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

NeurIPS 2020, 176회 인용

Park, MinKyu

2021.03.12

Dongguk University

Artificial Intelligence Laboratory

* 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

mkpark73@dongguk.edu

Semi-Supervised Learning

- Using labelled as well as unlabelled data to perform certain learning tasks

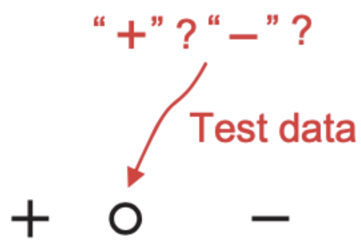
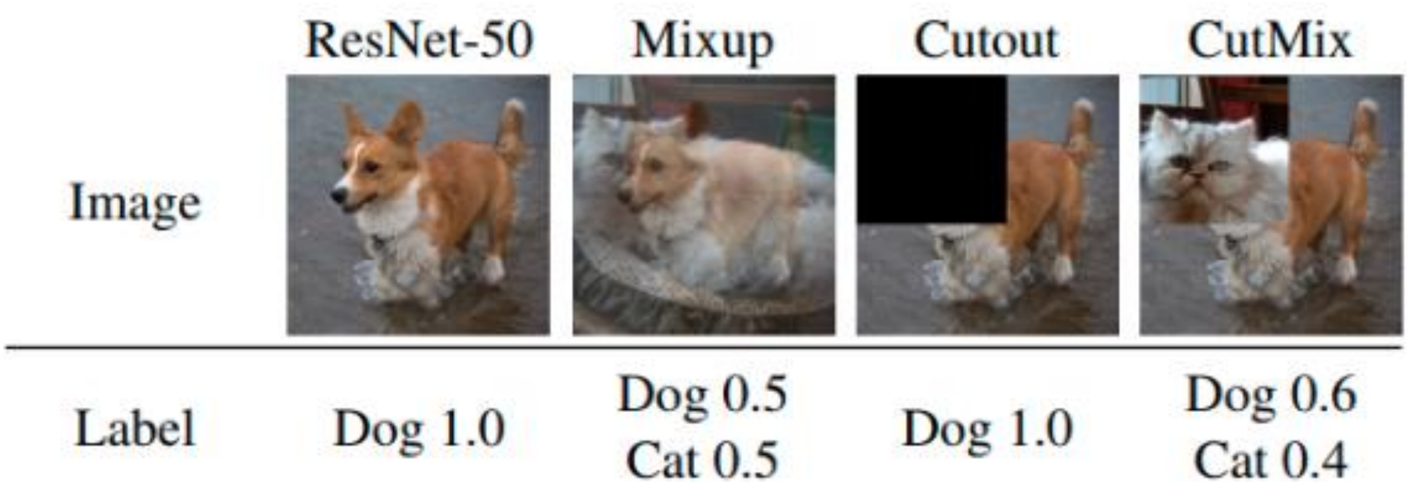


Figure 3. Illustration of the usefulness of unlabeled data.

