

Semantic Segmentation with Active Semi-Supervised Learning

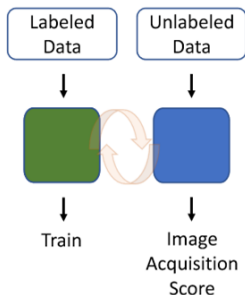
Aneesh Rangnekar, Christopher Kanan, Matthew Hoffman

- **Problem/Objective**
 - Desire to reduce labeling costs in semantic segmentation
- **Contribution/Key Idea**
 - Active learning
 - Semi-supervised learning

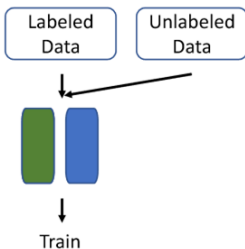
Semantic Segmentation with Active Semi-Supervised Learning

-Traditional active learning

1. Labeled data 사용하여 network를 train.
2. Learned network를 사용하여 unlabeled data pool 에 있는 샘플들 score. (scores for data acquisition).
3. Every active learning cycle C, a portion of samples from the unlabeled pool - "budget" B, is sampled, annotated and added to the labeled set.



-Traditional semi-supervised learning



-Active learning for semantic segmentation에 3가지 문제 존재.

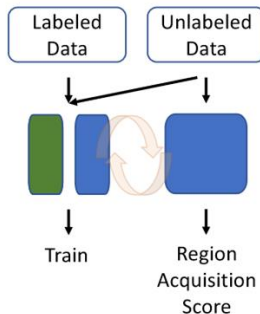
-이 문제를 완화하기 위해 Active learning에 Semi-supervised learning 도입함.

-Active learning + Semi-supervised learning

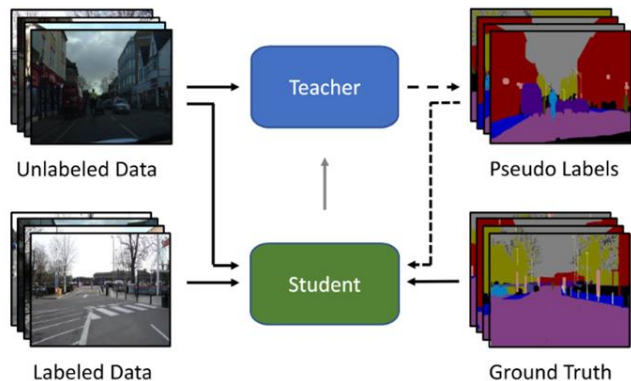
To mitigate these challenges, we leverage unlabeled data using semi-supervised learning during each active learning cycle C as a means of understanding the unlabeled data distribution. (Our approach used semi-supervised learning on the unlabeled data to generate region acquisition scores that can be queued for human annotations).

Teacher-student framework 사용하여 pixel마다 pseudo-label 만들.
(to learn better representations of the unlabeled data pool while assisting the network's learning with labeled data).

Entire images 샘플링 하는 대신 regions 샘플링.



-Teacher-student framework



Network 2개 사용.

-Student :

Cross-entropy로 train 됨.

(Unlabeled data의 cross-entropy : Teacher가 만든 unlabeled data의 pseudo labels와의 cross-entropy).

-Teacher :

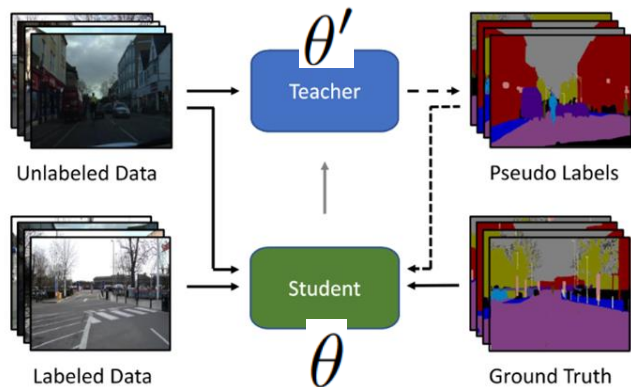
Student's parameters로 update 됨.

At the end of the SSL training for each active learning cycle, we rank the regions within the unlabeled pool of images based on an acquisition metric computed with the teacher's performance and then select the highest ranking set for labeling.

-A slow moving update (teacher network) results in more stable cumulative predictions, however, simply adopting this approach is difficult due to the overall class distribution bias in the labeled data.

