Semi-supervised Learning of FixMatch and after FixMatch

Data **M**ining & **Q**uality **A**nalytics Lab.

2023년 2월 3일(금)

조용원

발표자 소개

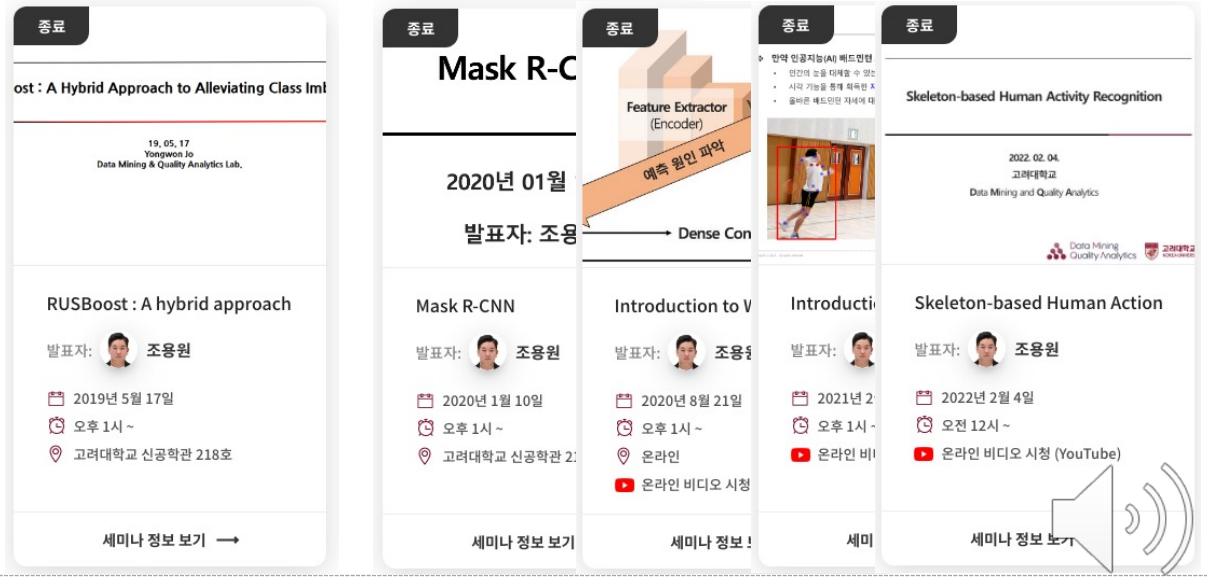
❖ **조용원(Yongwon Jo) – Ph.D. Candidate**

- 고려대학교 산업경영공학과 석•박통합과정 8학기 재학 중
- Data Mining & Quality Analytics Lab (김성범 교수님)

❖ **Research Interest**

- Semantic Segmentation and its applications
- Anomaly Detection and Segmentation
- Semi-supervised Learning for the Regression Problem

- RUSBoost
- Mask R-CNN
- Weakly-supervised Semantic Segmentation
- Human Pose Estimation
- Skeleton-based Human Activity Recognition

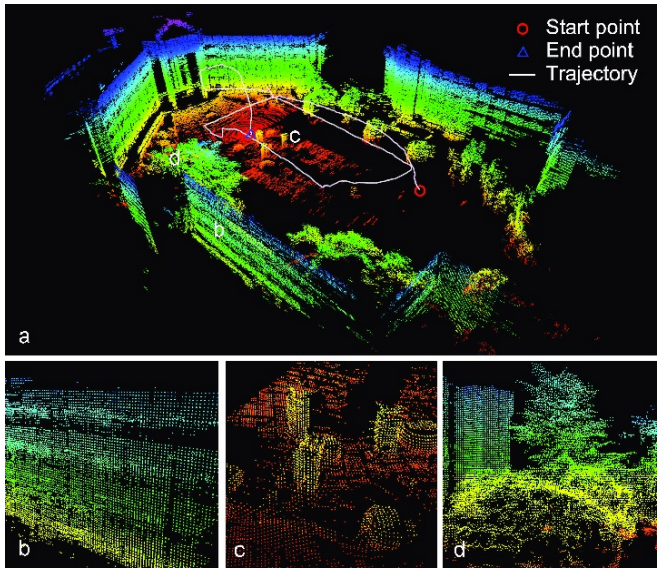


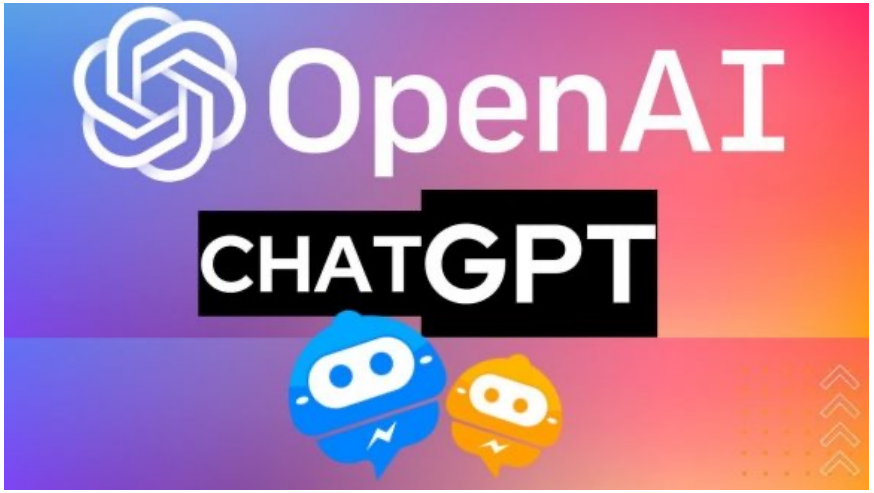
목차

1. Introduction

❖ 최신 Deep Learning 기반 인공지능 기술

- 로봇을 작동시켜 Point Cloud 형태로 지형을 파악할 수 있는 Deep Learning 모델(SLAM)
- ChatGPT와 같이 자연스러운 대화 가능한 Deep Learning 모델
- 위와 같은 모델 학습을 위해서는 많은 (입력-출력) 데이터 필요



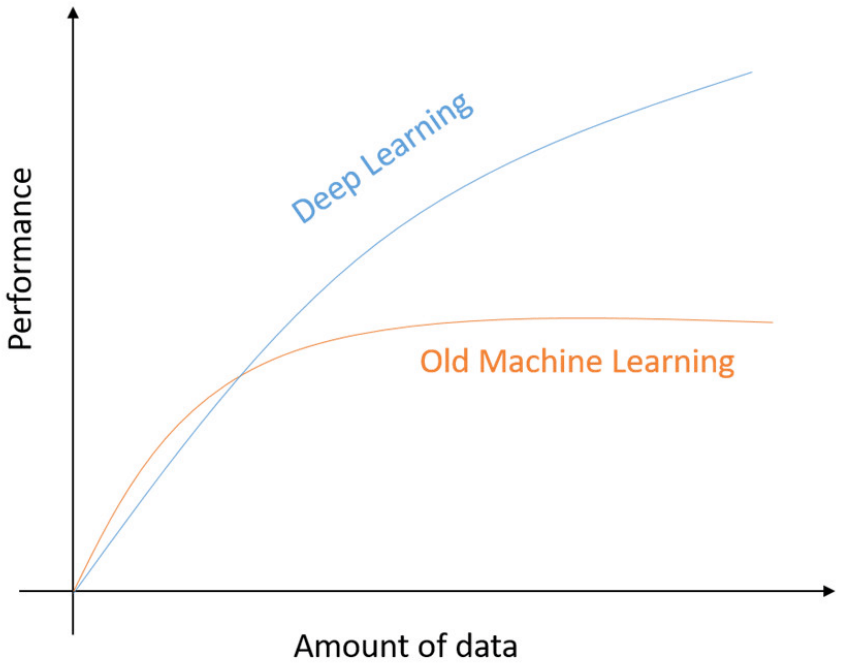


- https://www.researchgate.net/figure/Map-of-a-plaza-reconstructed-by-GP-We-show-those-points-whose-uncertainty-are-below_fig1_343415025
- https://blogs.chapman.edu/academics/2023/01/09/what-is-chatgpt-what-can-educators-do-about-cheating/

1. Introduction

❖ 최신 Deep Learning 기반 인공지능 기술

- Deep Learning 모델 성능은 관심 있는 현상을 설명할 수 있는 좋은 데이터가 많을수록 향상
- 즉, (입력-출력)으로 구성된 데이터(Labeled) 다수가 필요



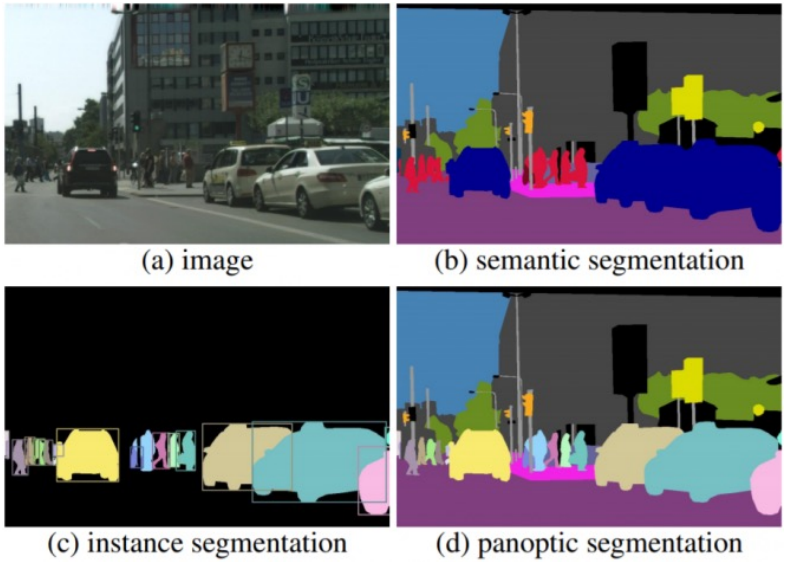
- https://www.researchgate.net/figure/The-performance-of-deep-learning-with-respect-to-the-amount-of-data_fig3_331540139

1. Introduction

❖ **Labeled 데이터 수집 시 발생하는 어려움**

- 일반적인 사람들이 쉽게 인식할 수 있는 범주를 픽셀별로 및 객체 경계 상자 표기 가능
- 하지만 표기해야 할 객체 수가 많아 특정 이미지 범주 표기 시 많은 인력 및 시간 필요

Image Segmentation 예시



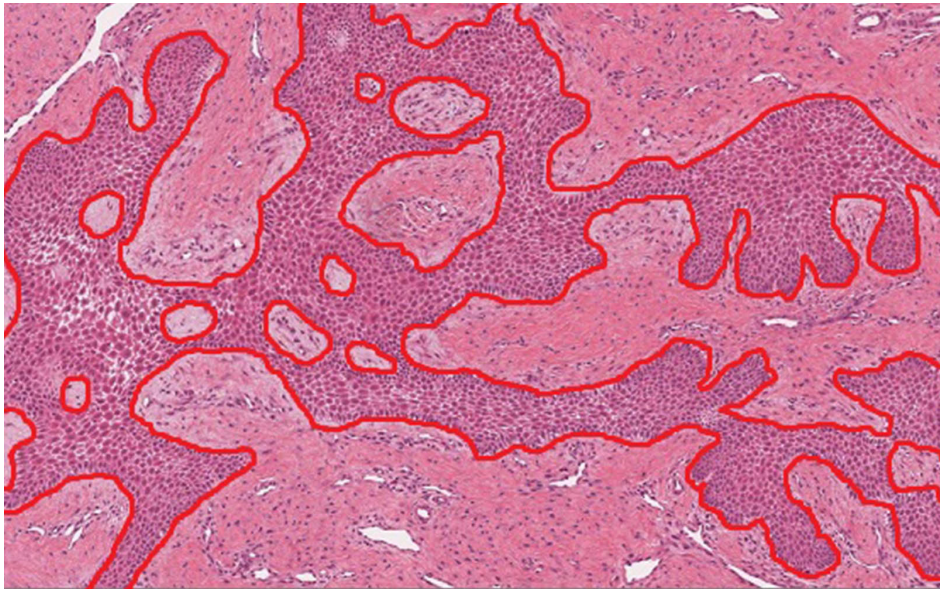
6D Object Detection 예시

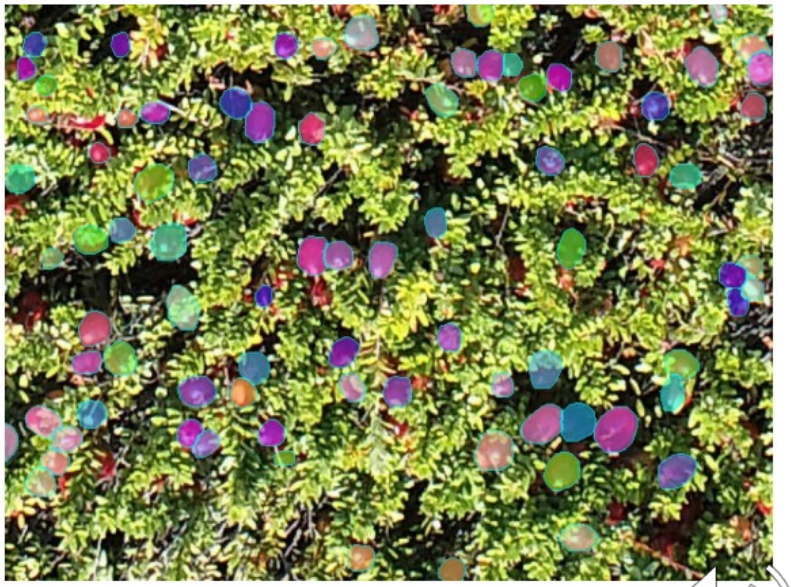
- https://engineer-mole.tistory.com/273

1. Introduction

❖ Labeled 데이터 수집 시 발생하는 어려움

- 의료 및 농업과 같이 특수한 분야에 대한 데이터 수집 시 전문가 지식 필요
- 즉, 전문가가 직접 데이터에 대한 레이블을 표기(Annotation) 해야 하는 상황
- 전문가 인원 수는 적고 인력 투입에 대한 비용은 크기에 데이터 수집이 어려움



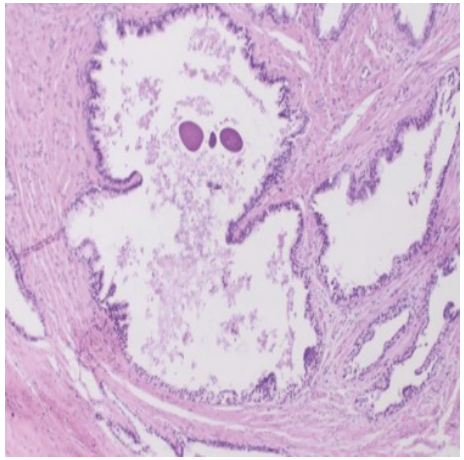
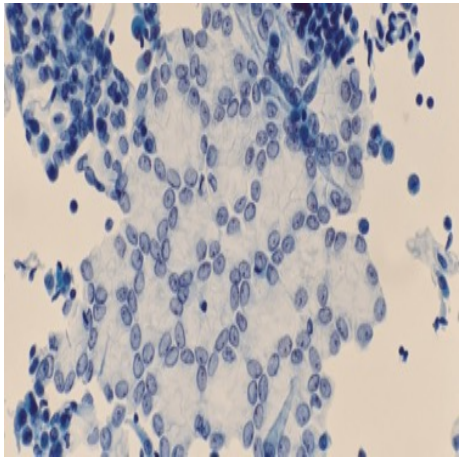
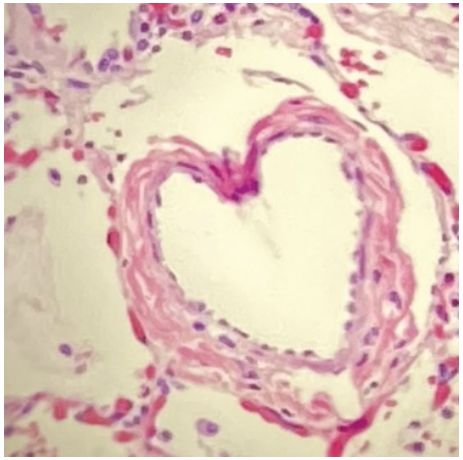
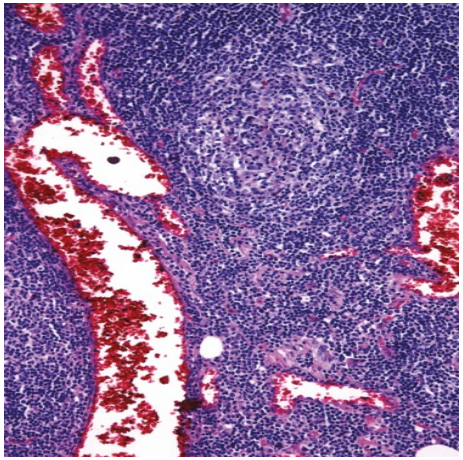


- https://digitalpathologyplace.com/10-best-pathology-resources-for-designing-image-analysis-algorithms/
- Akiva, P., Dana, K., Oudemans, P., & Mars, M. (2020). Finding berries: Segmentation and counting of cranberries using point supervision and shape priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 50-51).