Data Augmentation

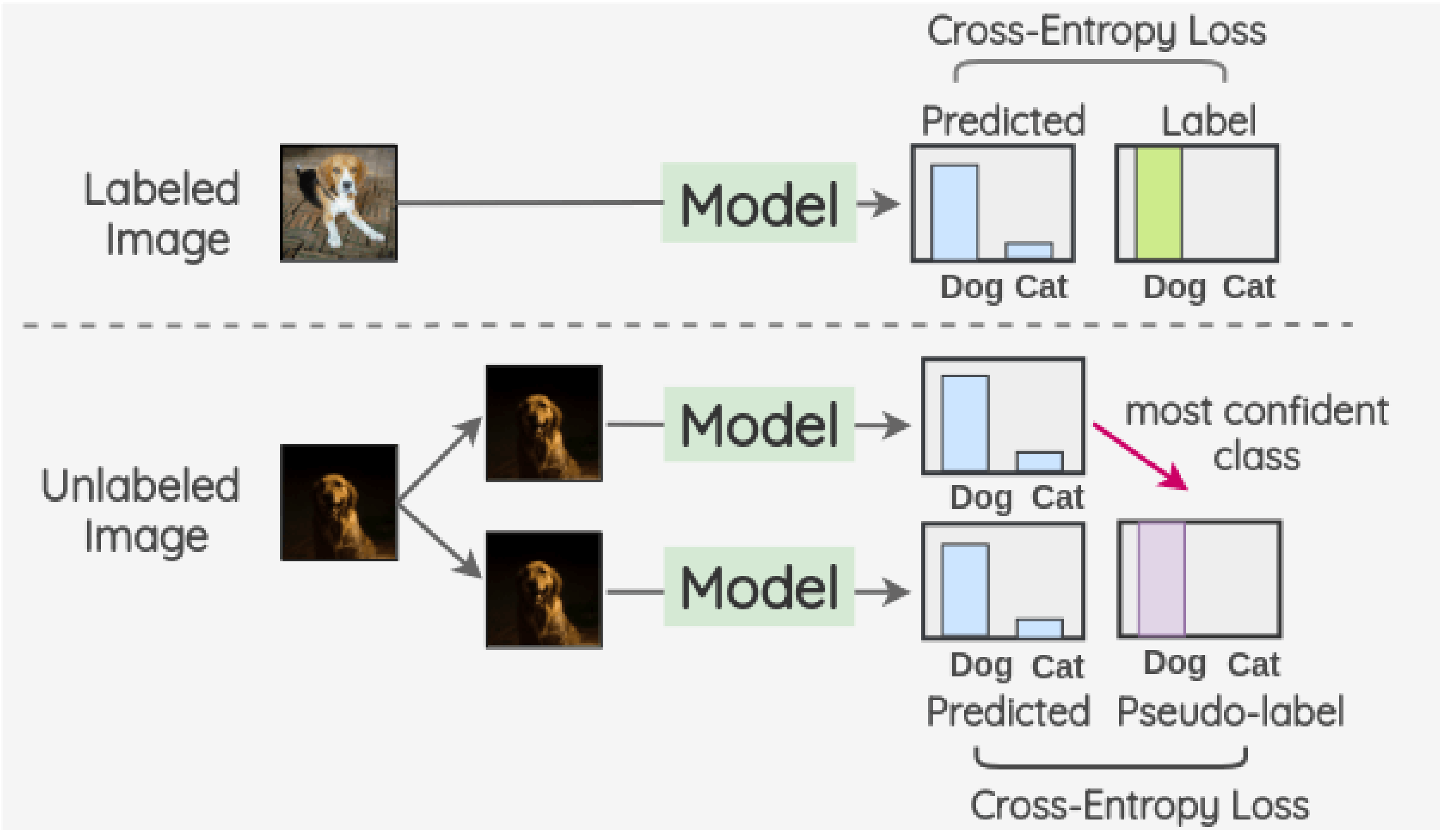
- **Data augmentation** significantly increases the diversity of data available for training our models, without actually collecting new data samples.
- Simple image data augmentation techniques like flipping, random crop, and random rotation are commonly used to train large models.

Overview of the results of Mixup, Cutout, and CutMix.



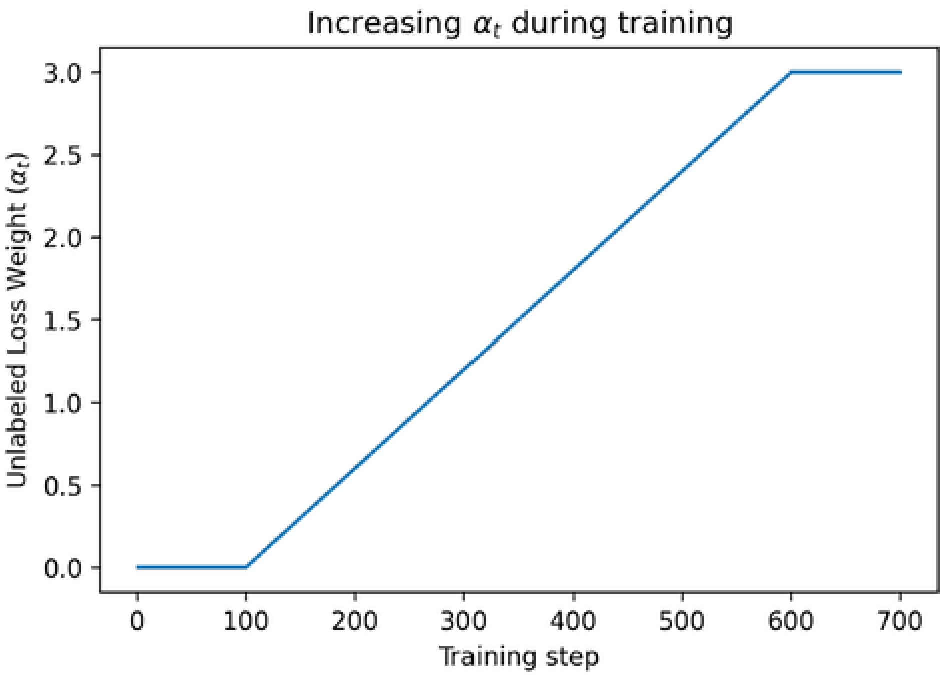
Self-Training

- In this semi-supervised formulation,
  - a model is trained on labeled data and used to predict pseudo-labels for the unlabeled data.
  - The model is then trained on both ground truth labels and pseudo-labels simultaneously.
- a. Pseudo-label
  - Dong-Hyun Lee proposed a very simple and efficient formulation called “Pseudo-label” in 2013.
  - The idea is to train a model **simultaneously** on a batch of both labeled and unlabeled images.

