R-CNN details

Will Cover

1. Back ground
   - Computer vision : Selective Search , SIFT , Hog
2. R-CNN Architecture
   - 3 modules
3. How to test ? (detect , forward)
   - NMS
4. How to evaluate ?
   - mAP , several metrics + Yolo
5. How to train ?
   - different IOU thresholds (0.5 , 0.3 , ?)
   - Bbox regresson easy understanding
6. Limits of R-CNN

RNN 하자 할 것 ~

1. Background

배경부터 대충 흐름을 알아야 뒤가 납득이 되더라구요..

"Image dection" 이란 분야는 나중에 생긴게 아니라 원래 있었고 (classification)

Computer vision skills + ML 로 연구가 진행되고 있었음.

- Computer vision skills → Selective, Exhaustive search, SIFT, HOG, (SIFT, HOG: low-level feature extractor (ex. edge))
- ML → SVM (for classification), Linear regressor (for Bbox regression)

근데 AlexNet 을 기점으로

ML <<< DL 이라는 걸 classification 분야에서 보여줌.

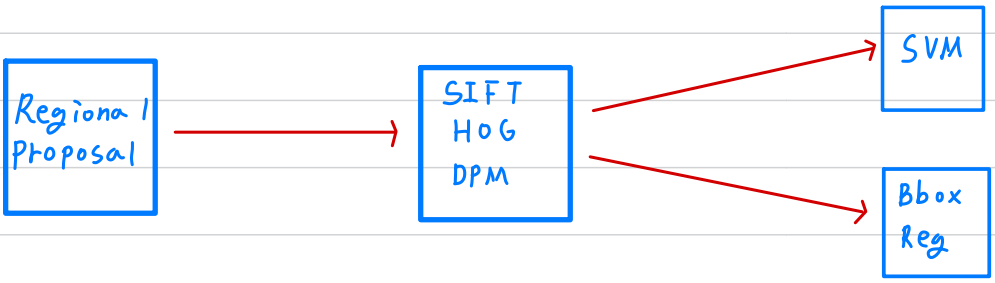
⇓

object-detection 연구원들 : 어 그러면 우리 하던거 고대로 쓰면서 이미지 특징 추출하는거만 SIFT, HOG, DPM 이런거 말고 CNN 쓸까 ??

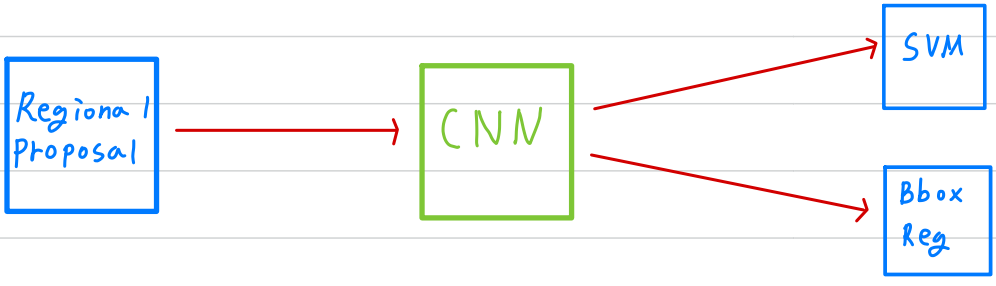
⇓

R-CNN 탄생

〈기존〉



〈변경〉



여기서 SVM, Bbox regressor 는 뒤에서 설명.

SIFT, HOG, DPM 등 low-level feature extractor 는 수식적으로 어렵고 DL 보다는 CV 영역에 해당하기 때문에 픽셀간 gradient 등을 통해 edge 같은 특징을 검출 하는 검출기 정도로만 이해.