1. Introduction

❖ **레이블링 되지 않은 데이터(Unlabeled)**

- 데이터를 사용해 설명하고자 하는 현상 자체를 촬영하는 것은 상대적으로 쉬움
- 최근 많은 Unlabeled 데이터와 적은 Labeled 데이터를 활용해 예측 모델을 학습하고자 하는 연구 등장

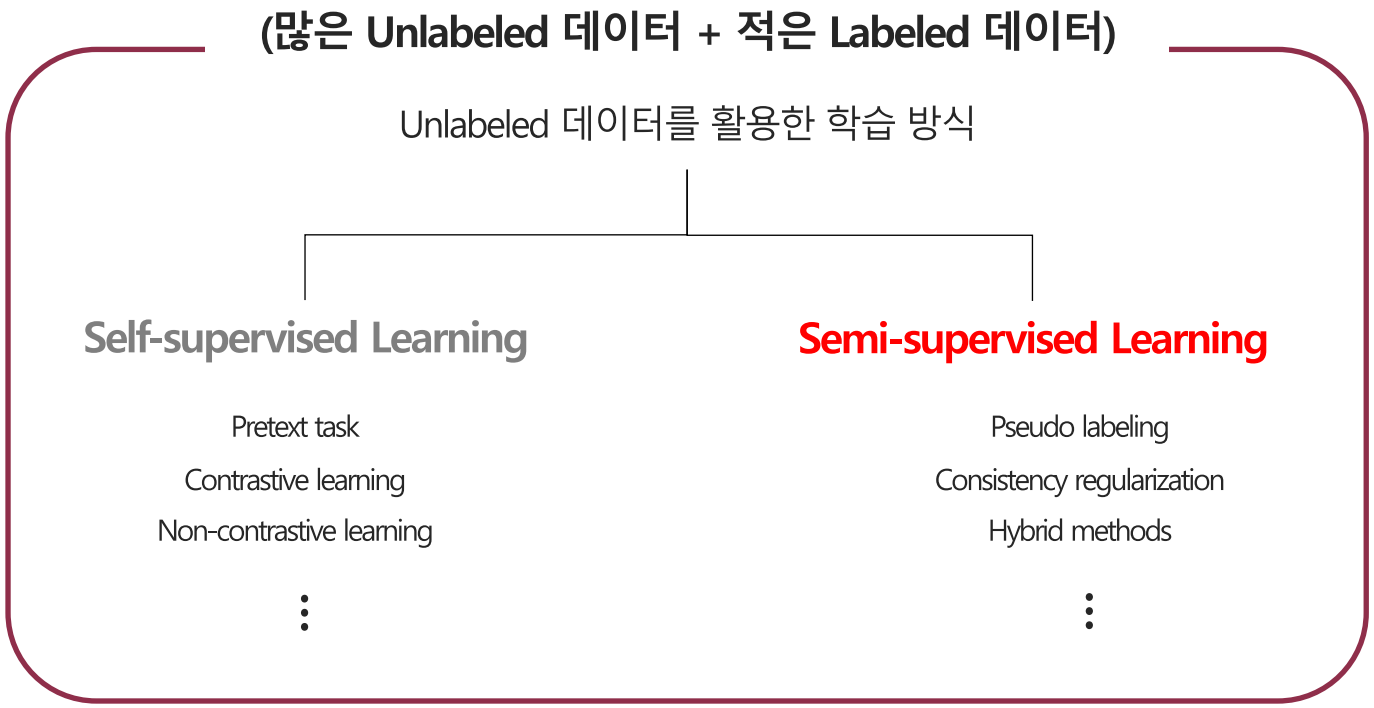





- https://thepathologist.com/inside-the-lab/picture-perfect-pathology

1. Introduction

❖ **(많은 Unlabeled 데이터 + 적은 Labeled 데이터) 활용 연구 개요**

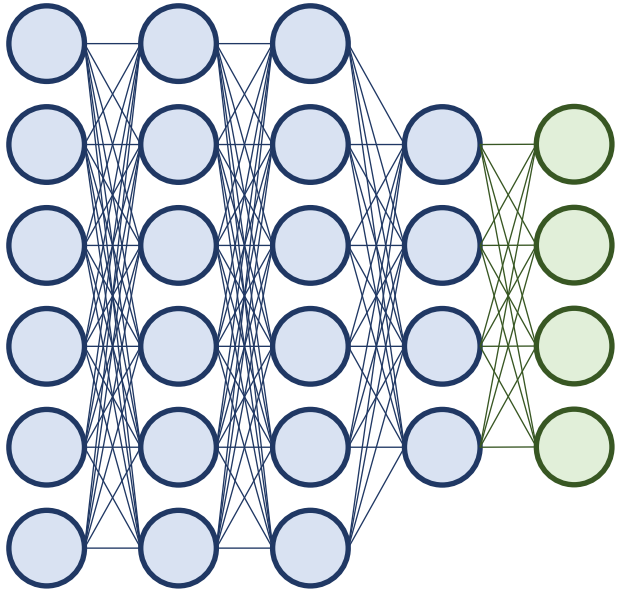
- 해당 연구 분야는 크게 Self-supervised Learning과 Semi-supervised Learning로 구분 가능
- 두 연구 분야는 Unlabeled 데이터 활용 방식에 차이 존재



1. Introduction

❖ **Self-supervised Learning** vs Semi-supervised Learning

- 두 단계로 나누어 원하는 예측 모델 학습 진행
  - ➢ 단계 1: Unlabeled 데이터에 대한 Supervision을 정의 후 신경망 모델 사전 학습 (Pre-training)
  - ➢ 단계 2: Labeled 데이터만 사용해 사전 학습된 모델을 미세 조정 진행 (Fine-tuning or Downstream)



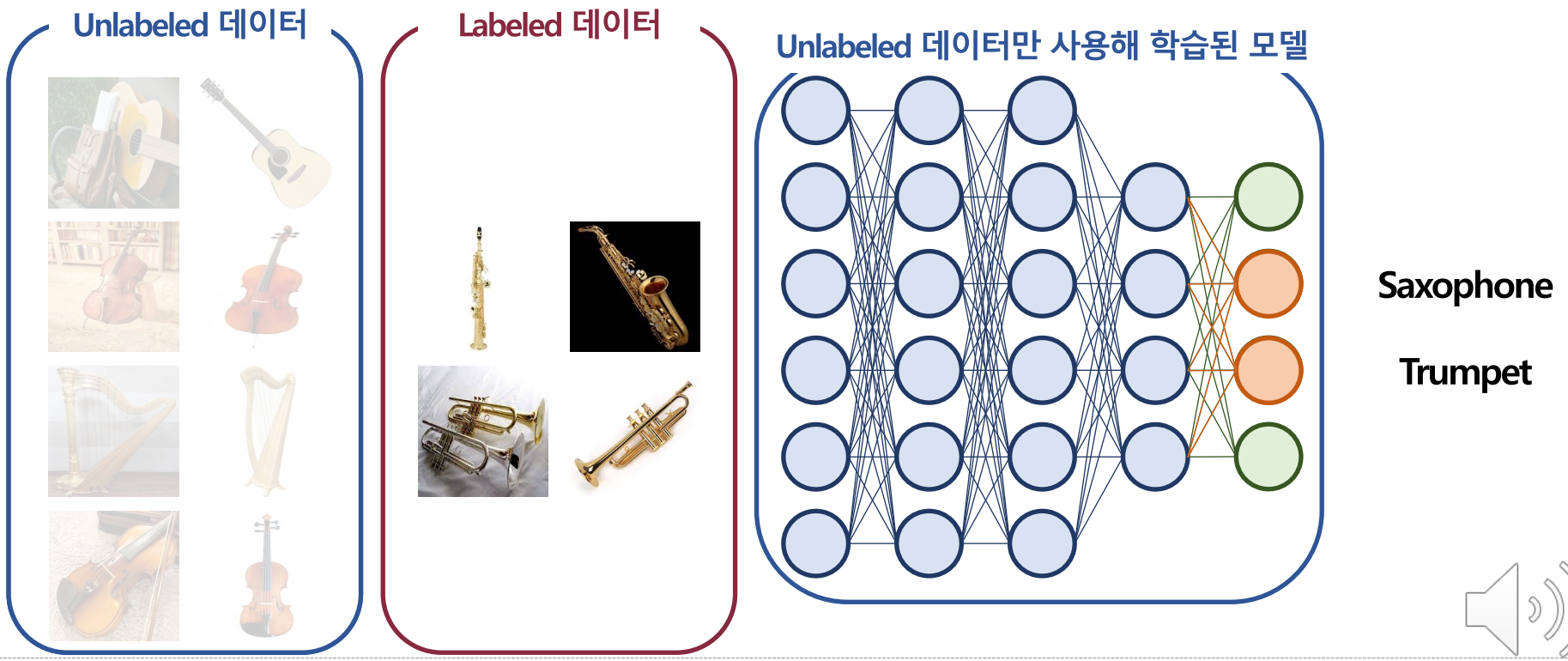
Self-supervision

- Pretext task
- Contrastive Learning
- Non-contrastive Learning

1. Introduction

❖ **Self-supervised Learning** vs Semi-supervised Learning

- 두 단계로 나누어 원하는 예측 모델 학습 진행
  - ➢ 단계 1: Unlabeled 데이터에 대한 Supervision을 정의 후 신경망 모델 사전 학습 (Pre-training)
  - ➢ 단계 2: Labeled 데이터만 사용해 사전 학습된 모델을 미세 조정 진행 (Fine-tuning or Downstream)

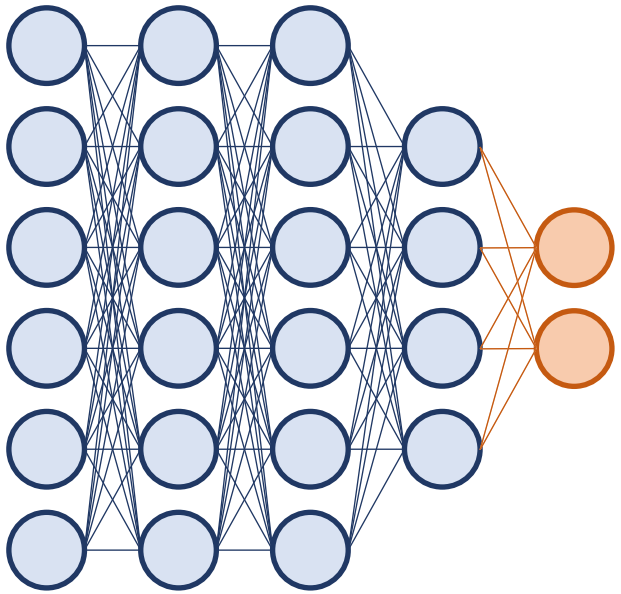


1. Introduction

❖ Self-supervised Learning vs **Semi-supervised Learning**

- Self-supervised Learning과 달리 한번에 (Unlabeled & Labeled) 데이터 모두 사용
- Labeled 데이터, Unlabeled 데이터 각각에 대한 손실 함수를 정의하여 신경망 모델 학습

$$Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$$



Saxophone

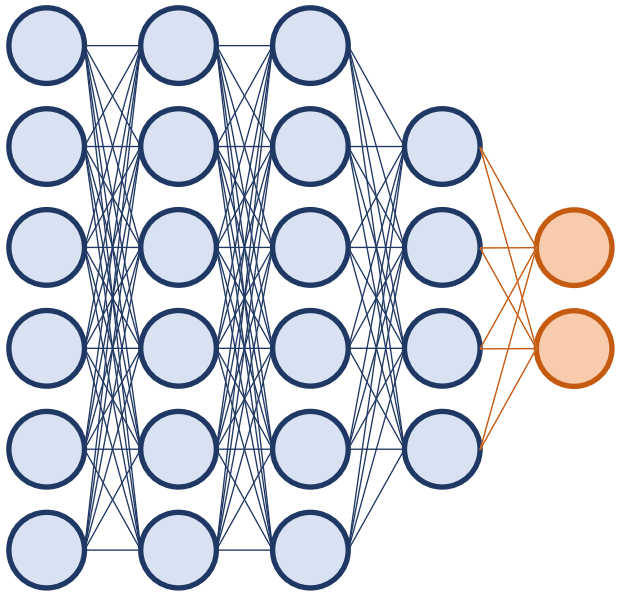
Trumpet

1. Introduction

❖ Self-supervised Learning vs **Semi-supervised Learning**

- Self-supervised Learning과 달리 한번에 (Unlabeled & Labeled) 데이터 모두 사용
- Labeled 데이터, Unlabeled 데이터 각각에 대한 손실 함수를 정의하여 신경망 모델 학습
- **'Unlabeled 데이터에 대한 손실 함수는 어떤 특징이 있어야 하지? 어떻게 정의하지?'**

$$Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$$



Saxophone

Trumpet

목차

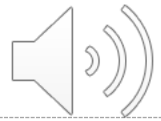
2. FixMatch

❖ **FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence**

- 2020년 11월 Neural Information Processing Systems(NeurallPS)에서 발표된 방법론
  - ➢ 2023년 1월 29일 기준 1,520회 인용되었으며 Google Research 소속 연구원들이 발표한 방법론
- 최신 Semi-supervised Learning 방법론의 기초가 되는 방법론

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence

Kihyuk Sohn* David Berthelot* Chun-Liang Li Zizhao Zhang Nicholas Carlini
Ekin D. Cubuk Alex Kurakin Han Zhang Colin Raffel
Google Research
{kihyuks,dberth,chunliang,zizhaoz,ncarlini,
cubuk,kurakin,zhanghan,craffel}@google.com



2. FixMatch

❖ **FixMatch**: Simplifying Semi-Supervised Learning with **Consistency** and Confidence

- Semi-supervised Learning 방법론 가정 중 Consistency Regularization에 관한 내용

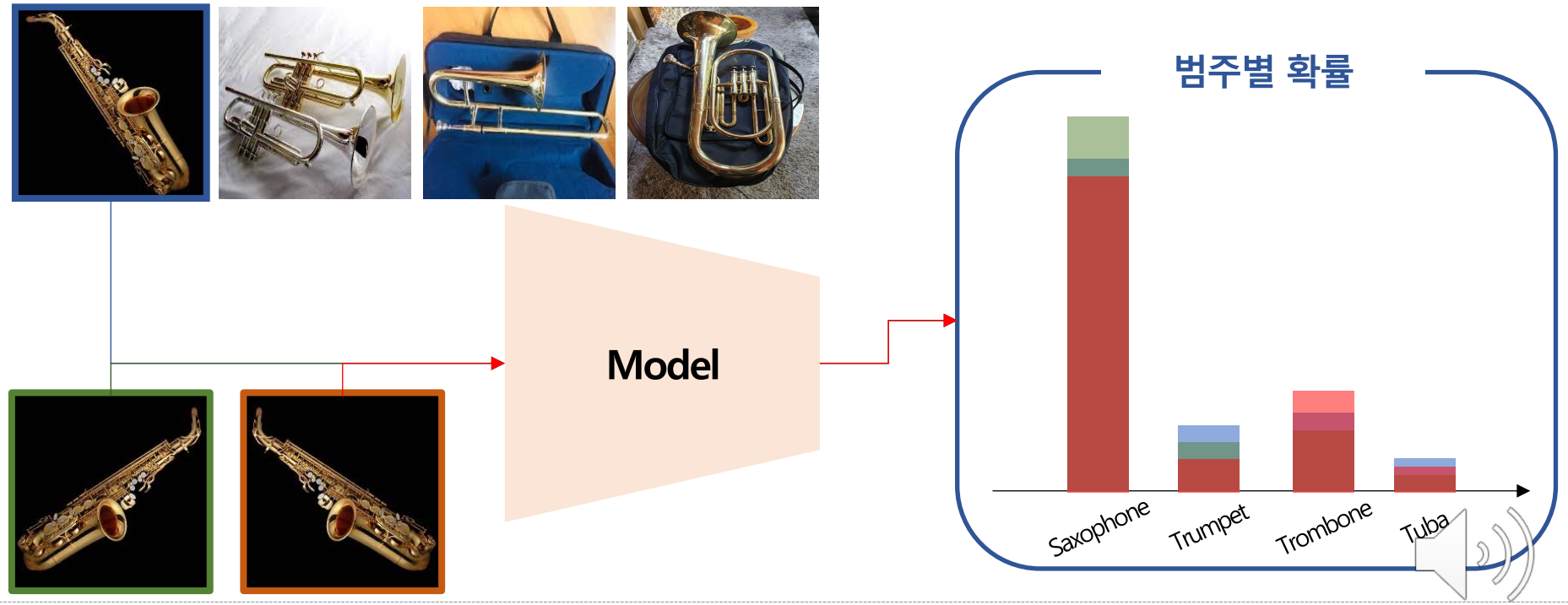
악기 종류를 분류하는 모델 학습 (총 범주는 4개)