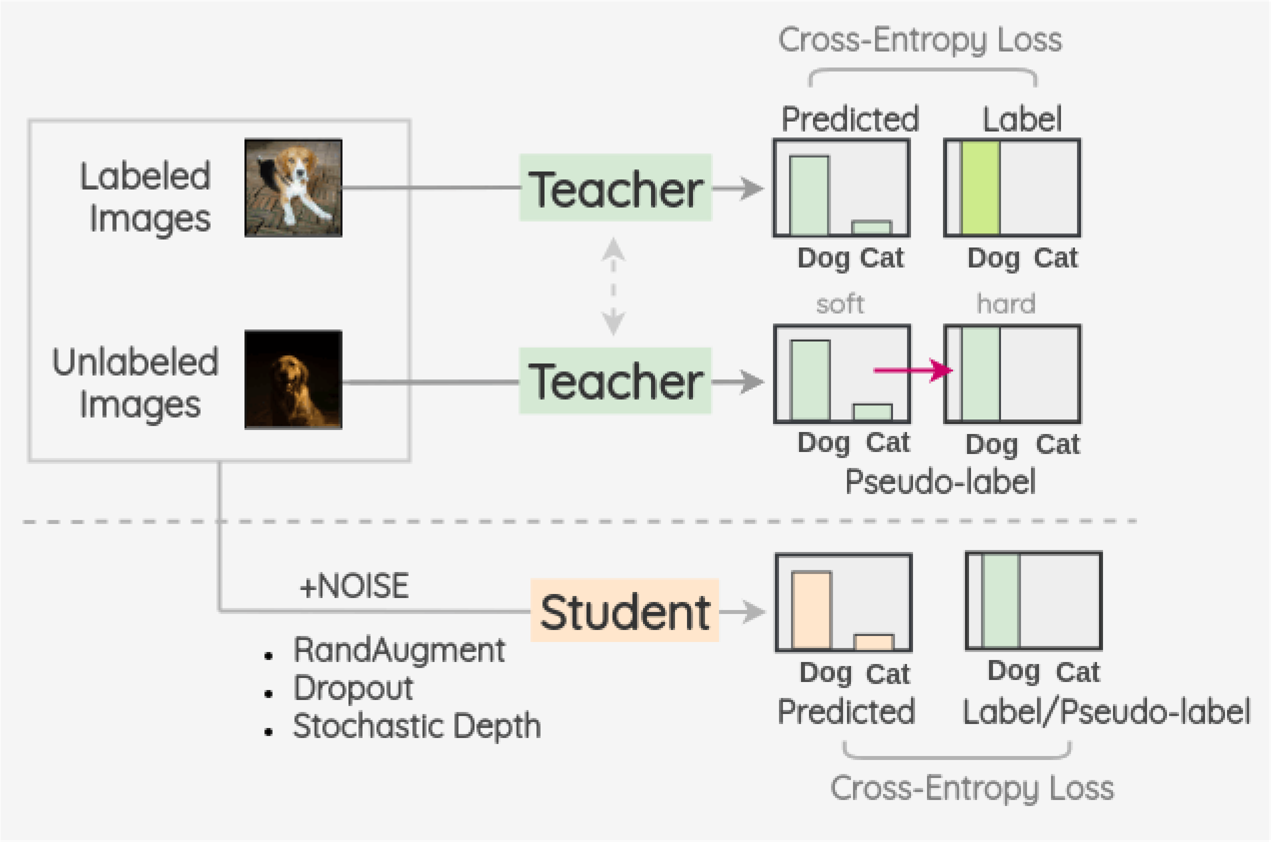

Cross-Entropy Loss
Predicted
Label
Labeled Image
Model
Dog Cat
Dog Cat
Unlabeled Image
Model
Model
most confident class
Dog Cat
Dog Cat
Dog Cat
Predicted
Pseudo-label
Cross-Entropy Loss

- The total loss is a weighted sum of the labeled and unlabeled loss terms.