Selective search 도 CV 영역이긴한데 납득이 안되는 부분만 간단히 이해하자면,

1. Segmentation 기법으로 유사 픽셀 묶음 화
2. 유사도 높은 인접 그룹끼리 병합

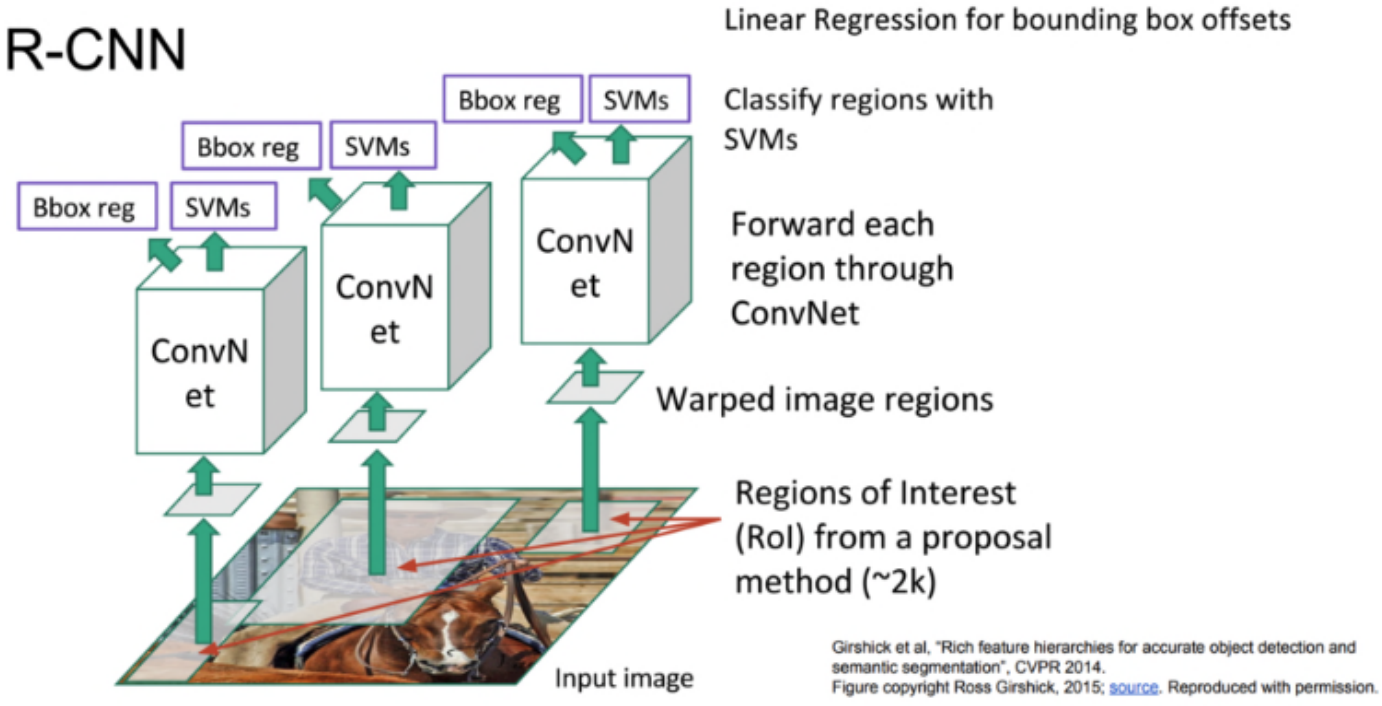
↓

유사도는 어떤 기준? 인접의 기준은?

https://wiserloner.tistory.com/1174

2. R-CNN Architecture

(저번 시간에 어느 정도 봤으니 리마인드만)



- 그림의 Conv. Net , Bbox reg 는 다 같은 모델이다.
  각 후보 영역에 대해 1번씩 적용된다는걸 표현한 것.

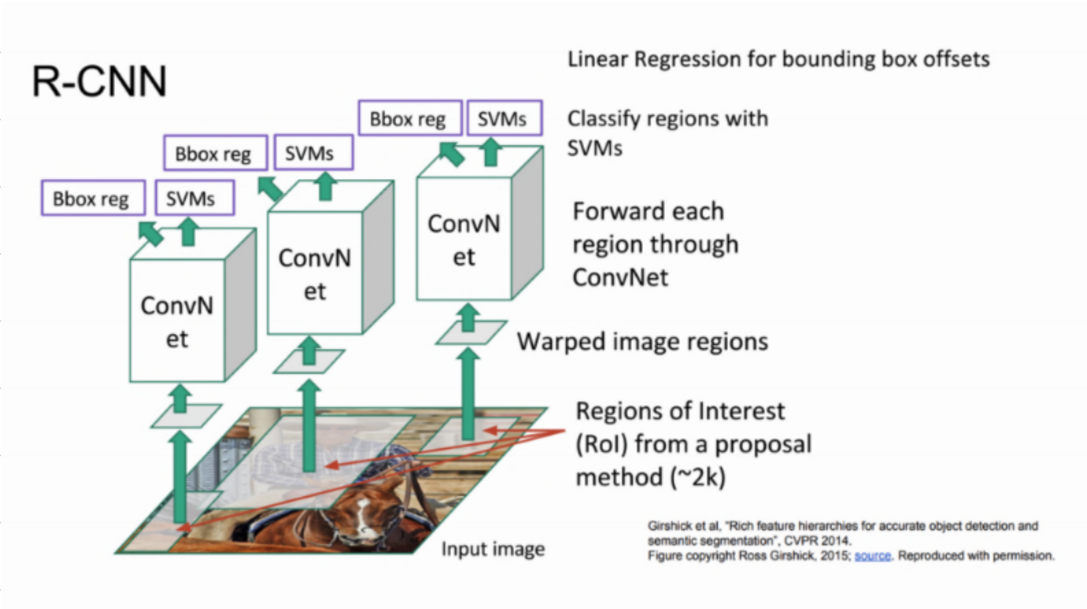
- 그림의 SVM 은 실제로 k+1 개 묶음 이다.
  K (검출 종류) + 1 (배경, 객체 없음)
  Y/N 로 만 출력하는 SVM 단일 분류기 k+1 개로 k+1 multi classification 하는것.