2. FixMatch

❖ **FixMatch**: Simplifying Semi-Supervised Learning with **Consistency** and Confidence

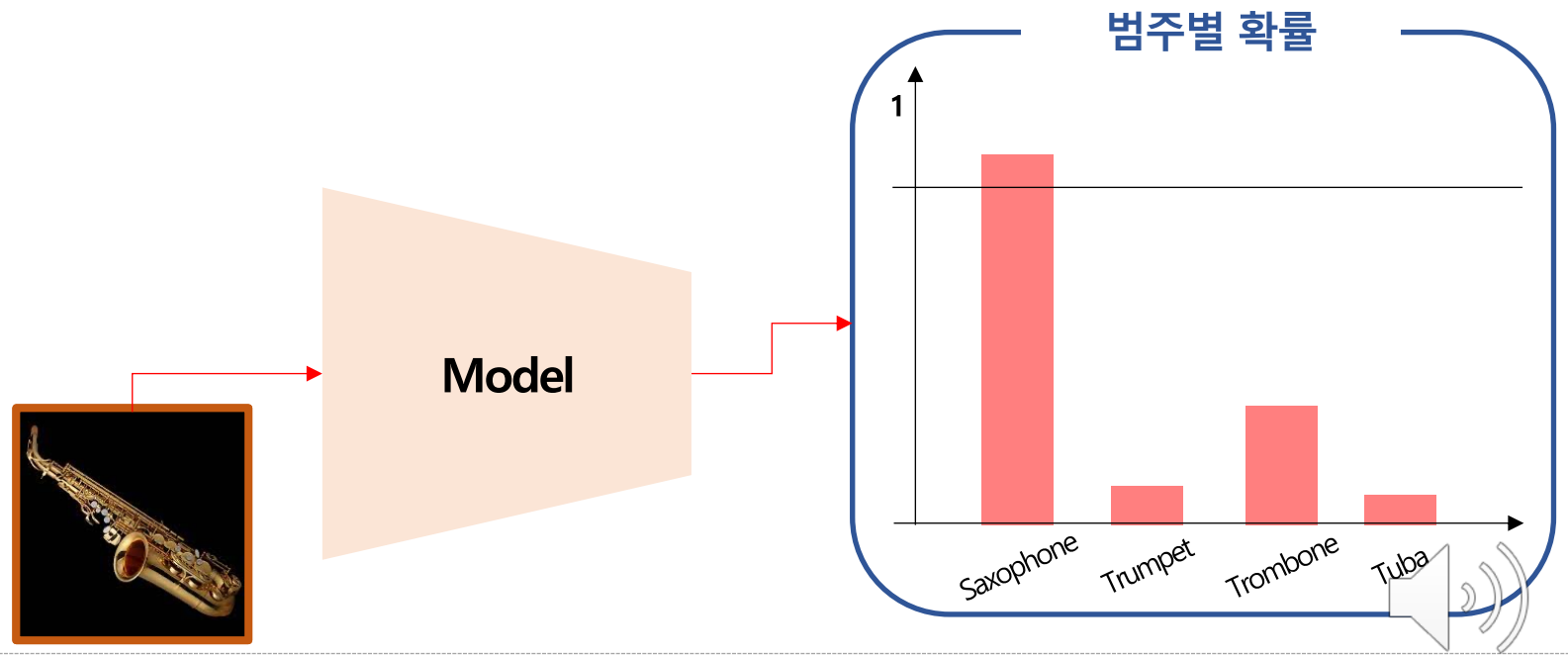
- Semi-supervised Learning 방법론 가정 중 Consistency Regularization에 관한 내용
- 입력 데이터에 증강 기법을 적용한 이미지를 다수 생성
- 증강된 이미지와 입력 데이터에 대한 예측 결과가 유사 해야 한다는 가정



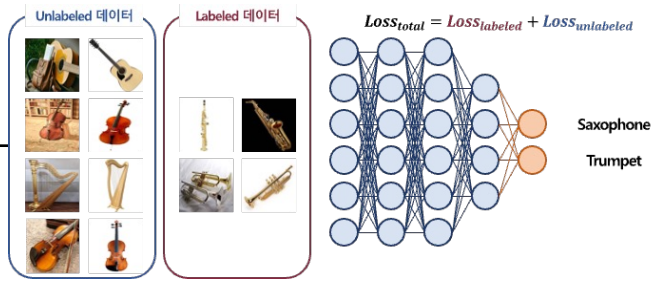
2. FixMatch

❖ **FixMatch**: Simplifying Semi-Supervised Learning with Consistency and **Confidence**

- Confidence 의미는 범주별 확률 값을 얼마나 신뢰할 수 있을 지를 의미
- 특히, Unlabeled 데이터에 대한 예측 확률이 어떻게 사용할 지에 관한 지표

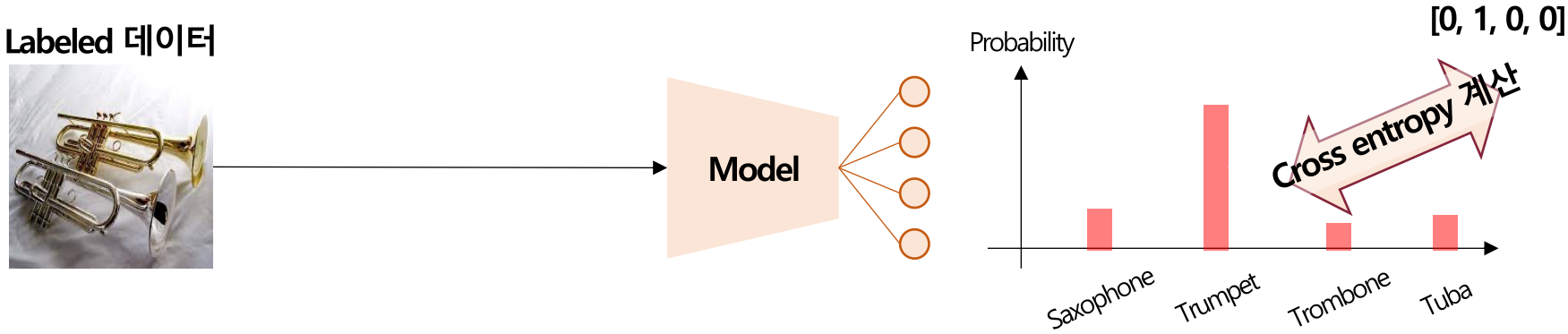


2. FixMatch



❖ FixMatch 에서 손실 함수 산출 과정 (Labeled 데이터)

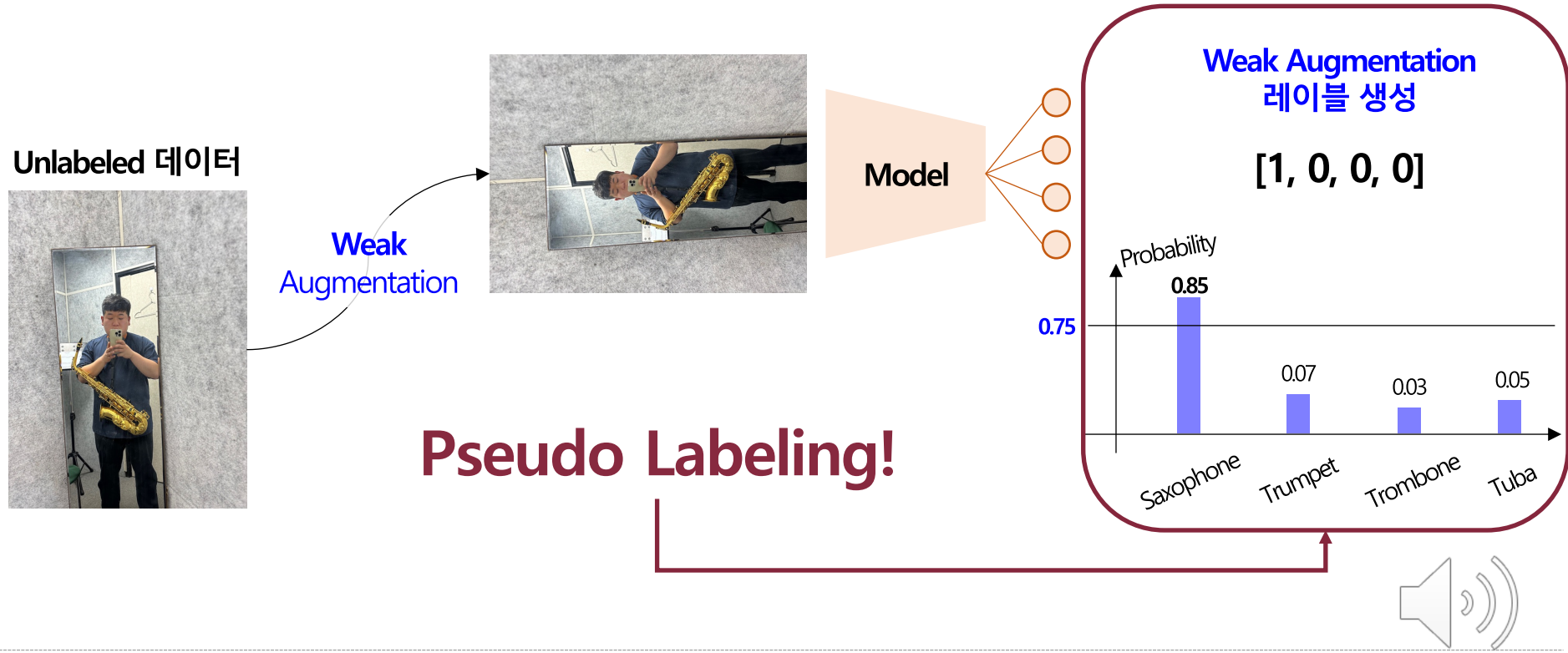
- 일반적인 지도학습과 같은 손실 함수를 사용
  - ➢ FixMatch는 분류 문제에 대한 Semi-supervised Learning 방법론이기에 Cross entropy 사용



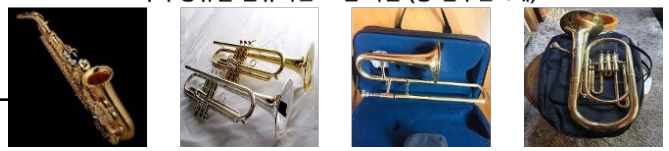
2. FixMatch

❖ FixMatch 에서 손실 함수 산출 과정 (Unlabeled 데이터)

- Unlabeled 데이터에 Weak & Strong Augmentation 적용
  - ➢ **Weak Augmentation**: Flip, Shift와 같이 객체에 변화가 발생하지 않는 이미지 데이터 증강 기법
  - ➢ Strong Augmentation: 픽셀 값을 변화시키는 여러 증강 기법 조합으로 생성

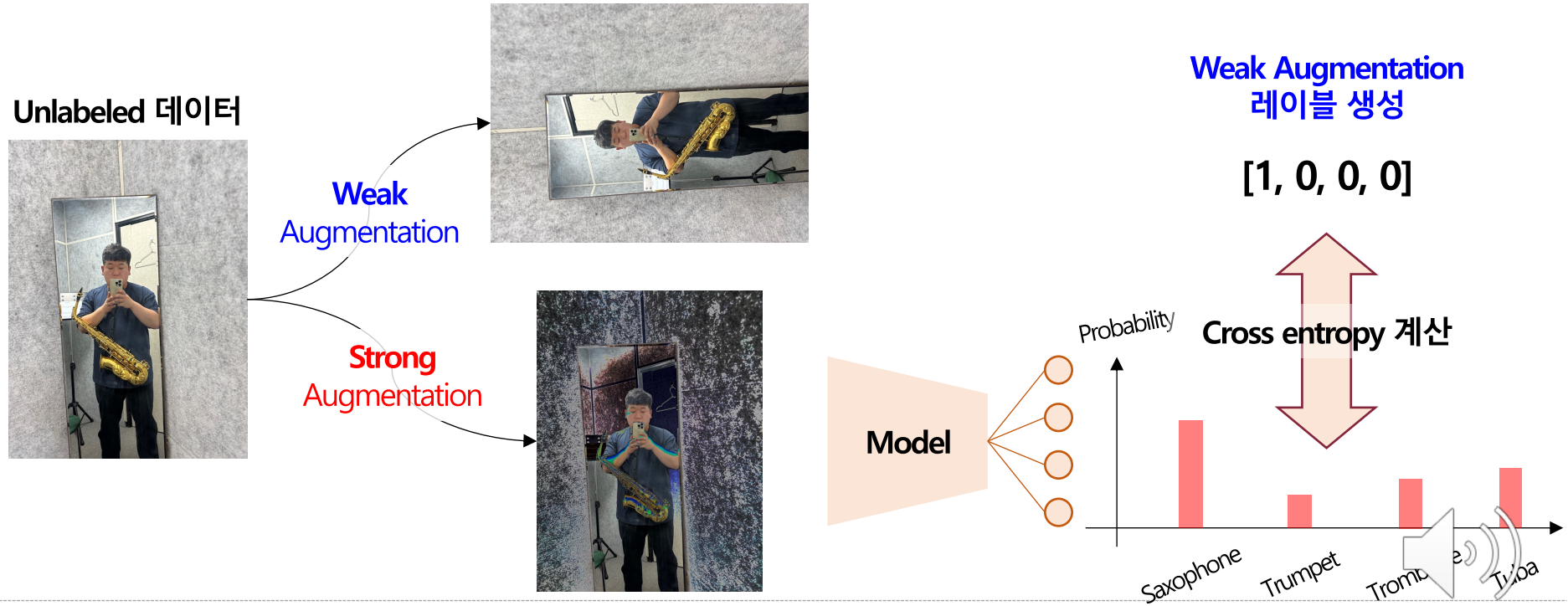


2. FixMatch

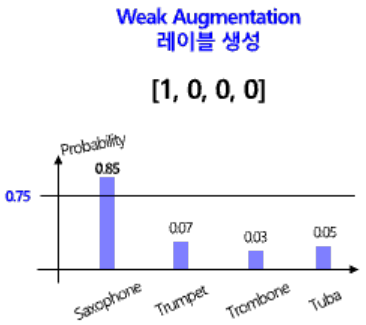


❖ FixMatch 에서 손실 함수 산출 과정 (Unlabeled 데이터)

- Unlabeled 데이터에 Weak & Strong Augmentation 적용
  - ➢ Weak Augmentation: Flip, Shift와 같이 객체에 변화가 발생하지 않는 이미지 데이터 증강 기법
  - ➢ Strong Augmentation: 픽셀 값을 변화시키는 여러 증강 기법 조합으로 생성