$$L = L_{labeled} + \alpha_t * L_{unlabeled}$$

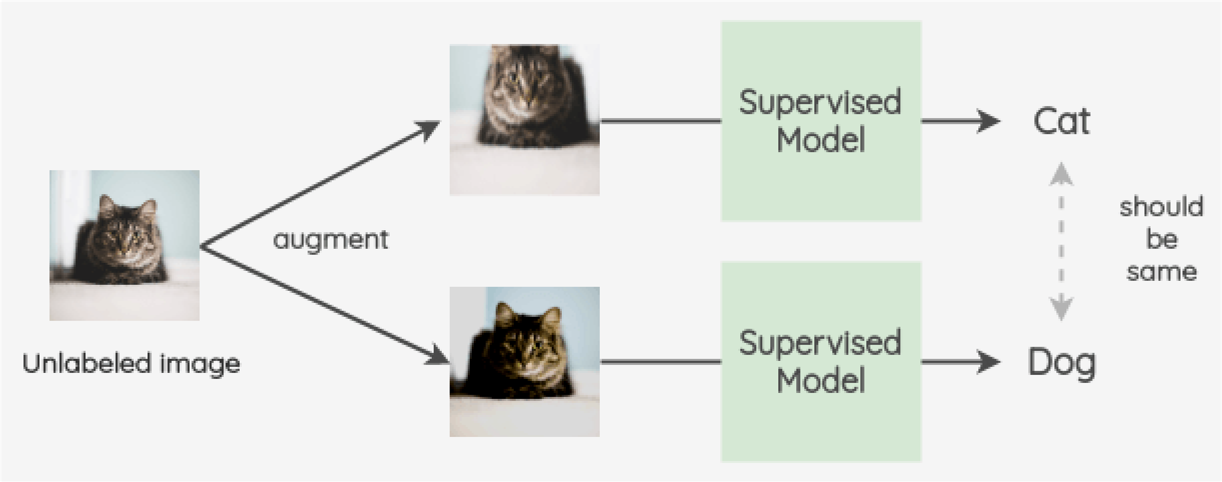


- b. Noisy Student
  - Xie et al. proposed a semi-supervised method inspired by Knowledge Distillation called “Noisy Student” in 2019.
  - The key idea is to train two separate models called “Teacher” and “Student”.

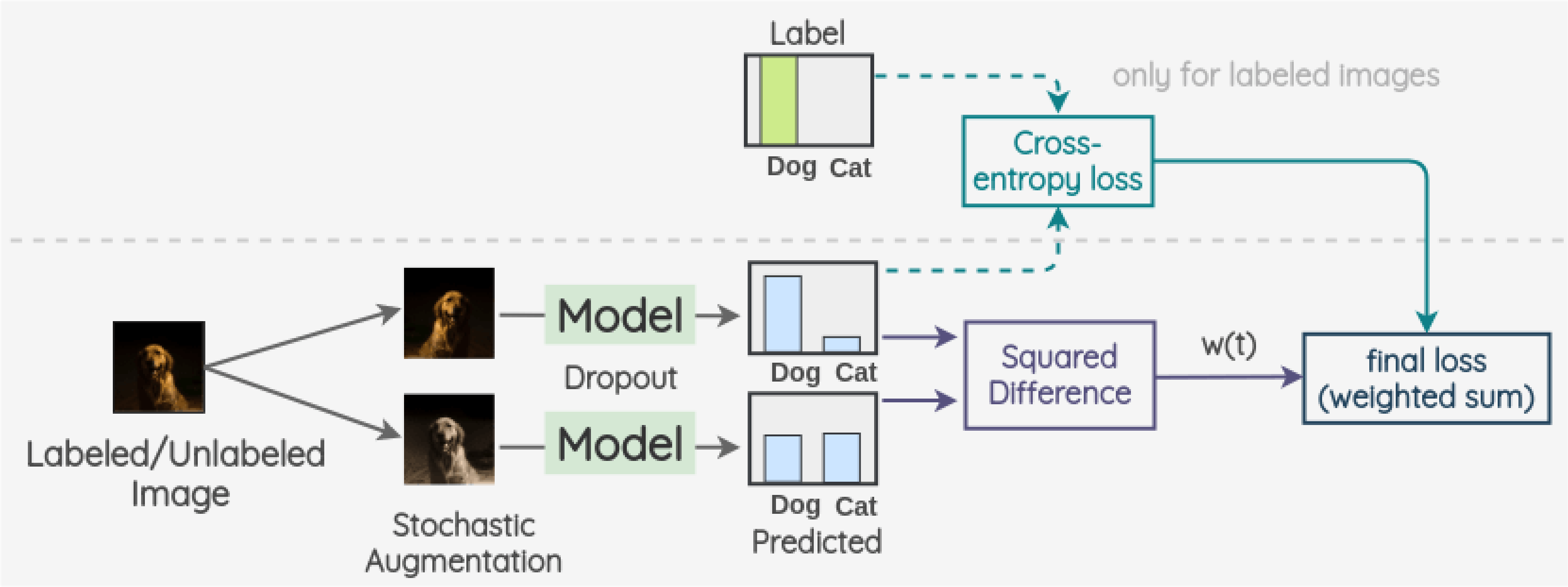


Consistency Regularization

- This paradigm uses the idea that model predictions on an unlabeled image should remain the same even after adding noise.