  P.S SVM 출력은 확률이 아니다. 0 or 1
  (뒤에 나올내용)

- Bbox 는 실제로 4개의 회귀식으로 이루어진 그룹이다.
  x, y, w, h 각각 담당.

3. How to test ?

(저번 시간에 어느 정도 봤으니 리마인드만)



★ 어떻게 저 값이 나와? 는 How to train 에서 다루니 "왜인지는 모르겠와 그걸 출력하게 끔 잘 학습이 되었나보다" 라는 가정을 하고 보자.

Image — R.P.(①) — CNN(②) { SVM(③), Bbox reg(④) } — result(⑤)

① 입력 : 이미지

출력 : S.S. 를 통한 후보 영역 (~2000) Warped.

② 입력 : ① 의 결과 뭉탱이에서 한 장씩

출력 : 잘 추출된 특징을 flatten 한 특성 벡터 (4096,)

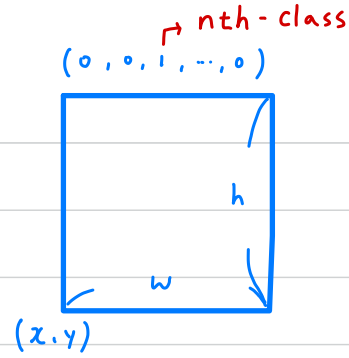
③ 입력 : ② 의 결과 벡터.

출력 : 하나만 1 나머지는 0인 K+1 사이즈 벡터 (0, 0, 1, 0, ⋯, 0)

④ 입력 : ② 의 결과 벡터

출력 : 정답 Bbox 의 절대좌표가 아니라 현재 ② 의 입력으로 들어온 이미지의 Bbox 가 x, y, w, h 일때 각 변수가 얼마나 변해야 정답이 될 수 있는지 그 '변화량' 을 x, y, w, h 로 표현한 값

⑤ 입력 :



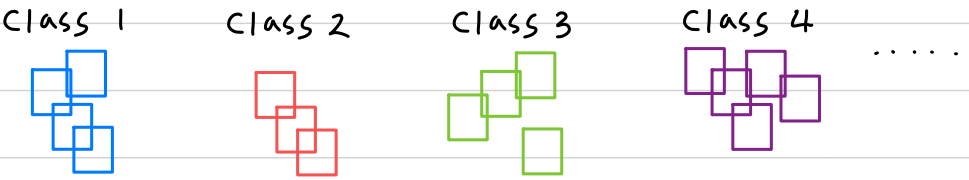
요런 꼴의 박스들 (좌표 및 분류 예측 결과)

출력 (최종 출력) : 입력에서 '배경' 이라는 레이블로 예측된 박스들은 버리고 나머지 박스들을 NMS 로 중복 제거 하여 선정된 자신 있는 박스들만 출력.

★ NMS (Non-Maximum Suppresion) ?

⑤ 에서 '배경' 레이블을 제외 하고 남은 박스들을 "클래스" 별로 묶는다.

ex)



각 클래스 별로 다음을 수행

- Confidence Score (확률값) 대로 내림차순
- 가장 스코어 높은 박스 기준으로 그 박스와 IoU > 0.5 이상인 박스들 제거.
- 그 다음으로 스코어 높은 박스 기준으로 〃
- 제거되지 않았거나 기준이 되지 않은 (조사 안된) 박스가 없을 때까지 반복

SVM 은 Softmax 와 다르게 확률이 아니라 다 1 일 것이다.
R-CNN 에서는 기준 박스가 랜덤하게 선택되지 않을까 예상.
NMS가 detection 분야의 범용 용어이니 알고 있자.
이후 모델들은 Softmax 씀.