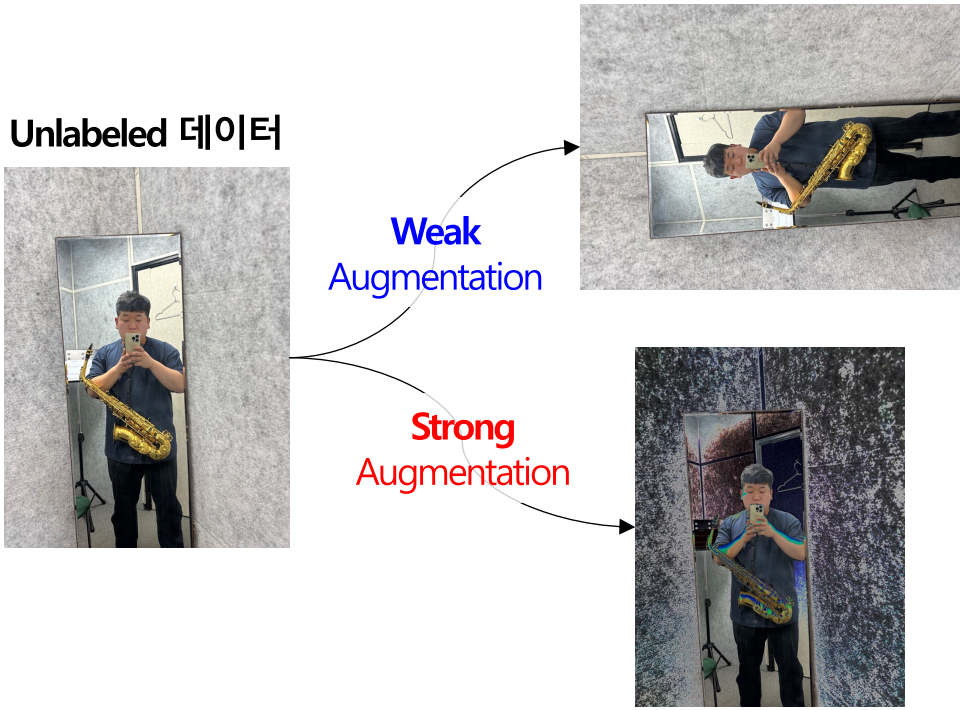


2. FixMatch

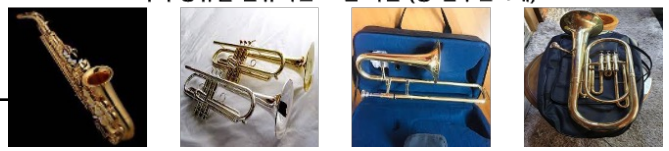


❖ **FixMatch 에서 손실 함수 산출 과정 (Unlabeled 데이터)**

- Unlabeled 데이터에 두 데이터 증강 기법을 적용해 서로 다른 이미지 생성
- Weak Augmented 이미지 예측 값 중 임계값 이상인 경우 Pseudo Labeling 진행

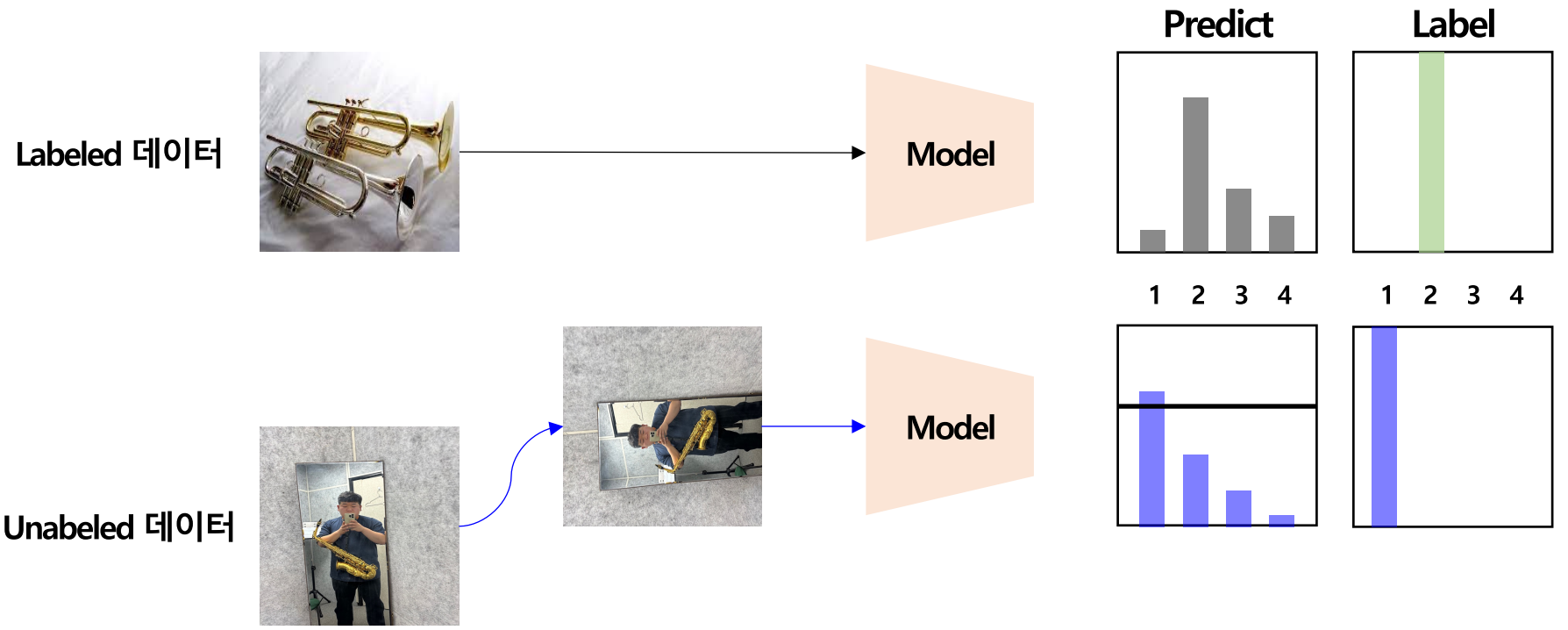


2. FixMatch

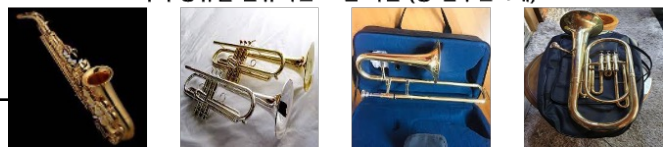


❖ FixMatch 요약

- $Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$ 를 계산해 심층 신경망 모델 학습

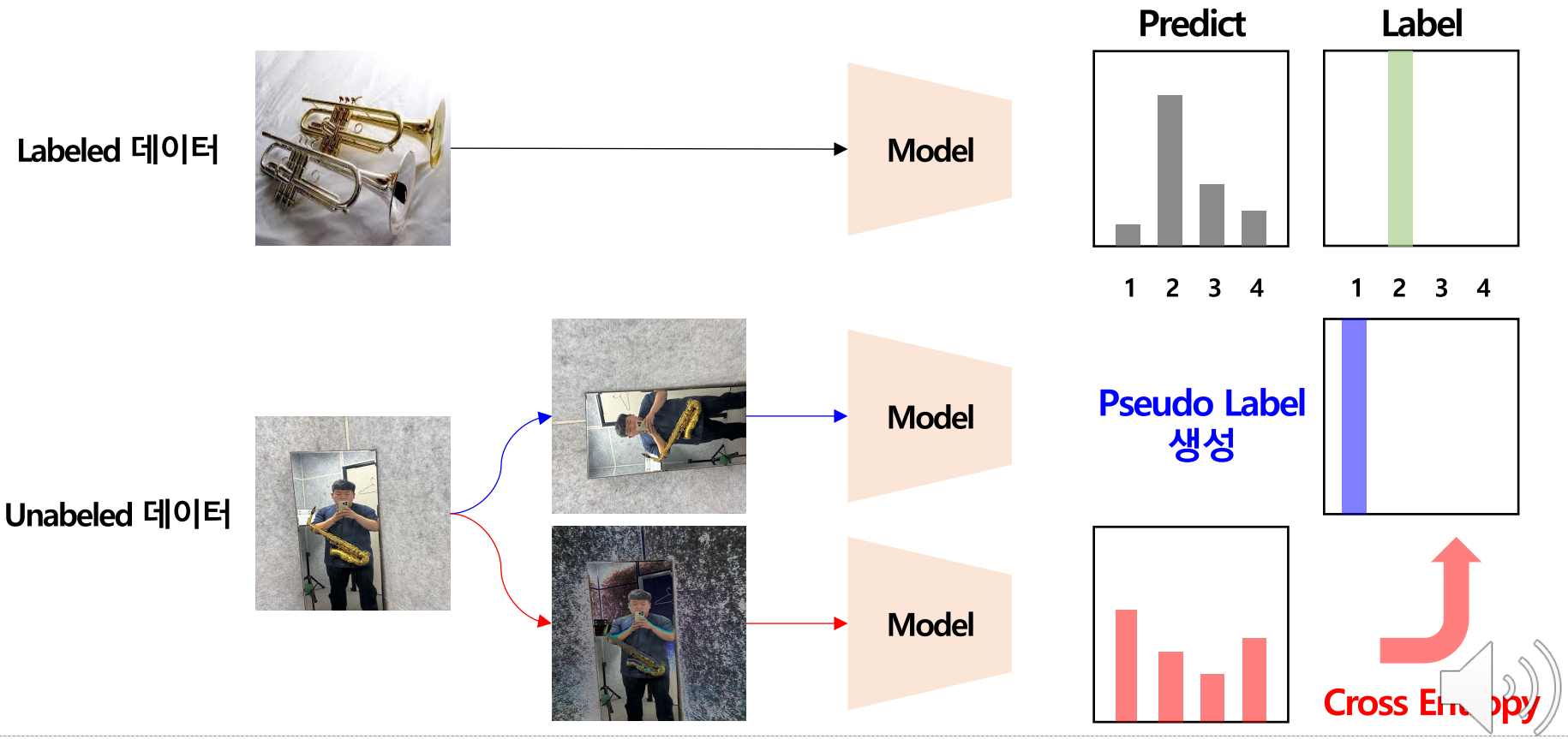


2. FixMatch

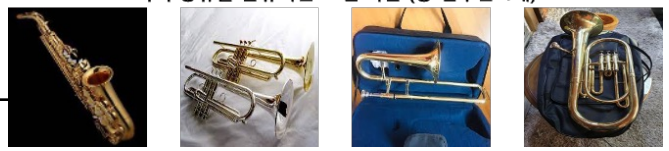


❖ FixMatch 요약

- $Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$ 를 계산해 심층 신경망 모델 학습



2. FixMatch



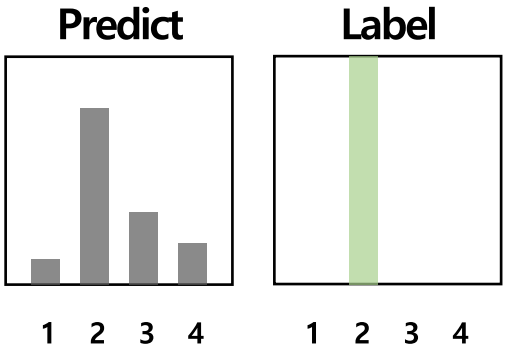
❖ FixMatch 요약

- $Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$ 를 계산해 심층 신경망 모델 학습

Labeled 데이터

$$Loss_{labeled} = \frac{1}{B}\sum_{b=1}^{B} CE(p_b, \widehat{p_b})$$

Model



임계값 넘는 경우만 손실함수 산출

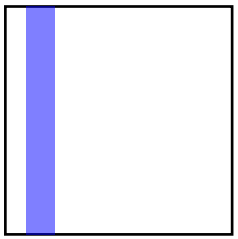
Unlabeled 데이터

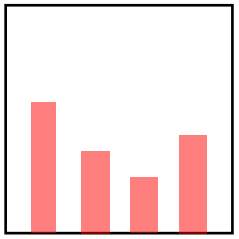
$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}(\max(q_{b,w}) > \tau)\, CE(q_{b,w}, q_{b,s})$$

Model

Pseudo Label ↔ Strong Augment 예측 값
Cross Entropy

Pseudo Label 생성





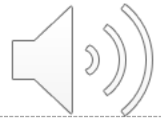
Cross Entropy

2. FixMatch

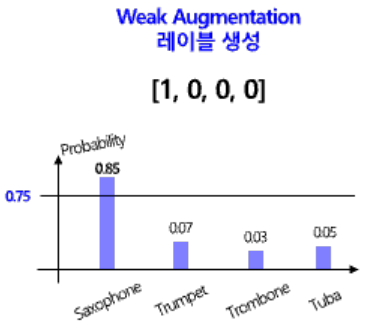
❖ FixMatch 실험 결과

- Strong Augmentation 조합 방법 두가지에 대한 실험과 기존 방법론 비교
  - ➢ RandAugment(RA), CTAAugment(CTA)
- 범주별 관측치 수를 변경하며 실험 진행해 학습 데이터 수가 적을 때 성능이 뛰어남을 확인

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels	1000 labels
Π-Model	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11	-	18.96±1.92	7.54±0.36	26.23±0.82
Pseudo-Labeling	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19	-	20.21±1.09	9.94±0.61	27.99±0.83
Mean Teacher	-	32.32±2.30	9.19±0.19	-	53.91±0.57	35.83±0.24	-	3.57±0.11	3.42±0.07	21.43±2.39
MixMatch	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28	10.41±0.61
UDA	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	**2.46**±0.24	7.66±0.56
ReMixMatch	**19.10**±9.64	**5.44**±0.05	4.72±0.13	**44.28**±2.06	**27.43**±0.31	**23.03**±0.56	**3.34**±0.20	**2.92**±0.48	2.65±0.08	**5.23**±0.45
FixMatch (RA)	**13.81**±3.37	**5.07**±0.65	**4.26**±0.05	48.85±1.75	28.29±0.11	**22.60**±0.12	**3.96**±2.17	**2.48**±0.38	**2.28**±0.11	7.98±1.50
FixMatch (CTA)	**11.39**±3.35	**5.07**±0.33	**4.31**±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	**2.64**±0.64	**2.36**±0.19	**5.17**±0.63

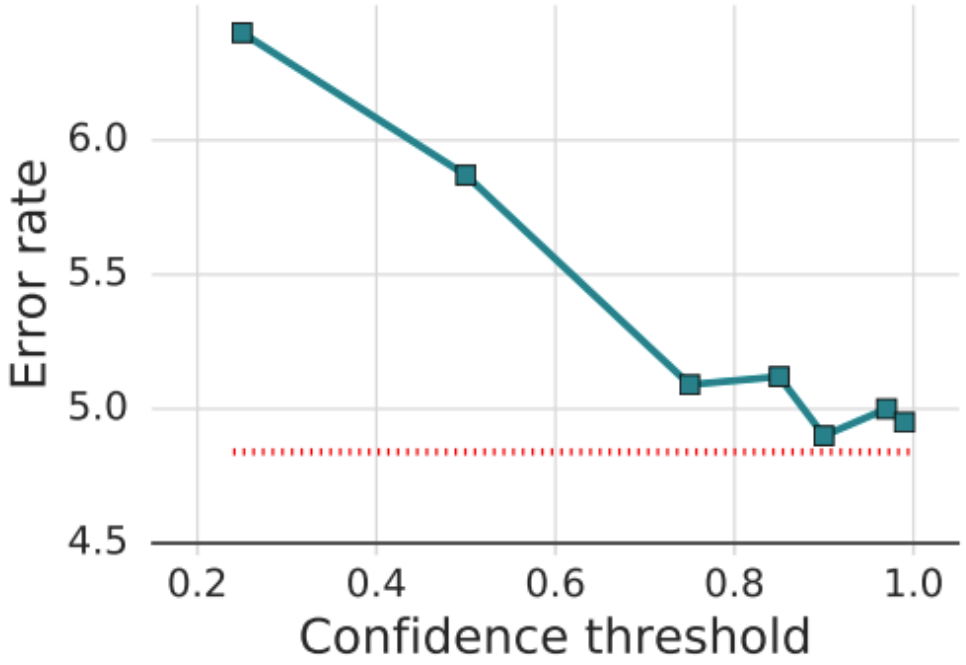


2. FixMatch



❖ FixMatch 실험 결과 – Ablation Study

- Weak Augmentation Pseudo Label 생성을 위한 임계값 변화에 따른 성능 변화
- Confidence 값이 높은 데이터만 손실 함수 산출에 사용한 경우 성능이 높음