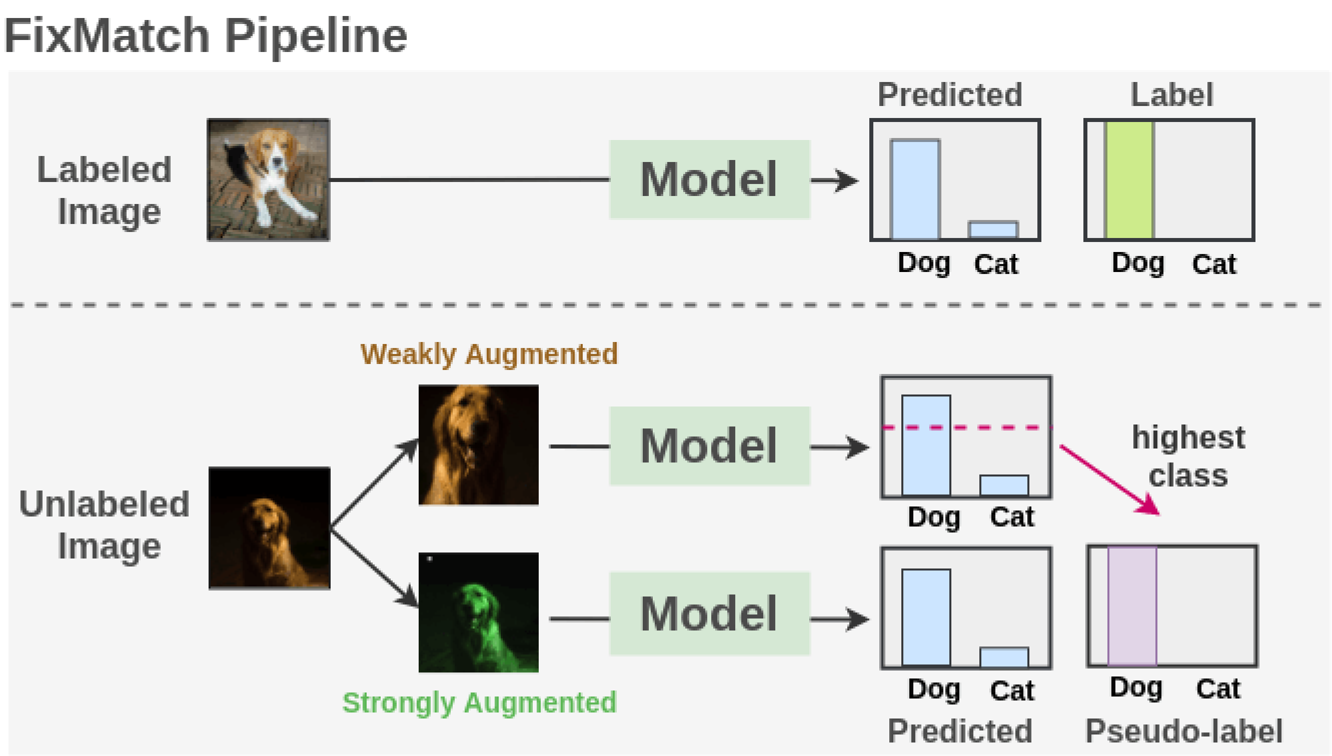


- π-model
  - This model was proposed by Laine et al. in a conference paper at ICLR 2017.
  - The key idea is to create two random augmentations of an image for both labeled and unlabeled data



FixMatch

- FixMatch borrows this idea from UDA and ReMixMatch to apply different augmentation i.e **weak augmentation on unlabeled image** for the pseudo-label generation and **strong augmentation on unlabeled image** for prediction.