이 문제를 해결하기 위해 2가지 정규화 방안 소개.
(Confidence Weighting & Balanced Classmix).

3.1. Teacher-Student Framework



Network 2개. (Student, Teacher).

Student 와 Teacher의 네트워크 구조 동일함.
Teacher가 Student's parameters의 EMA로 업데이트됨.

$$\theta' := m\theta' + (1 - m)\theta$$

(m: smoothness coefficient (momentum) : 0.99).

-Supervised loss for labeled images

Student가 'gt'와의 loss(cross-entropy)로 train 됨.

$$\mathcal{L}_{sup} = \ell_{ce}(\theta(x_l), y_l)$$

x_l : 네트워크에 들어가는 inputs.

y_l : corresponding gt labels.

-Unsupervised loss for unlabeled images

Student가 'Teacher가 만들어낸 pseudo labels'과의 loss(cross-entropy)로 train됨.

$$\mathcal{L}_{unsup} = \ell_{ce}(\theta(x_{u-s}), [\theta'(x_{u-w})])$$

'Teacher가 만들어낸 pseudo labels' : Teacher의 $x_{(u-w)}$ 에 대한 예측.

$x_{(u-w)}$: Teacher의 input으로 들어가는 weakly-augmented version of the images

$x_{(u-s)}$: Student의 input으로 들어가는 strong-augmented(perturbed) version of the images

[] : conversion of the logits to one hot vectors. 이 결과를 gt로 사용.

$[\theta'(x_{(u-w)})]$: one-hot vector indicating the corresponding pseudo label for the pixel.

-Final loss for training

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \eta \cdot \mathcal{L}_{unsup}$$

η : weighting for the unsupervised loss

: (이미지 내에 $p > 0.97$ 을 만족하는 픽셀 개수) / (이미지 내 총 픽셀 개수)

p : 해당 픽셀의 pseudo label 에 대한 최대 확률.

3.2. Confidence Weighting

-Final loss for training

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \eta \cdot \mathcal{L}_{unsup}$$

η : weighting for the unsupervised loss

: (이미지 내에 $p > 0.97$ 을 만족하는 픽셀 개수) / (이미지 내 총 픽셀 개수)

p : 해당 픽셀의 pseudo label 에 대한 최대 확률.

- η

: pseudo labeled pixels의 기여도

: mini-batch 세팅에서 image 단위로 적용됨.

: 이미지 내의 모든 픽셀들이 prediction confidence와 무관하게 동일한 중요도로 train 되는 문제 발생.

-문제발생 시나리오. (클래스 분포 편향 문제).

Student : 제한된 훈련 세트에 존재하는 클래스 분포에 빠르게 overfit 됨.

Teacher : Teacher 네트워크의 업데이트가 느리기 때문에, 픽셀 주석이 제한된 클래스에 편향되는 문제 발생.

Active learning 환경에서, 네트워크가 예측에 대해 너무 확신하게 되면 클래스 분포 편향 문제가 더 해로워짐. 이는 네트워크가 acquisition metric 을 계산할 때 잘못된 확신을 가지게 할 수 있기 때문.

-Unsupervised loss for unlabeled images (기존)

Student가 'Teacher가 만들어낸 pseudo labels' 과의 loss(cross-entropy)로 train됨.

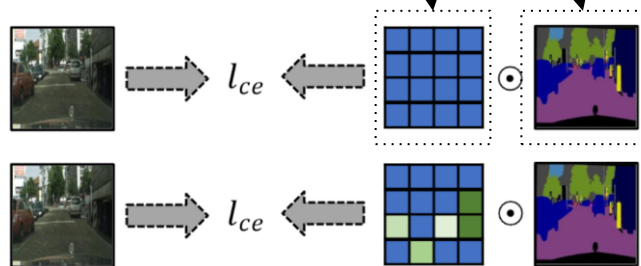
$$\mathcal{L}_{unsup} = \ell_{ce}(\theta(x_{u-s}), [\theta'(x_{u-w})])$$

-Unsupervised loss for unlabeled images (수정)

Student가 'Confidence weighted + Teacher가 만들어낸 pseudo labels' 과의 loss(cross-entropy)로 train됨.

$$\mathcal{L}_{unsup} = \ell_{ce}(\theta(x_{u-s}), p \cdot [\theta'(x_{u-w})])$$

p : Teacher로부터 나온 모든 pseudo labels 에 대한 해당되는 최대 확률



(파랑색 : high confidence, 초록색 : low confidence 를 나타냄.)