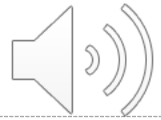


2. FixMatch

❖ Related works in the FixMatch paper

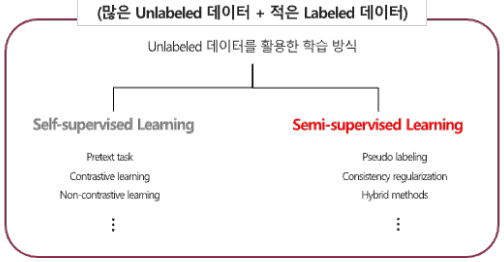
- 관련 연구 세션을 살펴보면 아래와 같은 표 존재
- 기존 방법론을 빠르게 살펴본다면 위 정보가 유용할 것으로 생각
- 차이점 위주로 살펴보길 권유

Algorithm	Artificial label augmentation	Prediction augmentation	Artificial label post-processing	Notes
TS / Π-Model	Weak	Weak	None	
Temporal Ensembling	Weak	Weak	None	Uses model from earlier in training
Mean Teacher	Weak	Weak	None	Uses an EMA of parameters
Virtual Adversarial Training	None	Adversarial	None	
UDA	Weak	Strong	Sharpening	Ignores low-confidence artificial labels
MixMatch	Weak	Weak	Sharpening	Averages multiple artificial labels
ReMixMatch	Weak	Strong	Sharpening	Sums losses for multiple predictions
FixMatch	Weak	Strong	Pseudo-labeling	



목차

3. SelfMatch



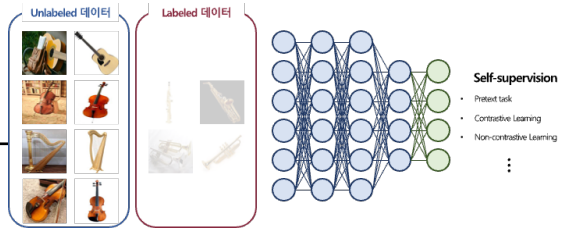
❖ SelfMatch: Combining Contrastive Self-supervision and Consistency

- 2021년 1월 arXiv에 올라온 방법론이며 삼성 SDS 소속 연구원들이 제안한 방법론
  - ➢ 2023년 1월 29일 기준 26회 인용
- Unlabeled 데이터 활용 방안 중 두 가지 학습 방식을 모두 사용

SelfMatch: Combining Contrastive Self-Supervision and Consistency for Semi-Supervised Learning

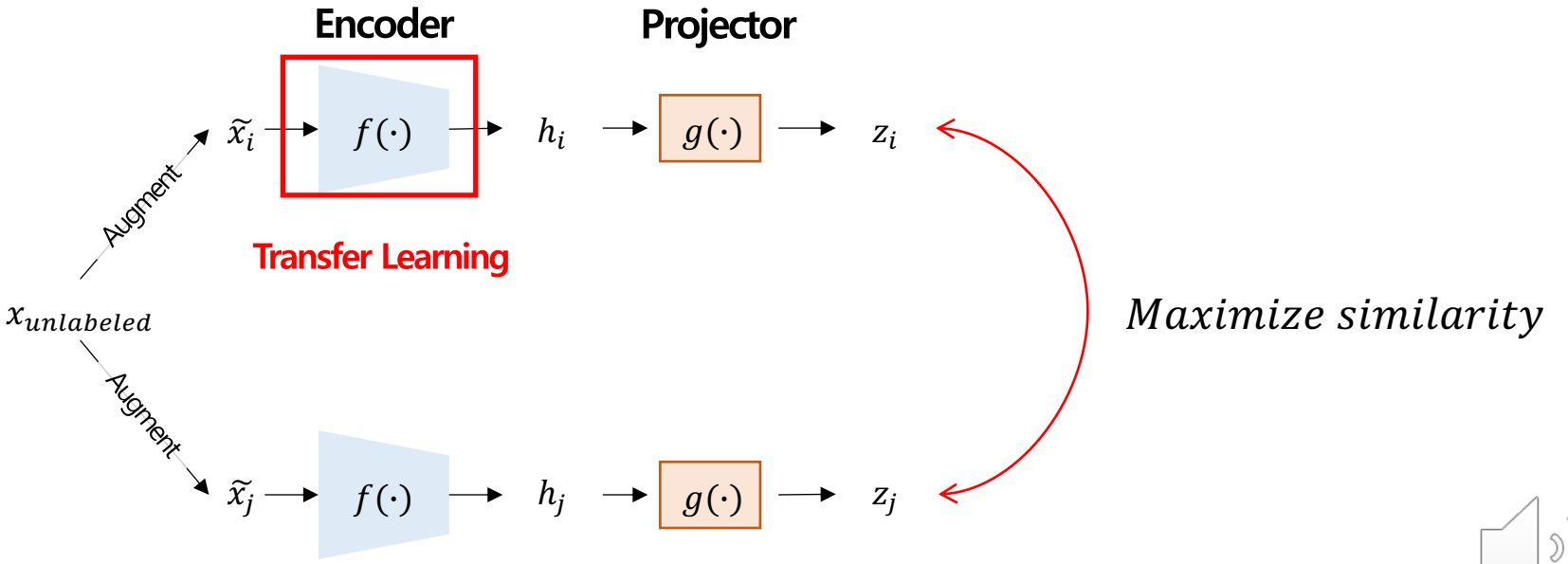
Byoungjip Kim, Jinho Choo, Yeong-Dae Kwon, Seongho Joe, Seungjai Min, Youngjune Gwon
Samsung SDS
{bjip.kim, jinho12.choo, y.d.kwon, drizzle.cho,
seungjai.min, gyj.gwon}@samsung.com

3. SelfMatch



❖ SelfMatch: Combining Contrastive Self-supervision and Consistency

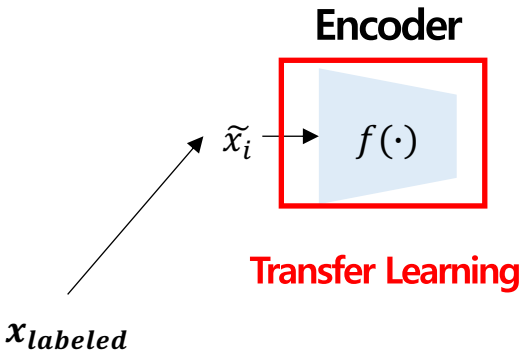
- SelfMatch 는 두 단계 학습으로 구성된 Semi-supervised Learning 방법론
  - ➢ 단계 1: Unlabeled 데이터를 사용해 **SimCLR** 방식으로 Encoder 사전학습
    - ✓ **같은 데이터에서 증강된 두 이미지(Positive pair) 는 가깝도록 학습**
    - ✓ **특정 이미지가 아닌 나머지 이미지(Negative samples) 는 모두 멀도록 학습**

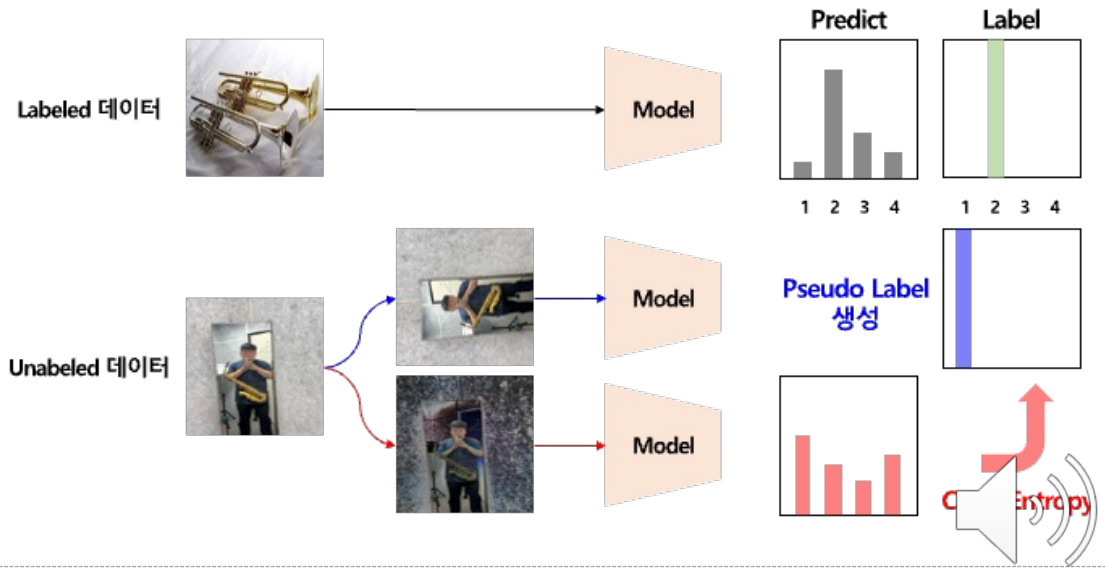


3. SelfMatch

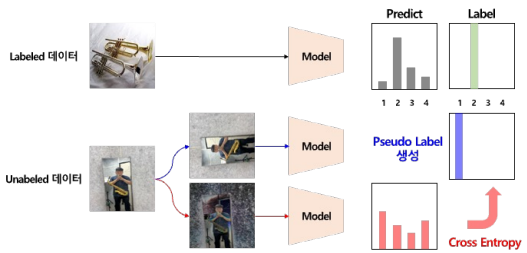
❖ SelfMatch: Combining Contrastive Self-supervision and Consistency

- SelfMatch 는 두 단계 학습으로 구성된 Semi-supervised Learning 방법론
  - ➢ 단계 1: Unlabeled 데이터를 사용해 SimCLR 방식으로 Encoder 사전학습
  - ➢ **단계 2: 사전 학습된 Encoder와 분류기에 대해 Labeled & Unlabeled 데이터를 사용해 FixMatch 방식으로 미세 조정 진행**





3. SelfMatch

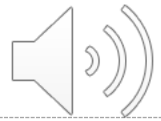


❖ SelfMatch 실험 결과

- 사전 학습을 추가하여 기존 Semi-supervised Learning 방법론 대비 성능 향상 성공
- Self-supervised Learning 과 Semi-supervised Learning 을 모두 활용했다는 것이 특징
- 하지만 효과적으로 (Unlabeled & Label) 데이터를 모두 사용했다고 평가하기는 어려움

Table 1: Comparison of accuracy for CIFAR-10 and SVHN.

	CIFAR-10			SVHN		
Method	40 labels	250 labels	4000 labels	40 labels	250 labels	1000 labels
Supervised		**95.87**			**97.41**	
Pseudo-Label	-	50.22±0.43	83.91±0.28	-	79.79±1.09	90.06±0.61
Π-Model	-	45.74±3.87	85.99±0.38	-	81.04±1.92	92.46±0.36
Mean Teacher	-	67.68±2.30	90.81±0.19	-	96.43±0.11	96.58±0.07
MixMatch	52.46±11.50	88.95±0.86	93.58±0.10	57.45±14.53	96.02±0.23	96.5±0.28
UDA	70.95±5.93	91.18±1.08	95.12±0.18	47.37±20.51	94.31±2.76	97.54±0.24
ReMixMatch	80.90±9.64	94.56±0.05	95.28±0.13	**96.66**±0.20	97.08±0.48	97.35±0.08
FixMatch(RA)	86.19±3.37	94.93±0.65	95.74±0.05	96.04±2.17	**97.52**±0.38	**97.72**±0.11
SelfMatch	**93.19**±1.08	**95.13**±0.26	**95.94**±0.08	**96.58**±1.02	**97.37**±0.43	**97.49**±0.07



목차

4. SimMatch

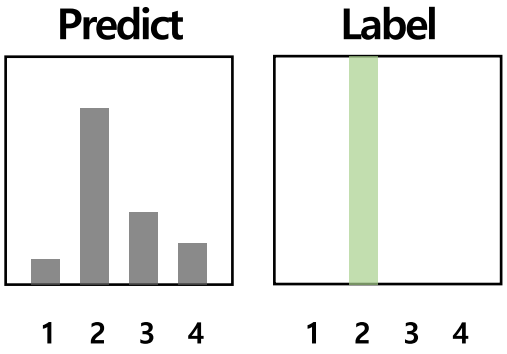
❖ FixMatch 요약

- $Loss_{total} = Loss_{labeled} + Loss_{unlabeled}$ 를 계산해 심층 신경망 모델 학습

Labeled 데이터

$$Loss_{labeled} = \frac{1}{B}\sum_{b=1}^{B} CE(p_b, \widehat{p_b})$$

Model

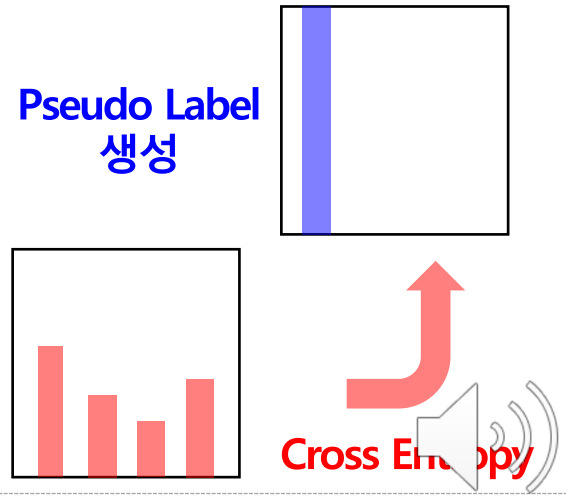


Unlabeled 데이터

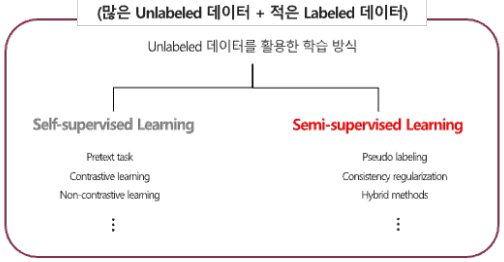
$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}\left(\max(q_{b,w}) > \tau\right) CE(q_{b,w}, q_{b,s})$$

Model

Model



4. SimMatch



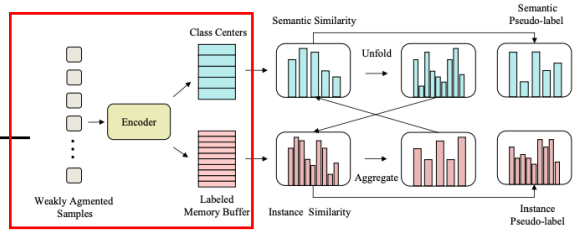
❖ SimMatch: Semi-supervised Learning with Similarity Matching

- 2022년 5월 Computer Vision and Pattern Recognition(CVPR)에서 발표된 방법론
  - ➢ 2023년 1월 29일 기준 25회 인용되었으며 여러 학교 및 기관이 합심해서 진행한 연구
- 기존 FixMatch에 Similarity 개념을 추가해서 성능 향상 성공