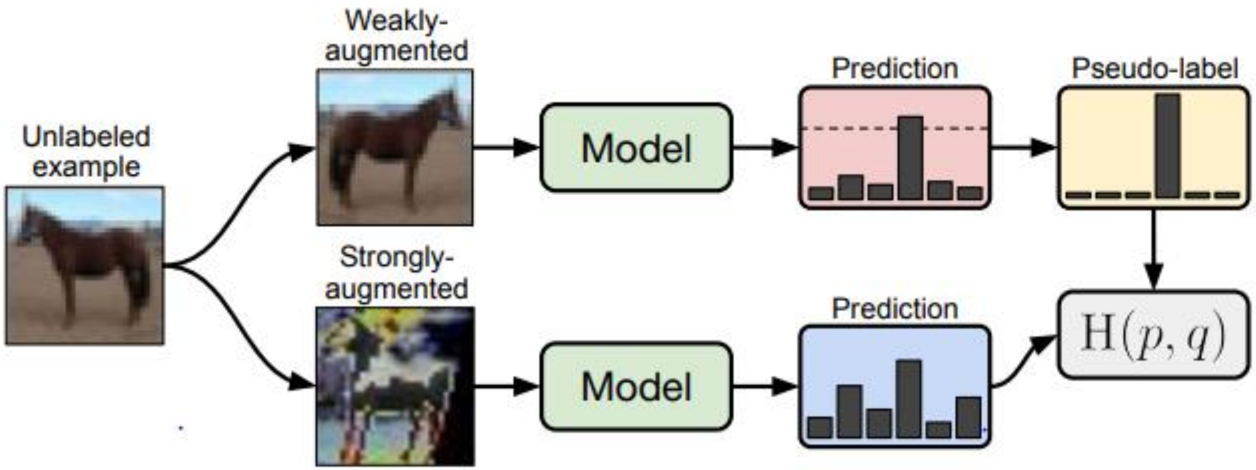
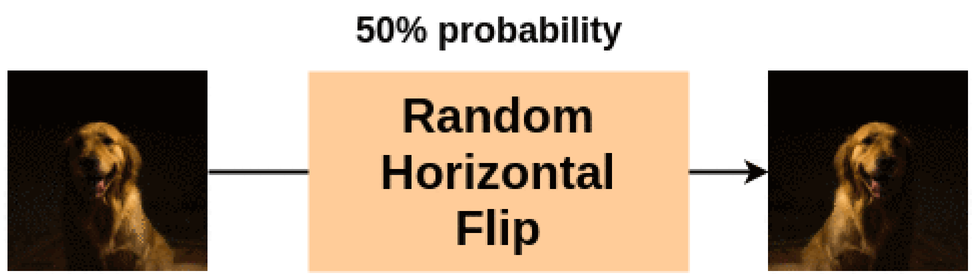
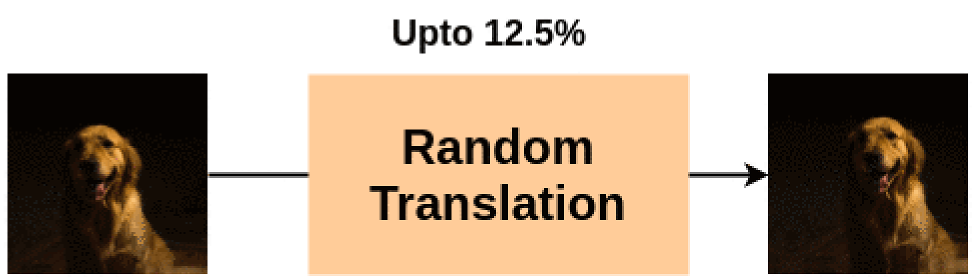


Figure 1: Diagram of FixMatch. A weakly-augmented image (top) is fed into the model to obtain predictions (red box). When the model assigns a probability to any class which is above a threshold (dotted line), the prediction is converted to a one-hot pseudo-label. Then, we compute the model's prediction for a strong augmentation of the same image (bottom). The model is trained to make its prediction on the strongly-augmented version match the pseudo-label via a cross-entropy loss.

- **1. Training Data and Augmentation**
  - **a. Weak Augmentation**
    - Random Horizontal Flip



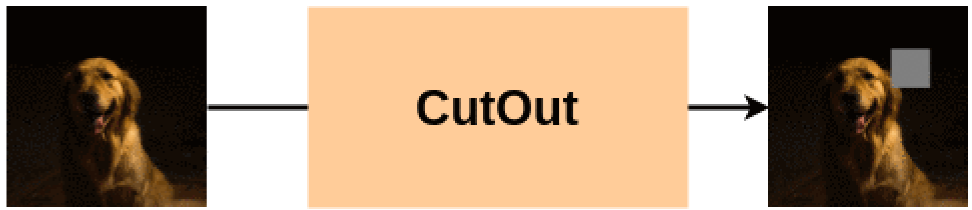
    - Random Vertical and Horizontal Translation