4. How to evaluate ?

R-CNN 아키텍쳐로 부터 **이미지 입력 ⟶ 선별된 박스 출력** 까지 과정까지 살펴봤는데, 그럼 이 출력의 **맞다 / 틀리다** 는 무슨 기준으로 정하지 ?

[스코어 정의]

분류가 맞았어도 박스가 너무 벗어났거나, 박스는 근접해도 분류가 틀려버리면 오답처리를 해야한다.

↳ 두 정보를 모두 고려할 지표를 만들자.

Score = Confidence score x IOU

- Score: 용어가 조금 오피셜을 찾기 힘듬. 저 곱한값 자체도 Confidence score 라고도 부르는것 같음.
- Confidence score: 헷갈릴수 있는데 박스의 확률 중 '정답' 레이블일 확률
  - 오분류 했다면 Softmax 기준으로는 아주 작은 값, SVM 기준으로는 0 일것.
- IOU: 박스가 많이 벗어나면 0에 가까울것.

[임계치 설정]

예를들어 Conf = 0.8, IOU = 0.6 으로 Score = 0.48 를 얻었다면 몇 이상부터 **맞았다** 라고 해야되지 ?

↳ Average Precision !!!!

이진분류 문제를 잠시 생각해보자.

0, 1 중 하나로 정해야 하는 모델의 출력이 0.7 이면 1로 분류해야 되는 걸까 ?

↳ 그건 **임계치** 에 따라 다르다.

0.5를 기준으로 나누기로 했으면 1로 분류하는 모델이 되겠지만, 이 **임계치** 라는 건 Task 에 따라 천차만별이다.

암 양성/악성 문제라면 1로 틀리는 것과 0으로 틀리는 것이 무게가 다른 오류이기 때문에 임계치가 아주 높아 (ex. 0.95) 0.7 도 0으로 분류하는 모델이 될 수도 있다.

내부 계산 과정은 동일한데 임계치에 따라 최종출력이 달라질 수 있는 것이다!!

그만큼 임계치는 중요하고 Task 성격에 맞아야한다!!

일반적으로

임계치가 높으면
- 모델은 가장 확실할때만 1로 분류할 것이기 때문에
- 모델이 출력한 1은 정답일 확률이 높다. (정밀도 ↑)
- 그러나 1을 많이는 출력 안할테니
- 정답인 1을 놓치는게 많을것이다. (재현율 ↓)

임계치가 낮으면
- 모델은 조금만 높아도 1로 분류할 것이기 때문에
- 모델이 출력한 1은 정답일 확률이 낮다. (정밀도 ↓)
- 그러나 1을 많이 출력할테니
- 정답인 1을 놓치는게 적을것이다. (재현율 ↑)

물론 **모델성능이 좋을수록** 저 반비례 관계인 지표는 차이가 미미할 것이고
모델성능이 나쁠수록 저 반비례 관계인 지표는 차이가 엄청날 것이다.

자 그럼 다시 본론으로, **객체 검출** 이란 Task는 몇 임계치가 적당할까?

