
 - Magnitude와 Angle을 알 수 있다 (Magnitude : 크기, Angle : 각도)
 - Gradient는 바로 옆 Pixel로만 계산



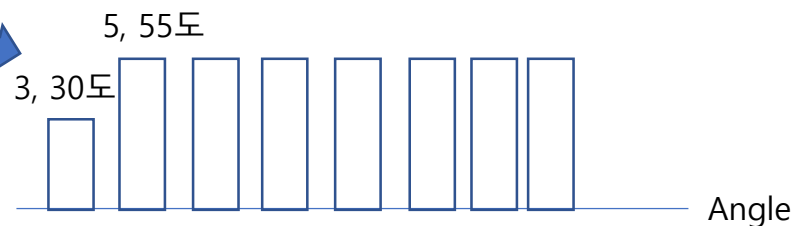
6) Generating Features

- 4x4 window



8Bins : 360도를 각 45도씩 8개로 구분

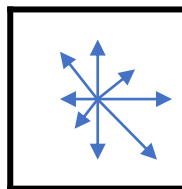
- Magnitude와 Angle로 Bin pareto로 표현 할 수 있음



4x4 Window를 8개의 Bin으로 표현

16x8 = 128개 Dimension Vector

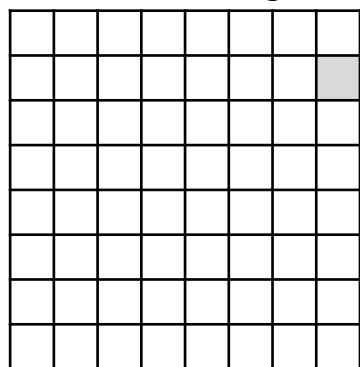
→ 한 개의 Key Point가 128개의 Dimension이 된다



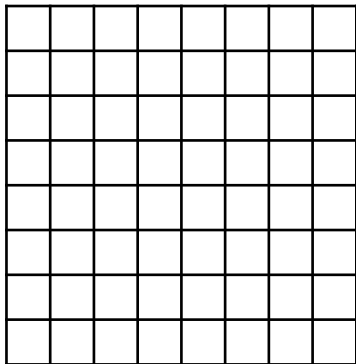
- Orientation Gradient를 빼주면 Align이 된다

※ HoG (Histogram of Oriented Gradients)

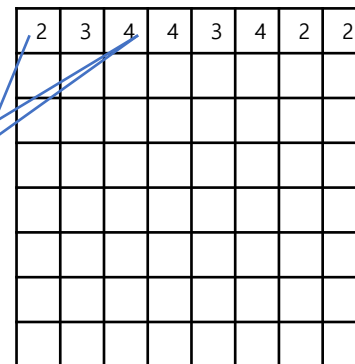
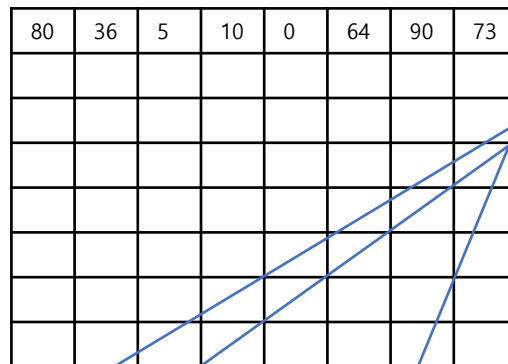
- 1) Calculate the Gradient Images : SIFT와 같이 Magnitude와 Angle을 구함
- 2) Calculate Histogram of Gradients in 8x8 Cells



8x8 Cells



X,Y축의 Gradient Magnitude와 Angle을 구함

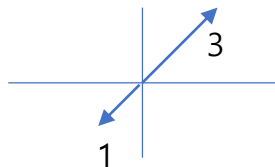


Gradient Magnitude

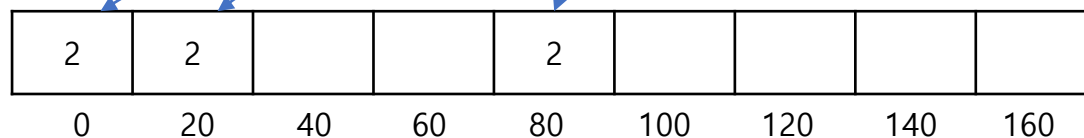
Gradient Direction(Angle)

9Bins (0, 20.....160)

: 벡터의 크기가 상쇄 되는 것을 반영



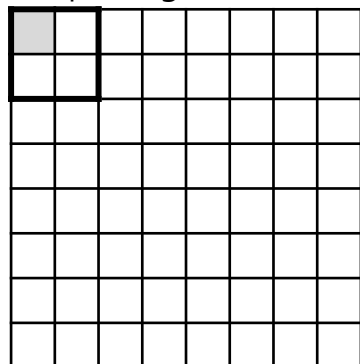
방향이 정반대이면 상쇄하여 표현



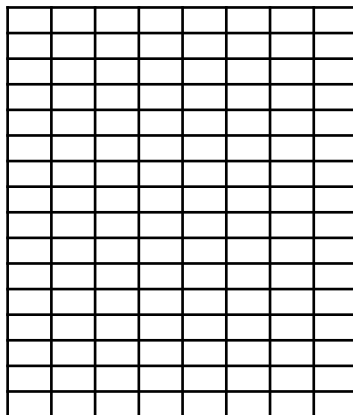
- 3) 16x16 Block Normalization

: Combined 4 Histogram

(2x2 pooling개념으로 생각하면 됨)



- 4) Calculate the HoG Feature Vector



- 36 Vectors (4Histogram * 9 Bins)
- (8x8)(8x16)로 구성 된 64x128을 2x2 width로 stride 1, padding 0이면 7x15인 105개 Block임
- Vector는 $36 \times 105 = 3780$