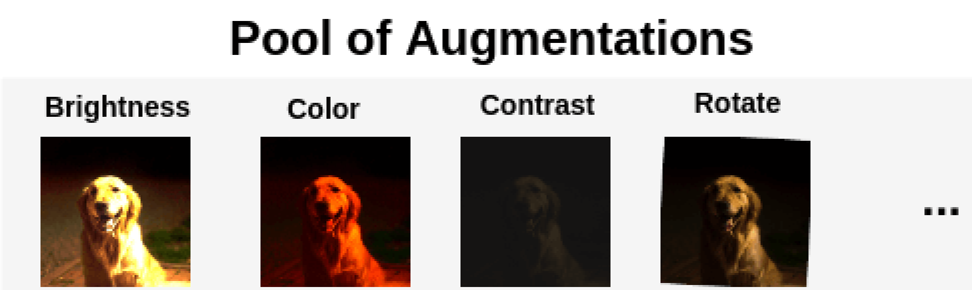
1. Training Data and Augmentation

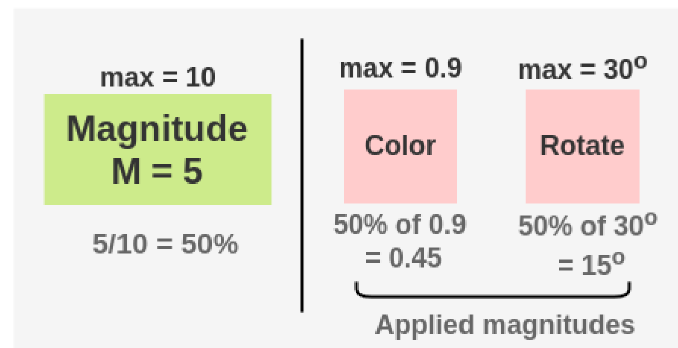
– Strong Augmentation

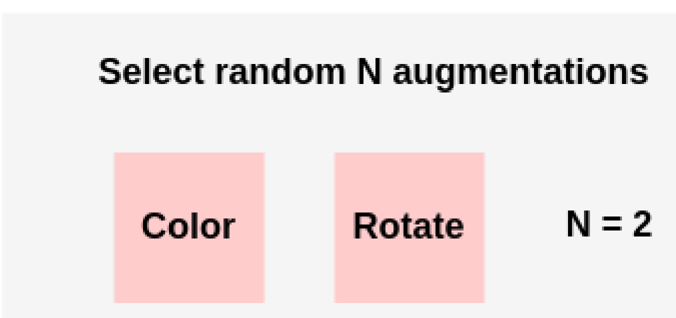
- 1. Cutout

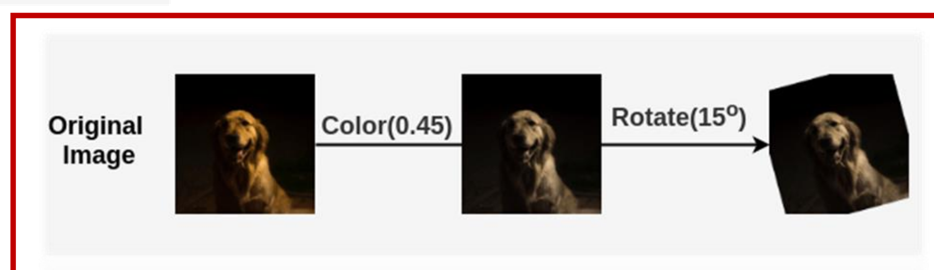


- 2. AutoAugment Variants

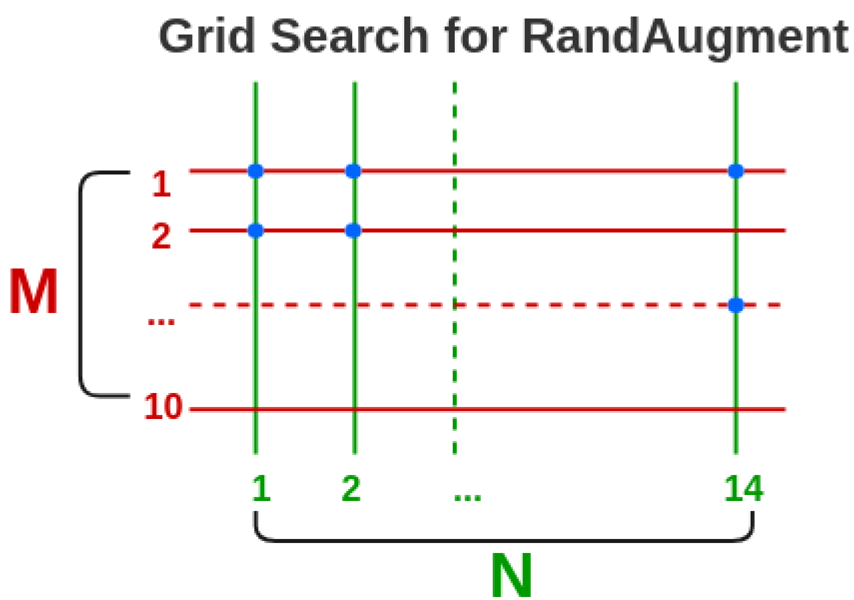




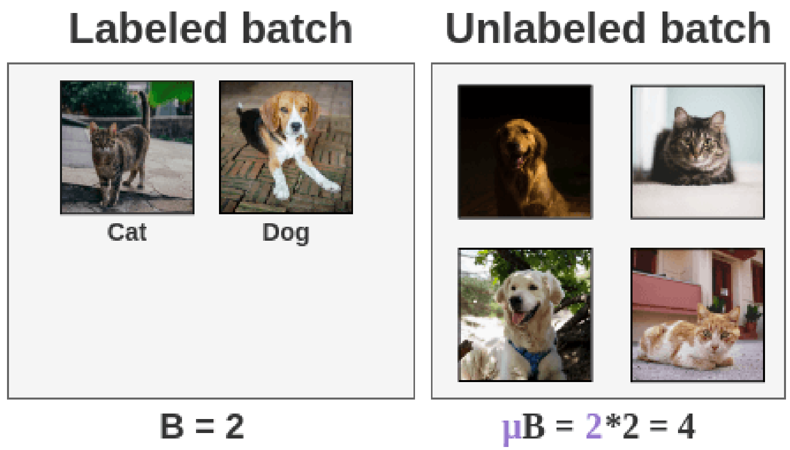