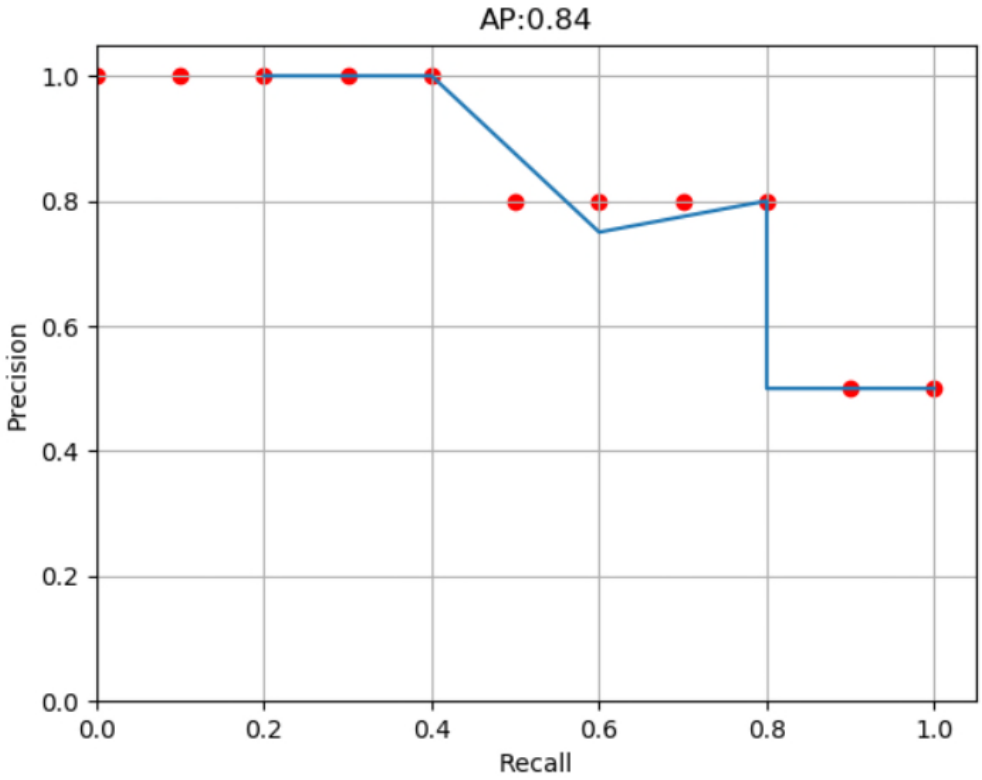
↳ **모른다!! 완벽한 임계치를 정하기 애매하다!!**

이때 똑똑한 어느 연구원의 머리속

"근데 좋은 모델일수록 임계치 변화에 **정밀/재현** 차이가 크지 않으니까
다양한 임계치를 적용했을때 **정밀/재현** 값을 종합적으로 고려해보는게 어때??"
(왜 그랬어..)

PR 곡선 : 임계치를 변화시켜가며 precision/recall 값을 기록한 그래프.

이거 실제로는 임계, 정밀, 재현 의 관계라서
3차원 그래프인데 앞두개 다 짜르고 2차원으로
찌부시켜놔서 이해 너무 힘들었음...



이런 식으로 점을 기록하여 그래프를 그렸을때 아래 면적을 AP 라고 정의했다 ~~~~

실제로는 보정을 조금해서 계산이 쉽도록 직각 형태로 만들어서 면적 계산.

그럼 mAP 는 ?? 각 클래스에 대해서 수 요짓거리를 통해 모델별 AP가 계산되면 평균 (mean) 값 취한 값 !!

Object Detection 분야에서는 이 mAP 를 성능지표로 씁니다.

+ TMI

- % 로 표현해서 뭔가 accuracy 처럼 좋은 모델은 mAP 97% 이렇게 나올것 같지만 실제로 mAP 는 가늠치 같은거 없는 복잡한 지표. (Yolo v3 도 Pascal VOC 50~60 % mAP)
- '다양한' 임계치를 통해 라는 말에서 저 '다양한' 은 대회 주최측이 정해서 계산이 천차만별 (Pascal metrics, COCO metrics, ... 등)
- 수 이것 때문에 Pascal 측에서 계산구간을 한번 변경한 적이 있는데 그 계산방식이 너무 엉터리라서 Yolo 개발자가 욕 엄청하고 대회 참가 명축고 V3 에서 개발 손절 했다고 함.

5. How to train ?

ㅠㅠㅠ 누가 이거 한주치 분량으로 계획했어 ...

사전 용어 하나

[end-to-end training]

↳ 입력이 들어와 출력이 나가기까지 중간에 있는 과정 모두가 **하나의** 손실함수를 통해 학습할 수 있는 구조 , 학습하는 것.

R-CNN 이 딥따 어려운 이유 .. end-to-end 구조가 아니다 ...

CNN 모듈 , SVM 모듈 , Bbox 모듈 다 따로 학습시켜 줘야 한다.. 지저스 .

심지어 순서도 있다. CNN 먼저 학습해야 SVM, Bbox reg 학습가능.

└ 이 둘은 병렬 학습 가능.

하나씩 살펴보자.

1. CNN module

정리글에는 '이미지넷 데이터로 pre-train 한다 → Pascal VOC 로 fine-tuning 한다' 라고 되어 있는데 이건 처음부터 구현할 때 얘기고 실제로 버클리 연구팀은 pre-train을 하진 않았고 이미지넷 우승한 AlexNet 공개되어 있는 가중치 그대로 가져왔다고 함.

가져 온 Alexnet 으로부터 이제 fine-tuning을 해야하는데 순서가 다음과 같음.

1. classifier 바꾸기
2. 학습데이터 정의하기
3. 기존 CNN 처럼 학습하기.