SimMatch: Semi-supervised Learning with Similarity Matching

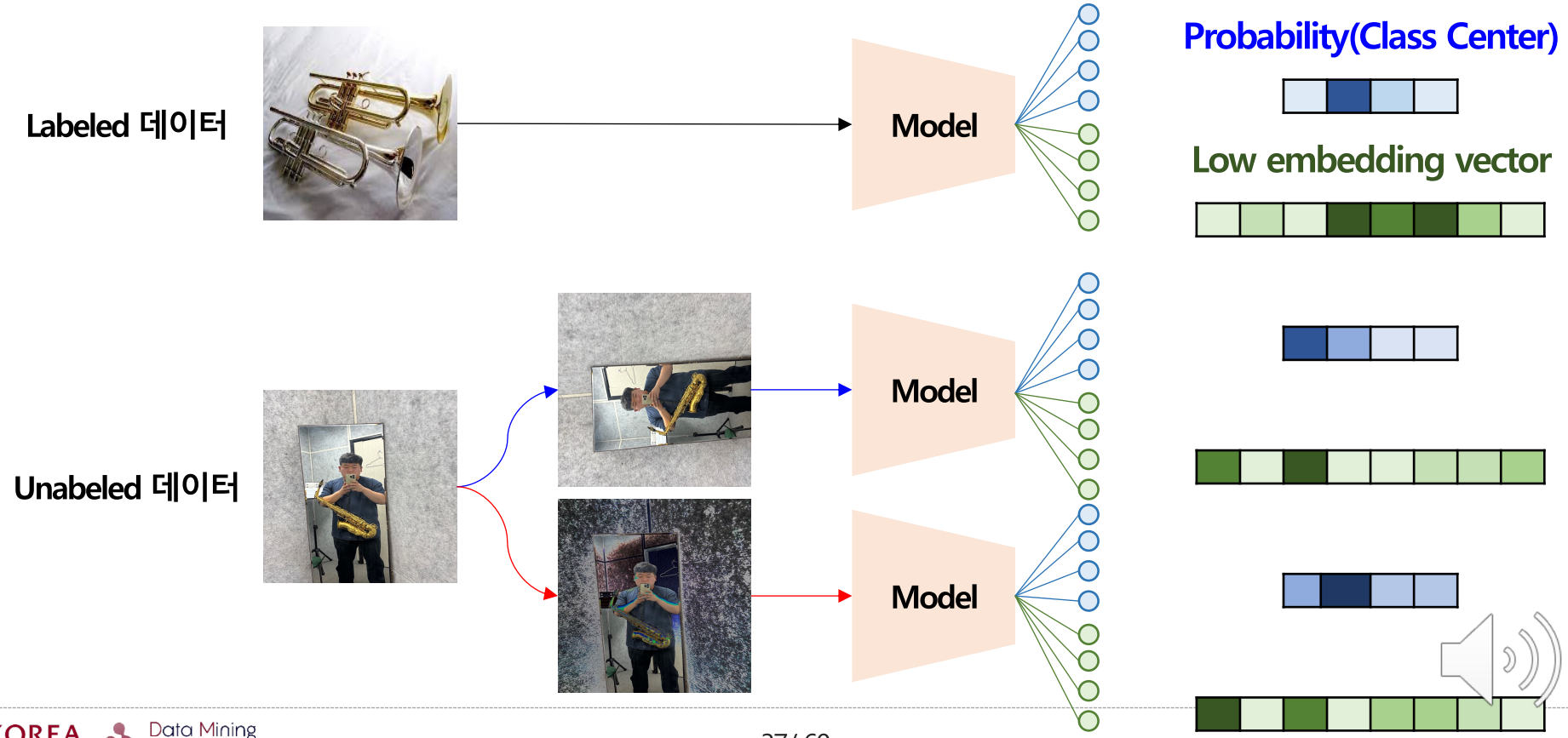
Mingkai Zheng[1,2] Shan You[2*]
Lang Huang[3] Fei Wang[4] Chen Qian[2] Chang Xu[1]
[1]School of Computer Science, Faculty of Engineering, The University of Sydney
[2]SenseTime Research [3] The University of Tokyo
[4]University of Science and Technology of China

4. SimMatch

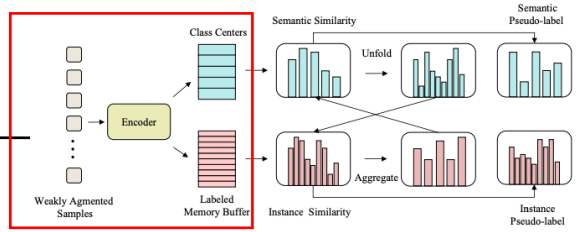


❖ SimMatch: Semi-supervised Learning with Similarity Matching

- FixMatch와 유사하게 Labeled 데이터와 Weak Augmented, Strong Augmented 데이터 사용
- Model 출력 값을 두 가지(Probability, Low embedding vector) 산출

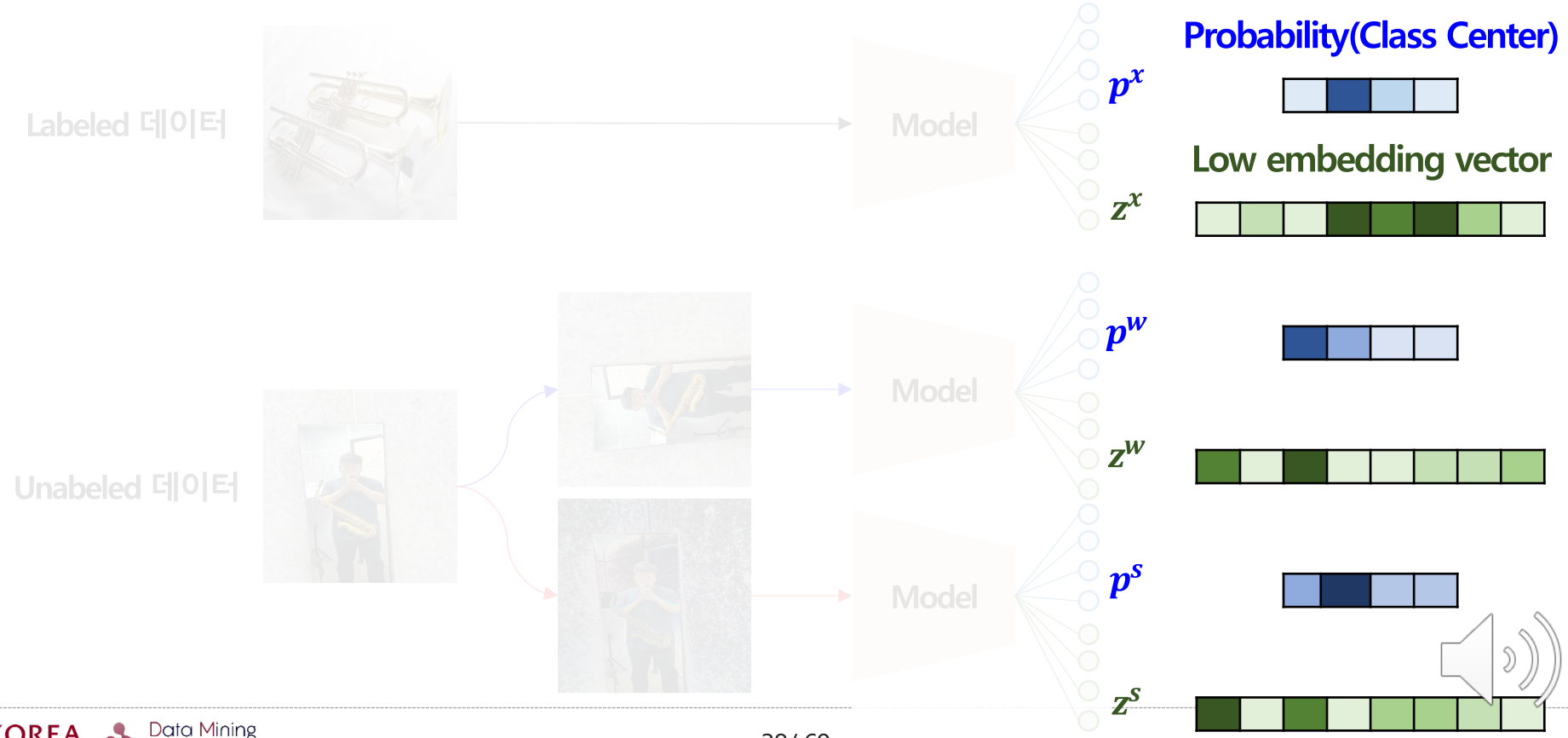


4. SimMatch

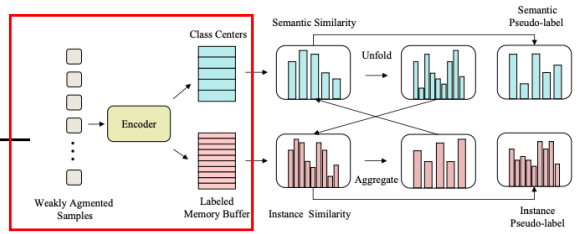


❖ SimMatch: Semi-supervised Learning with Similarity Matching

- Model 출력 값을 두 가지(Probability, Low embedding vector) 산출
- **Probability는 Semantic Similarity 역할/ Low embedding vector는 Instance Similarity 역할**



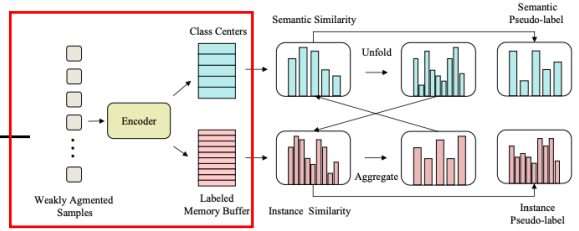
4. SimMatch



❖ SimMatch 내 **Labeled 데이터**에 대한 손실 함수

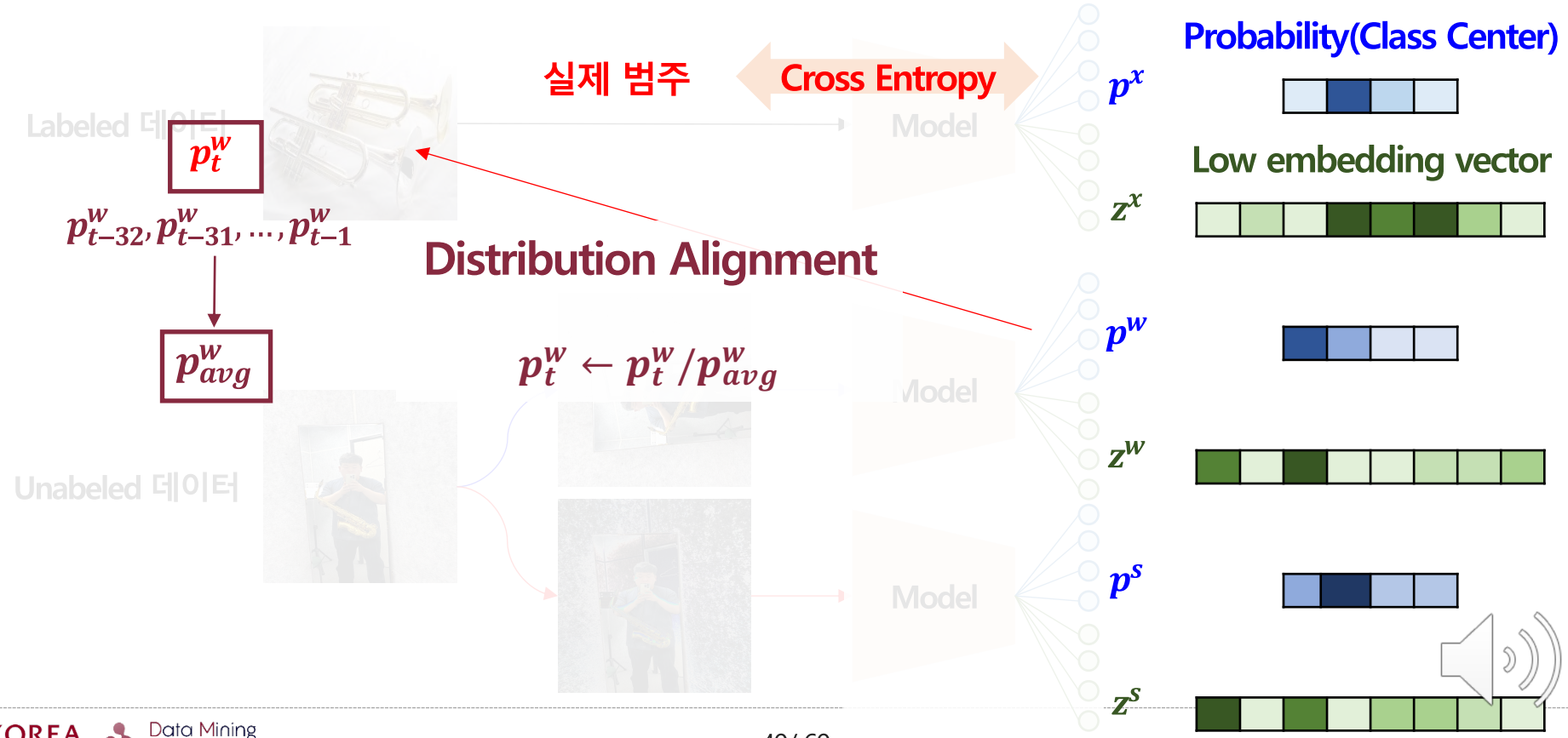
- Labeled 데이터에 대한 손실함수는 실제 Label과 예측 확률 사이 Cross entropy 계산

4. SimMatch

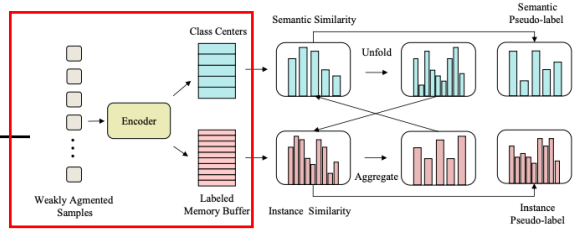


❖ SimMatch 내 **Unabeled 데이터**에 대한 손실 함수(**Distribution Alignment, DA**)

- Pseudo Label(p_t^w) 이 너무 급격하게 변화하면 학습에 부정적인 영향을 미친다고 알려짐
- 따라서 Weak augmented 데이터에 대한 과거 32시점간 예측 확률 값 평균을 사용해 p^w 보정



4. SimMatch



❖ SimMatch 내 **Unabeled 데이터**에 대한 손실 함수(**Distribution Alignment, DA**)

- p^w 값을 보정해 분포를 안정화 한 후 p^s와 p^w 사이 Cross entropy 계산(Unlabeled 손실함수)

실제 범주 — Cross Entropy

p_t^w

$p_{t-32}^w, p_{t-31}^w, \cdots, p_{t-1}^w$

p_{avg}^w

Distribution Alignment

$p_t^w \leftarrow p_t^w / p_{avg}^w$

$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}(\max(DA(p^w)) > \tau)\, CE(DA(p^w), p^s)$$

Probability(Class Center)

p^x

Low embedding vector

z^x

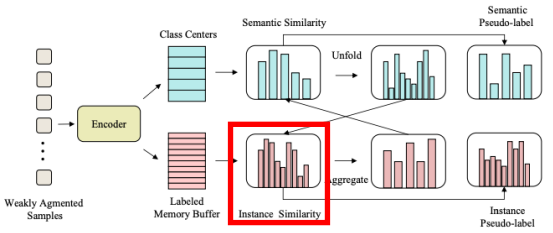
p^w

z^w

p^s

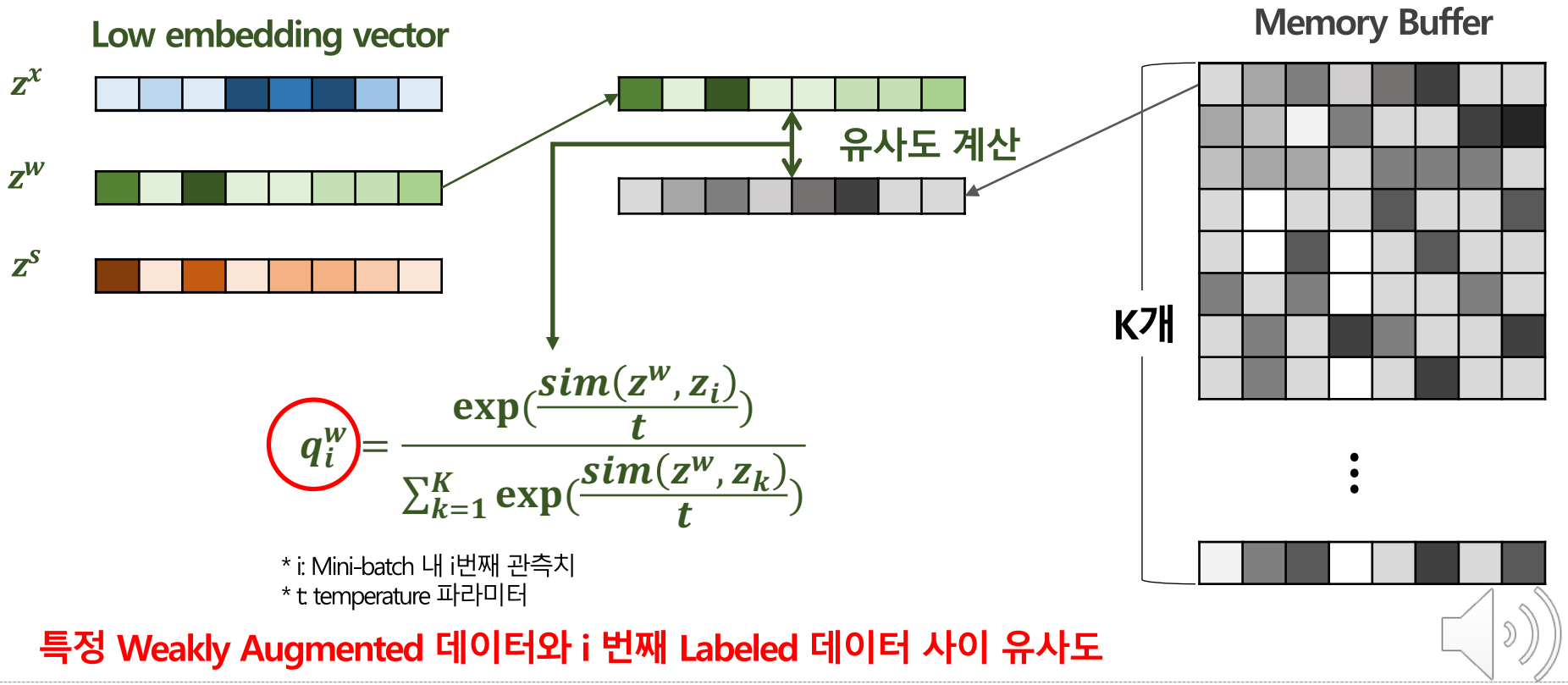
z^s

4. SimMatch



❖ Instance Similarity 정의

- Memory Buffer는 모든 Labeled 데이터에 대한 Embedding vector로 구성
- 특정 Embedding vector와 Memory Buffer 사이 유사도 계산 → **Instance Similarity(q) 분포**