효과 : High confidence pseudo labels and annotated labels 가 Low confidence pseudo labels 보다 더 많이 중요도 가질 수 있게 해줌.

3.3. Balanced ClassMix

문제 : Tail classes imbalance

해결 : Data augmentation 해서 oversampling.

(즉, Unsupervised loss 구할 때 strong augmentation 방식 중 하나로 추가 함).

-Replay buffer

: 이전 샘플들과 NN을 train할 새로운 샘플들을 한데 모음.

: 계속 잊어버리는 거 완화해줌.

: 메모리 제약으로 replay buffer의 사이즈가 제한됨.

: 다양한 샘플링 방식이 있지만, 본 논문에서는 uniform sampling 함.

1. Replay buffer의 사이즈를 M으로 초기화함.

2. 매 iter마다, buffer에 이미지 채움.

3. 현재 iter 에서, buffer로부터 이미지를 random sampling 함.

///단, ClassMix data augmentation 을 위한 확률분포는 수정됨.

(ClassMix : 전체 클래스 집합에서 마스크를 uniform sampling.)

Our setting : Labeled data 내 클래스 분포에 기반하여, head classes 보다 tail classes 로부터 더 많은 샘플들을 얻기 위해 sampling rate을 bias 시킴.

2개의 분리된 분포 (head, tail) 로부터 ClassMix를 위한 클래스들을 샘플링하는 대신, 1개의 결합된 분포로부터 샘플링하는 방식을 취함.///

4. Sampling한 이미지들로 아래 unsupervised loss 구함.

$$\mathcal{L}_{unsup} = \ell_{ce}(\theta(x_{u-s}), p \cdot [\theta'(x_{u-w})])$$

즉, Unlabeled image들을 저장해 두는 replay buffer를 사용하고, tail class 의 sample들을 더 많이 확보하기 위해 이 buffer에서 sample들을 random sampling하여 ClassMix 기법을 적용함.

-Final loss for training (기존)

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \eta \cdot \mathcal{L}_{unsup}$$

η : weighting for the unsupervised loss

: (이미지 내에 $p > 0.97$ 을 만족하는 픽셀 개수) / (이미지 내 총 픽셀 개수)

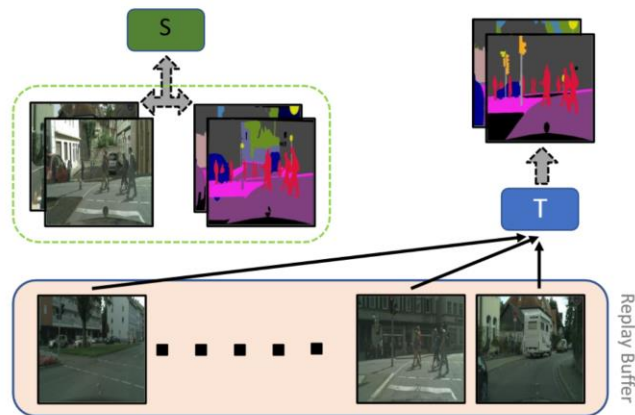
p : 해당 픽셀의 pseudo label 에 대한 최대 확률.

-Final loss for training (수정)

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \eta_1 \cdot \mathcal{L}_{unsup1} + \eta_2 \cdot \mathcal{L}_{unsup2}$$

Thus, for the current mini-batch, we get two different xus , where the first set corresponds to within mini-batch augmentations (Lunsup1), and the second set corresponds to replay augmentations (Lunsup2).

For computation efficiency, we only maintain the ClassMix-d versions equal to the batch size.



3.4. Sampling Strategy

-S4AL(Semi-supervised semantic segmentation for active learning)

-Acquisition metric 4개.

1. Random sampling
2. Least confidence
3. Softmax entropy. <- Best.
4. Softmax margin.

4.1. Datasets

Dataset : CityScapes, CamVid

Labeled set : Train set 에서 10%를 random sampling.

Unlabeled set : Train set 의 나머지 90%.

(위에서 Labeled set 으로 포함된 이미지 제외하고 전부).

Ignore index도 라벨링 대상에 포함시킴.