1. classifier 바꾸기

기존 AlexNet 은 이미지넷에 맞게 최상층이 1000 사이즈 Dense 층.

이걸 (k+1) 사이즈 Dense 로 갈아 낀다.

2. 학습데이터 정의하기.

Regional Proposal 이 test 에만 진행되는 것이 아니라 train 에서도 진행됨.

2000 개의 후보 중에서 IOU > 0.5 를 기준으로 True (객체이다), False (아니다) 로 표기.

T/F 비율 차이가 심할테니 순서 조정 + 오버/다운 샘플링 을 통해서 32개 T, 96개 F 마다 가중치 갱신이 이루어 질 수 있도록 mini-batch = 128 세팅

★ False 데이터 안쓰는거 아님!! 2000 장 다씀

3 기존 CNN 처럼 학습하기

★ fine-tuning 시에는 Dense - softmax 로 학습하지만 Conv net 학습이 끝나고 SVM, Bbox 학습시에는 Dense 층 제거하고 최종 feature map 을 flatten 한 특성 벡터를 입력으로 전달

2. SVM module

+ 왜 SVM을 썼는가 시간 남으면 전달

SVM 은 학습이 끝난 CNN의 출력을 입력으로 받아 K+1 개의 이진 분류기가 학습을 진행.

알고 넘어가야 할 것 : 학습데이터.

SVM 이 학습하는데 쓰이는 데이터는 CNN 때와 다름 (Bbox 는 또 다를 극혐)

CNN 이 2000 개 중 Ground Truth 와 IOU > 0.5 인 "후보영역" 을 True 데이터로 썼다면, SVM 은 "Ground Truth" 만 True 데이터로 썼다. (후보에서 안씀)

False 데이터는 Ground Truth 와 IOU < 0.3 인 후보 데이터를 씀.

이처럼 데이터 분류 비율을 조정하기 위해 "확실한 False" 만 골라서 쓰는 기법을 Hard-negative mining 이라고 함.

3. Bbox regressor

이 모듈은 "회귀" 모델이기 때문에 학습데이터에 레이블이 없음. IOU > 0.6 인 후보 이미지를 통해 학습하는데, "이미지 특성 ⟶ 좌표" 라는 회귀가 어떻게 가능한지 직감적인 이해만 해보자.

(x,y) = center, (w,h) = width, height
Proposal: $P=(P_x, P_y, P_w, P_h)$
Ground truth: $G=(G_x, G_y, G_w, G_h)$

현재 학습에 쓰이고 있는 후보 이미지의 좌표 정보를 P_x, P_y, P_w, P_h 라 하고 정답 이미지의 좌표정보를 G_x, G_y, G_w, G_h 라고 하면

P_x, P_y, P_w, P_h 의 정보를 모르는데 이미지 특성 (feature vector)을 입력으로 하여 G_x, G_y, G_w, G_h 로 예측 한다는 거 상식적으로 말이 안된다.

핵심은 G_x, G_y, G_w, G_h 를 예측하는게 아니라 다음을 예측하는 것이다.

$$t_x = (G_x - P_x)/P_w \quad (6)$$
$$t_y = (G_y - P_y)/P_h \quad (7)$$
$$t_w = \log(G_w/P_w) \quad (8)$$
$$t_h = \log(G_h/P_h). \quad (9)$$

회귀의 목표 값을 t 로 설정하는 것.
P, G의 정보가 모두 담기고 "차이"를 P로 표현한 수치.

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2$$

$$d_\star(P) = \mathbf{w}_\star^{\mathrm{T}} \boldsymbol{\phi}_5(P)$$

Bbox regressor 를 d 라고 표현 했을때
d(feature vector) 의 결과가 t 와의 오차가 줄어들도록 하는것 !!!

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

d 를 통해 회귀 수치가 산출되면 다음의 식을 통해
P_x, P_y, P_w, P_h 를 조정하여 $\hat{G}$ 을 최종 좌표로 쓰는것 !!!

6. Limits

- 앞서 봤듯이 너무 … 복잡함. end-to-end 가 안돼서 …
- 오래 걸림 .. 한 장당 2000 장의 후보군을 그것도 순차적으로 학습/예측 해야함..
- 메모리가 엄청 필요. 3 모듈 각각 학습하는 동안 후보이미지 계속 어딘가에 저장되어 있어야함..
- Warping 으로 인한 후보들 정보손실

그럼에도 "첫 사례" 라는 중요한 의미를 갖는 모델인건 맞다.