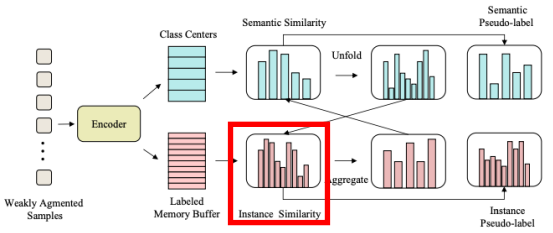


$$q_i^w = \frac{\exp\left(\frac{sim(z^w, z_i)}{t}\right)}{\sum_{k=1}^{K} \exp\left(\frac{sim(z^w, z_k)}{t}\right)}$$

* i: Mini-batch 내 i번째 관측치
* t: temperature 파라미터

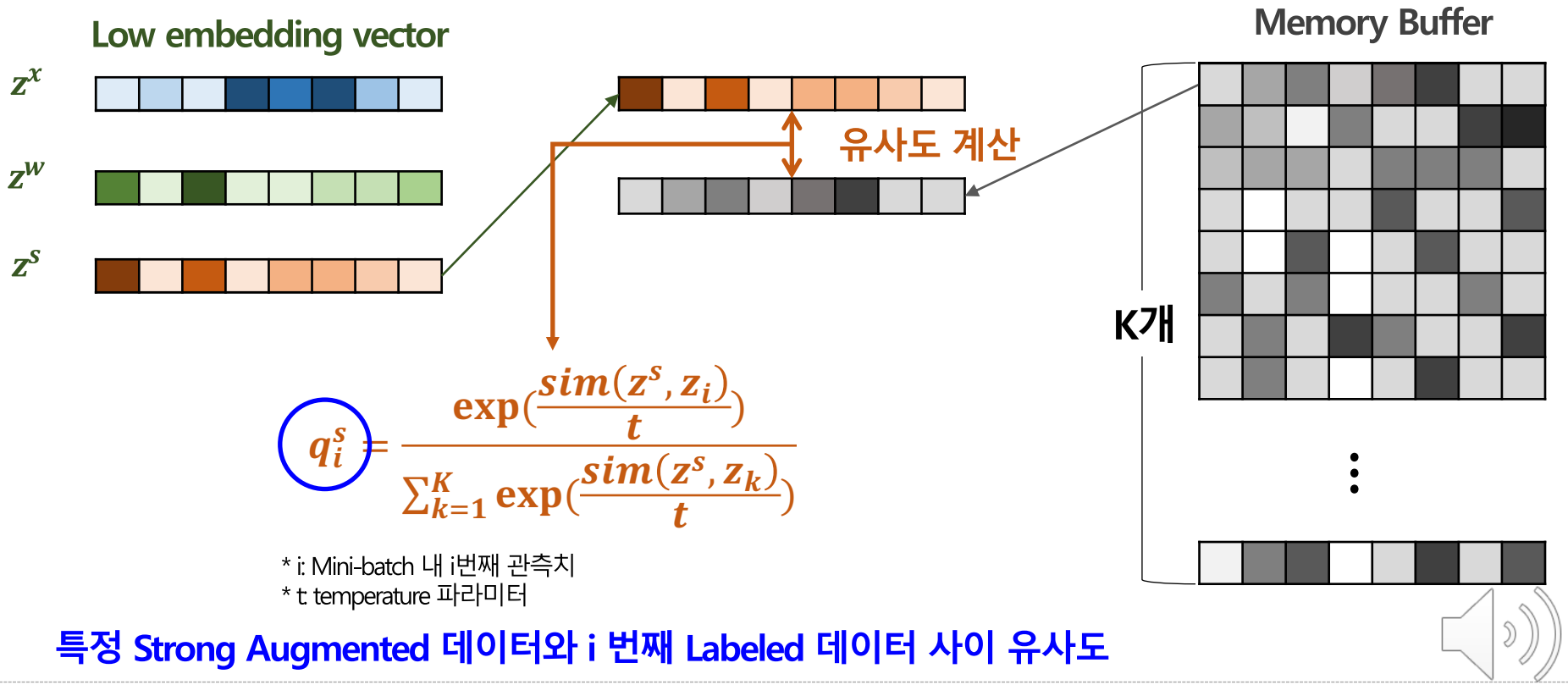
특정 Weakly Augmented 데이터와 i 번째 Labeled 데이터 사이 유사도

4. SimMatch



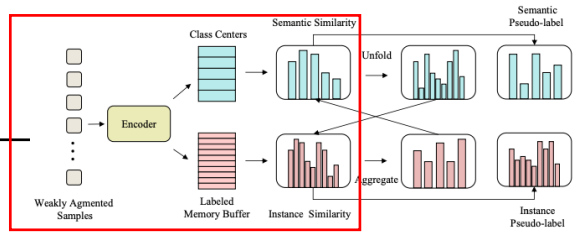
❖ Instance Similarity 정의

- Memory Buffer는 모든 Labeled 데이터에 대한 Embedding vector로 구성
- 특정 Embedding vector와 Memory Buffer 사이 유사도 계산 → **Instance Similarity(q) 분포**



특정 Strong Augmented 데이터와 i 번째 Labeled 데이터 사이 유사도

4. SimMatch



❖ **Labeled 손실 함수 + Unlabeled 손실함수 + Instance Similarity 손실함수**

- Instance Similarity 손실 함수는 유사도 분포 차이를 최소화
- 세 손실함수를 합하여 전체 손실 함수 정의 후 모델 학습

Labeled 데이터 → Model → p^x (Probability(Class Center)), z^x (Low embedding vector)

$$Loss_{labeled} = \frac{1}{B}\sum_{b=1}^{B} CE(y, p^x)$$

Unlabeled 데이터 → Model → p^w, z^w

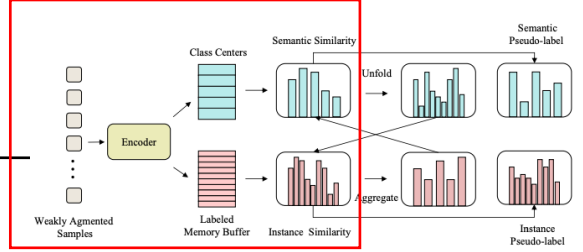
$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}(\max(DA(p^w)) > \tau)\, CE(DA(p^w), p^s)$$

역할: (Weak↔Labeled) 분포 == (Strong↔Labeled) 분포

Model → p^s, z^s

$$Loss_{instance} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} CE(q^w, q^s)$$

4. SimMatch



❖ **Weakly Augmented 데이터 두 Similarity에 레이블 정보를 넣어보자!**

- 두 유사도를 산출할 때 Label 정보가 사용되지 않음(단순한 모델 출력 값)
- Label 정보를 사용해 p^w, q^w 를 보정하여 더 정확한 **Semantic**/ **Instance** Similarity 산출해보자!

Labeled 데이터

$$Loss_{labeled} = \frac{1}{B}\sum_{b=1}^{B} CE(y, p^x)$$

Model

p^x Probability(Class Center)

z^x Low embedding vector

Unabeled 데이터

$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}(\max(DA(p^w)) > \tau)\, CE(DA(p^w), p^s)$$

$$Loss_{instance} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} CE(q^w, q^s)$$

Model

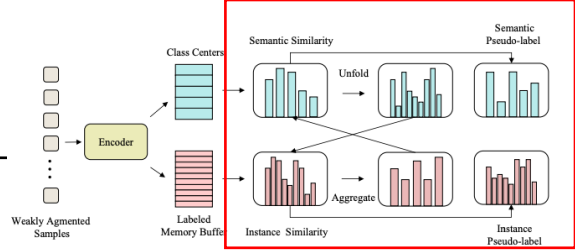
p^w

z^w

p^s

z^s

4. SimMatch



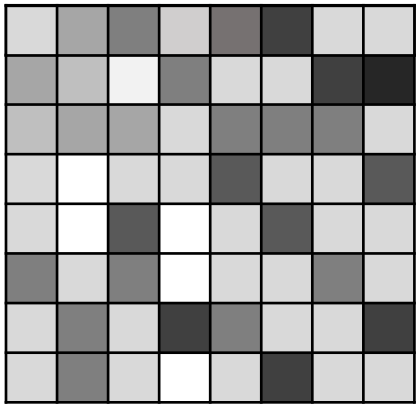
❖ **Weakly Augmented 데이터에서 산출한 두 Similarity에 레이블 정보를 넣어보자!**

- **Label Memory Buffer**는 모든 Label 값 자체가 포함된 벡터(Labeled 데이터 개수는 8개)

Label Memory Buffer

1
2
3
4
1
1
2
4

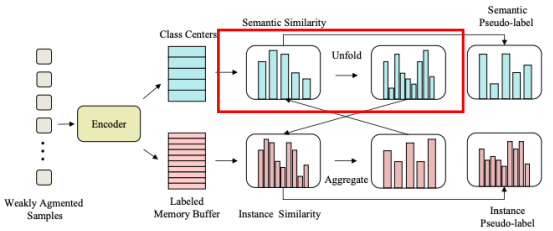
Memory Buffer(MB)



4. SimMatch

p: *Semantic similarity*

q: *Instance similartiy*



❖ Label 정보를 활용해 Instance Similarity 보정(Unfolding)

- **Label Memory Buffer**는 모든 Label 값 자체가 포함된 벡터(Labeled 데이터 개수는 8개)
- Weakly Augmented 데이터에 대한 Instance similarity 분포를 보정

Weakly Augmented 데이터 예측 확률(Semantic Similarity)

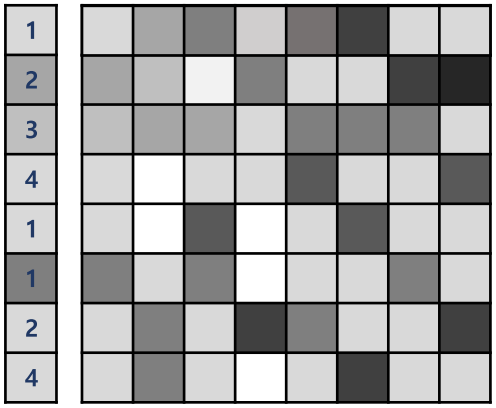
범주	Saxophone	Trumpet	Trombone	Tuba
p_i^w	0.8	0.01	0.1	0.09

Weakly Augmented 데이터와 MB Vector(j=1,2,….,8) 사이 Similarity 분포

MB	1	2	3	4	5	6	7	8
범주	1	2	3	4	1	1	2	4
q_j^w	0.2	0.05	0.01	0.04	0.3	0.25	0.05	0.01

p_i^{unfold}	0.8	0.01	0.1	0.09	0.8	0.8	0.01	0.09

Memory Buffer(MB)



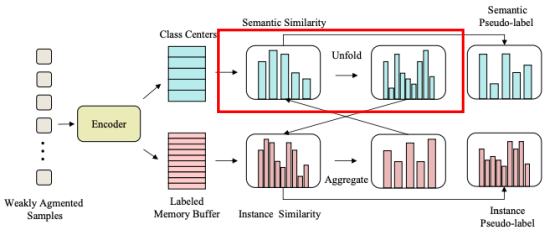
i번째 Labeled 데이터의 실제 범주
Weakly Augmented 데이터 예측 확률

$$p_i^{unfold} = p_j^w, where\ class(q_j^w) = class(p_i^w)$$

4. SimMatch

p: *Semantic similarity*

q: *Instance similartiy*



❖ Label 정보를 활용해 Instance Similarity 보정(Unfolding)

- **Label Memory Buffer**는 모든 Label 값 자체가 포함된 벡터(Labeled 데이터 개수는 8개)
- Weakly Augmented 데이터에 대한 Instance similarity 분포를 보정

Weakly Augmented 데이터 예측 확률(Semantic Similarity)

범주	Saxophone	Trumpet	Trombone	Tuba
p_i^w	0.8	0.01	0.1	0.09

Weakly Augmented 데이터와 MB Vector(j=1,2,….,8) 사이 Similarity 분포

MB	1	2	3	4	5	6	7	8
범주	1	2	3	4	1	1	2	4
q_j^w	0.2	0.05	0.01	0.04	0.3	0.25	0.05	0.01
$\hat{q}_i$	0.26	0.00	0.00	0.01	0.40	0.33	0.00	0.00

p_i^{unfold}	0.8	0.01	0.1	0.09	0.8	0.8	0.01	0.09

범주가 일치하는 경우, 유사도 ↑

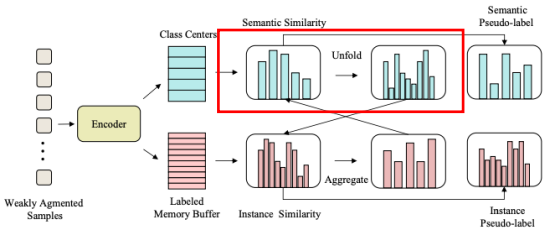
범주가 일치하지 않는 경우, 유사도 ↓

$$\hat{q}_i = \frac{q_i^w * p_i^{unfold}}{\sum_{k=1}^{K} q_k^w * p_k^{unfold}}$$

4. SimMatch

p: *Semantic similarity*

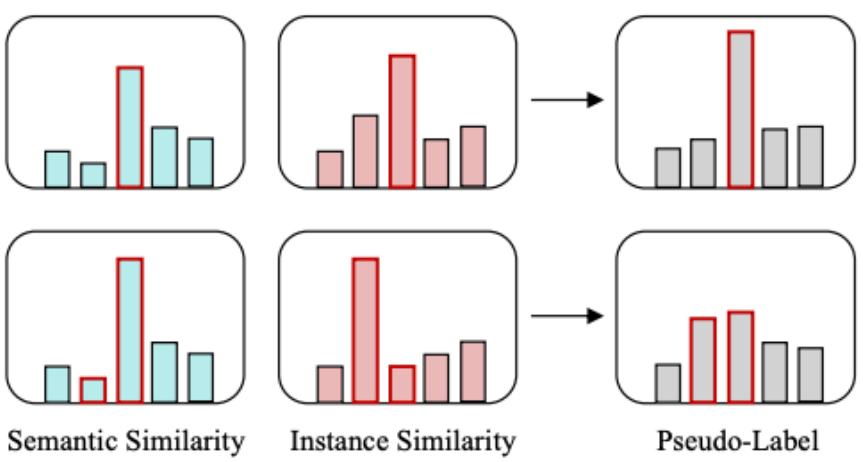
q: *Instance similartiy*



❖ Label 정보를 활용해 Instance Similarity 보정(Unfolding)

- **Label Memory Buffer**는 모든 Label 값 자체가 포함된 벡터(Labeled 데이터 개수는 8개)
- Weakly Augmented 데이터에 대한 Instance similarity 분포를 보정

Weakly Augmented 데이터 예측 확률



$\hat{q}_i$	0.26	0.00	0.00	0.01	0.40	0.33	0.00	0.00

Instance similarity 보정!

p_i^{unfold}	0.8	0.01	0.1	0.09	0.8	0.8	0.01	0.09

범주가 일치하는 경우, 유사도 ↑

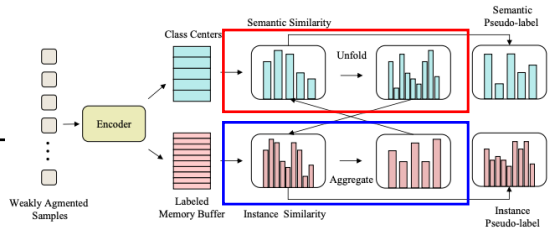
범주가 일치하지 않는 경우, 유사도 ↓

$$\hat{q}_i = \frac{q_i^w * p_i^{unfold}}{\sum_{k=1}^{K} q_k^w * p_k^{unfold}}$$

4. SimMatch

p: *Semantic similarity*

q: *Instance similartiy*



❖ 보정된 Instance Similarity를 사용해 Semantic Similarity 보정(Aggregation)

- 보정된 Instance similarity($\widehat{q_i^w}$)를 바탕으로 Semantic Similarity 보정 진행
  - ➢ 보정된 유사도 중 특정 범주에 속하는 유사도 합(q_i^{agg})을 산출하여 기존 Semantic Similarity와 가중합

Weakly Augmented 데이터 예측 확률

범주	Saxophone	Trumpet	Trombone	Tuba
p_i^w	0.8	0.01	0.1	0.09

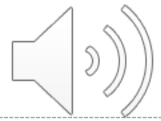
범주	Saxophone	Trumpet	Trombone	Tuba
q_i^{agg}	**0.99**	0.00	**0.00**	**0.01**

Weakly Augmented 데이터와 MB Vector(j=1,2,….,8) 사이 Instance Similarity

MB	1	2	3	4	5	6	7	8
범주	1	2	3	4	1	1	2	4
q_j^w	0.2	0.05	0.01	0.04	0.3	0.25	0.05	0.01
$\widehat{q_i^w}$	**0.26**	**0.00**	**0.00**	**0.01**	**0.40**	**0.33**	**0.00**	**0.00**

$$q_i^{agg} = \sum_{j=1}^{K} \mathbb{I}\left(class(p_i^w) = class(q_i^w)\right)\widehat{q_j^w}$$

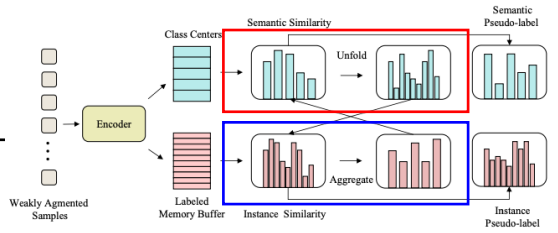
p_i^{unfold}	0.8	0.01	0.1	0.09	0.8	0.8	0.01	0.09



4. SimMatch

p: *Semantic similarity*

q: *Instance similartiy*



❖ 보정된 Instance Similarity를 사용해 **Semantic Similarity 보정(Aggregation)**

- 보정된 Instance similarity($\widehat{q_i^w}$)를 바탕으로 Semantic Similarity 보정 진행
  - ➢ 보정된 유사도 중 특정 범주에 속하는 유사도 합(q_i^{agg})을 산출하여 기존 Semantic Similarity와 가중합

Weakly Augmented 데이터 예측 확률

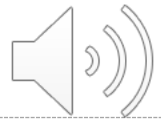
범주	Saxophone	Trumpet	Trombone	Tuba
p_i^w	0.8	0.01	0.1	0.09

범주	Saxophone	Trumpet	Trombone	Tuba
q_i^{agg}	**0.99**	0.00	0.00	0.01

$\alpha = 0.9$ $1 - \alpha = 0.1$

범주	Saxophone	Trumpet	Trombone	Tuba
$\widehat{p_i^w}$	0.82	0.01	0.09	0.08

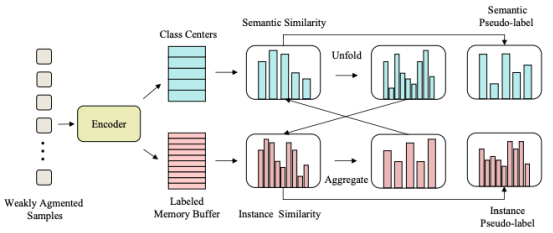
Sematic similarity 보정!



4. SimMatch

p: *Semantic similarity*

q: *Instance similartiy*



❖ SimMatch 학습을 위한 최종 손실 함수 정의

- **Unfolding**, **Aggregation** 된 값을 사용해 Similarity 손실 함수 산출
- 즉, 레이블 데이터 정보를 반영해 Similarity 보정한 후 손실 함수 산출

Labeled 데이터

$$Loss_{labeled} = \frac{1}{B}\sum_{b=1}^{B} CE(y, p^x)$$

Model

p^x

Class center(Probability)

z^x

Low embedding vector

$$Loss_{unlabeled} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{I}\left(\max\left(\widehat{DA(p^w)}\right) > \tau\right) CE\left(\widehat{DA(p^w)}, p^s\right)$$

Unabeled 데이터

Model

p^w

z^w

$$Loss_{instance} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} CE\left(\widehat{q^w}, q^s\right)$$

Model

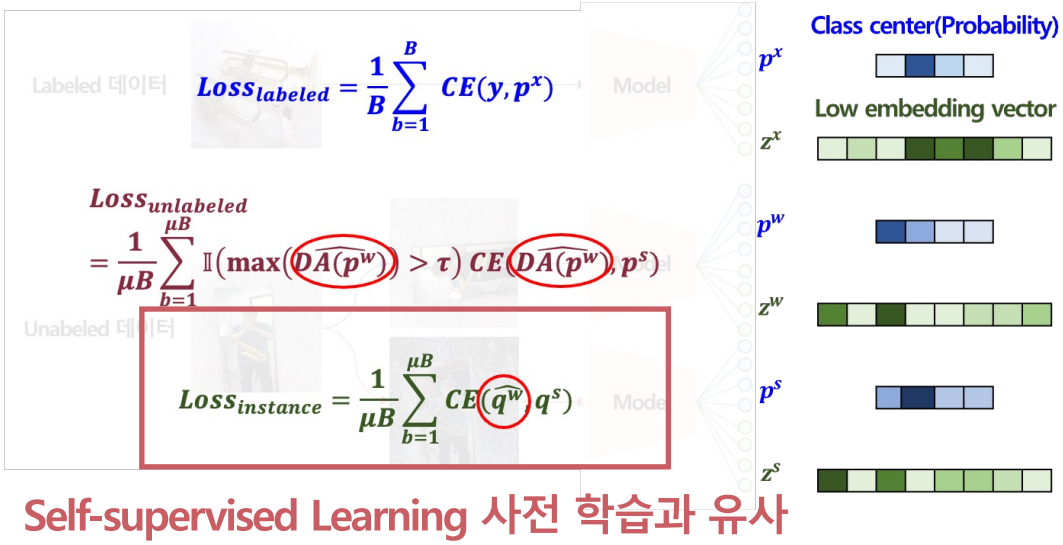
p^s

z^s

4. SimMatch

❖ **FixMatch vs SimMatch(FixMatch 에서 추가된 내용)**

- Unlabeled 손실 함수 산출 시, Distribution Alignment 후 p^w 산출
- Instance 관련 손실 함수 추가 (Semantic & Instance Similarity)
  - ➢ 기존 Consistency Regularization에 관측치 별 특징(Similarity)이 유사해지도록 학습
- Label 정보를 활용해 Weakly Augmented 데이터로 부터 생성하는 Pseudo Label 보정



Self-supervised Learning 사전 학습과 유사

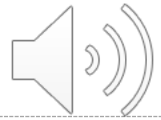
4. SimMatch

❖ SimMatch 실험 결과 – 과거 방법론과 비교(CIFAR-10/ CIFAR-100)

- 과거 Semi-supervised 및 FixMatch 등과 비교 진행
- 학습 데이터 수가 적을 때(40 Labels, 400 Labels)에서 특히 성능 향상이 두드러짐

Table 1. Top-1 Accuracy comparison (mean and std over 5 runs) on CIFAR-10 and CIFAR-100 with varying labeled set sizes.

	CIFAR-10			CIFAR-100		
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels
Π-Model [34]	-	45.74±3.97	58.99±0.38	-	42.75±0.48	62.12±0.11
Pseudo-Labeling [35]	-	50.22±0.43	83.91±0.28	-	42.62±0.46	63.79±0.19
Mean Teacher [52]	-	67.68±2.30	90.81±0.19	-	46.09±0.57	64.17±0.24
UDA [56]	70.95±5.93	91.18±1.08	95.12±0.18	40.72±0.88	66.87±0.22	75.50±0.25
MixMatch [6]	52.46±11.50	88.95±0.86	93.58±0.10	32.39±1.32	60.06±0.37	71.69±0.33
ReMixMatch [5]	80.90±9.64	94.56±0.05	95.28±0.13	55.72±2.06	72.57±0.31	76.97±0.56
FixMatch(RA) [48]	86.19±3.37	94.93±0.65	95.74±0.05	51.15±1.75	71.71±0.11	77.40±0.12
Dash [57]	86.78±3.75	**95.44**±0.13	95.92±0.06	55.24±0.96	72.82±0.21	78.03±0.14
CoMatch [36]	93.09±1.39	95.09±0.33	-	-	-	-
SimMatch(Ours)	**94.40**±1.37	95.16±0.39	**96.04**±0.01	**62.19**±2.21	**74.93**±0.32	**79.42**±0.11



4. SimMatch

❖ **SimMatch 실험 결과 – 과거 방법론과 비교(ImageNet 1%, 10% 데이터)**

- 적은 학습 Epoch 및 사전 학습 없이 최고 성능을 달성
- (MoCo V2 + CoMatch) 방식으로 많은 학습으로 Epoch 유사한 성능 달성

Table 2. Experimental results on ImageNet with 1% and 10% labeled examples.

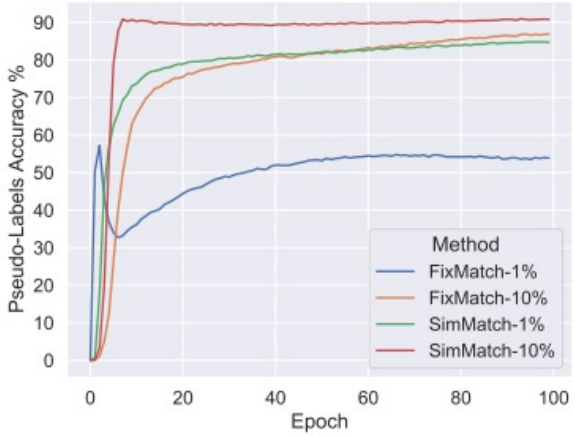
Self-supervised Pre-training	Method	Epochs	Paramters (train/test)	Top-1 Label fraction 1%	Top-1 Label fraction 10%	Top-5 Label fraction 1%	Top-5 Label fraction 10%
None	Pseudo-label [35, 62]	~100	25.6M / 25.6M	-	-	51.6	82.4
	VAT+EntMin. [24, 41, 62]	-	25.6M / 25.6M	-	68.8	-	88.5
	S4L-Rotation [62]	~200	25.6M / 25.6M	-	53.4	-	83.8
	UDA [56]	-	25.6M / 25.6M	-	68.8	-	88.5
	FixMatch [48]	~300	25.6M / 25.6M	-	71.5	-	89.1
	CoMatch [36]	~400	30.0M / 25.6M	66.0	73.6	86.4	**91.6**
PCL [37]	Fine-tune	~200	25.8M / 25.6M	-	-	75.3	85.6
SimCLR [14]		~1000	30.0M / 25.6M	48.3	65.6	75.5	87.8
SimCLR V2 [15]		~800	34.2M / 25.6M	57.9	68.4	82.5	89.2
BYOL [25]		~1000	37.1M / 25.6M	53.2	68.8	78.4	89.0
SwAV [10]		~800	30.4M / 25.6M	53.9	70.2	78.5	89.9
WCL [65]		~800	34.2M / 25.6M	65.0	72.0	86.3	91.2
MoCo V2 [16]	Fine-tune	~800	30.0M / 25.6M	49.8	66.1	77.2	87.9
	CoMatch [36]	~1200	30.0M / 25.6M	67.1	73.7	**87.1**	91.4
MoCo-EMAN [8]	FixMatch-EMAN [8]	~1100	30.0M / 25.6M	63.0	74.0	83.4	90.9
None	**SimMatch (Ours)**	~400	30.0M / 25.6M	**67.2**	**74.4**	**87.1**	**91.6**

4. SimMatch

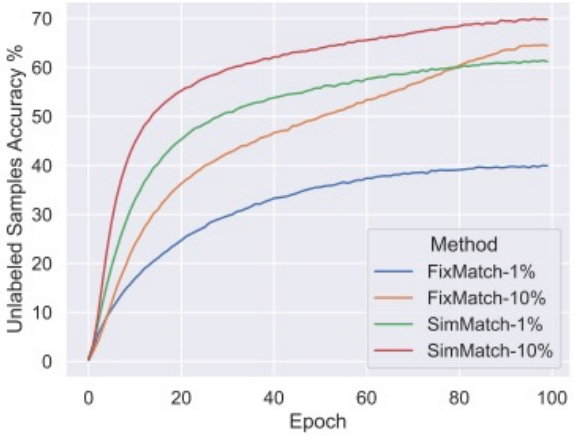
❖ FixMatch vs SimMatch

- ImageNet 학습 데이터 중 1%, 10%만 사용해 학습한 경우에 대한 세 가지 성능 표기
  - ➢ Pseudo-Labels: Weak Augmented 데이터에 대한 Pseudo-Label 이 얼마나 정확한지를 나타내는 그림
  - ➢ Unlabeled Samples: Weak/ Strong Augmented 데이터에 예측 범주와 실제 범주 사이 정확도 시계열 그림
  - ➢ Validation: 검증 데이터 셋 정확도 시계열 그림

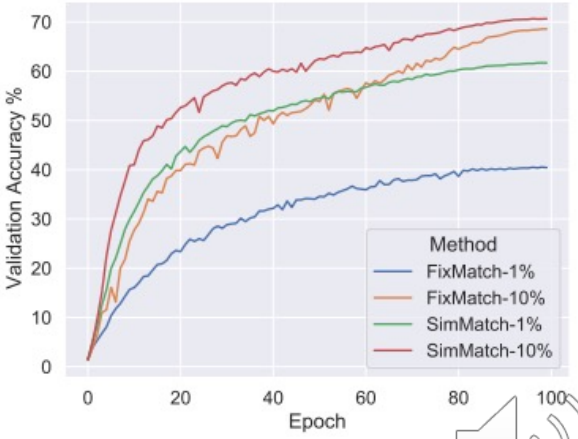
FixMatch 1% **FixMatch 10%** **SimMatch 1%** **SimMatch 10%**



(a) Pseudo-Labels Accuracy

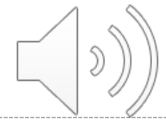


(b) Unlabeled Samples Accuracy



(c) Validation Accuracy

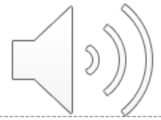
목차



5. Conclusion

❖ Summary

- 실제 현장에서는 (입력-출력) 데이터를 수집하기 어려움
- 이를 위해 Self-supervised/ Semi-supervised Learning 연구가 이루어지는 상황
- 특히 이번 세미나에서는 Semi-supervised Learning에 집중
  - ➢ FixMatch: Unlabeled 데이터에 Weak/ Strong Augmentation을 진행해 예측 값이 유사해지도록 학습
  - ➢ SelfMatch: Self-supervised Learning 방식으로 모델 사전 학습 후 FixMatch 학습
  - ➢ SimMatch: Semantic & Instance similarity 정의 후 이들을 사용해 Unlabeled 데이터 활용
  - ➢ SimMatch는 Semi&Self supervised Learning 방법론 개념을 통합했다는 판단



Reference

1. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., ... & Li, C. L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems, 33, 596-608.

2. Kim, B., Choo, J., Kwon, Y. D., Joe, S., Min, S., & Gwon, Y. (2021). Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning. arXiv preprint arXiv:2101.06480.

3. Zheng, M., You, S., Huang, L., Wang, F., Qian, C., & Xu, C. (2022). Simmatch: Semi-supervised learning with similarity matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14471-14481).

4. 2022년 2학기 강필성 교수님 강의자료(Semi-supervised Learning: Holistic Methods)

5. Li, J., Xiong, C., & Hoi, S. C. (2021). Comatch: Semi-supervised learning with contrastive graph regularization. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9475-9484).

6. SimMatch Github(https://github.com/kyle-1997/simmatch)

감사합니다