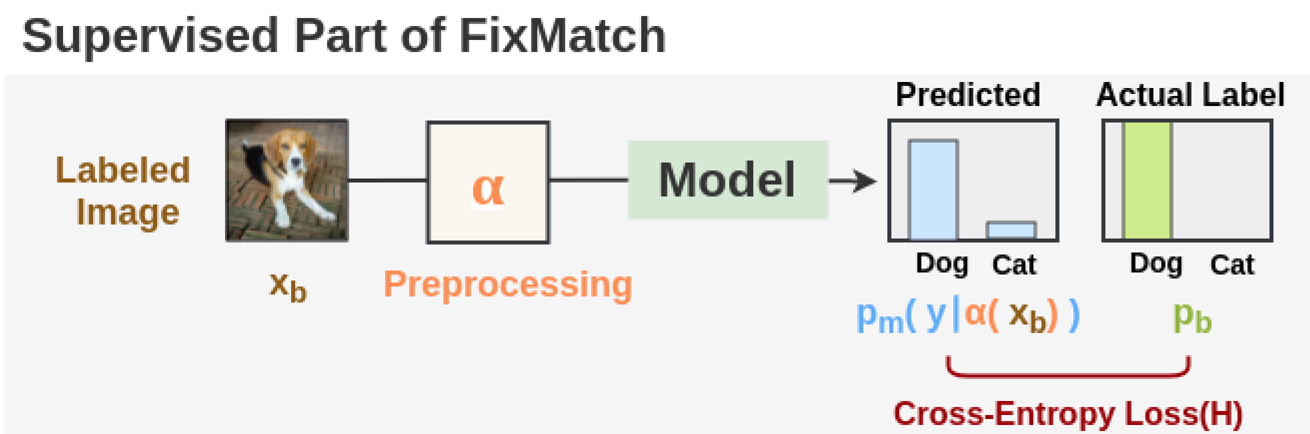
- The values of N and M can be found by hyper-parameter optimization on a validation set with a grid search.



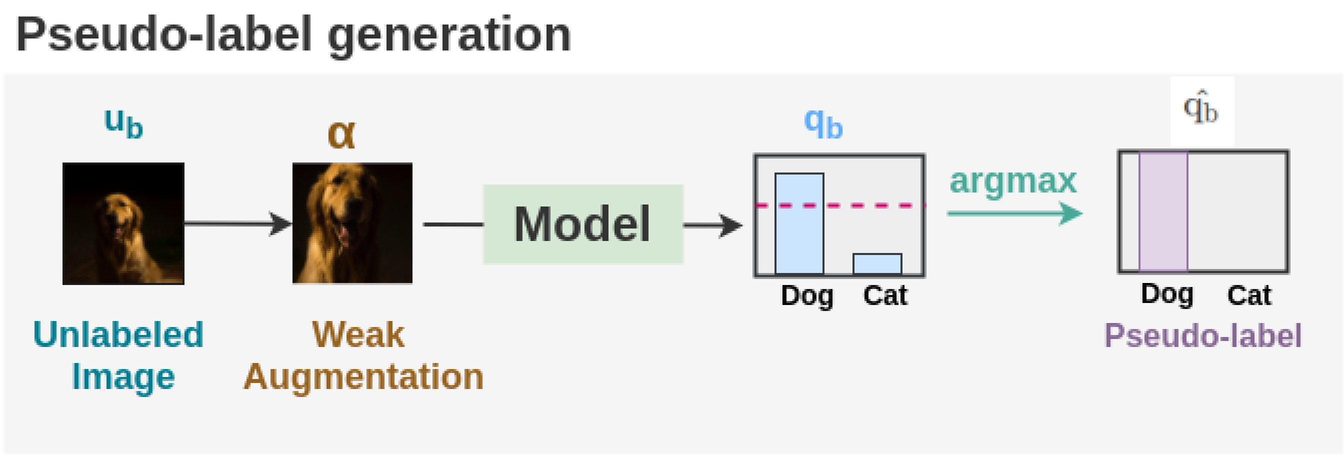
- 2. **Model Training and Loss Function**
  - **Step 1: Preparing batches**



  - **Step 2: Supervised Learning**



- **Step 3: Pseudolabeling**

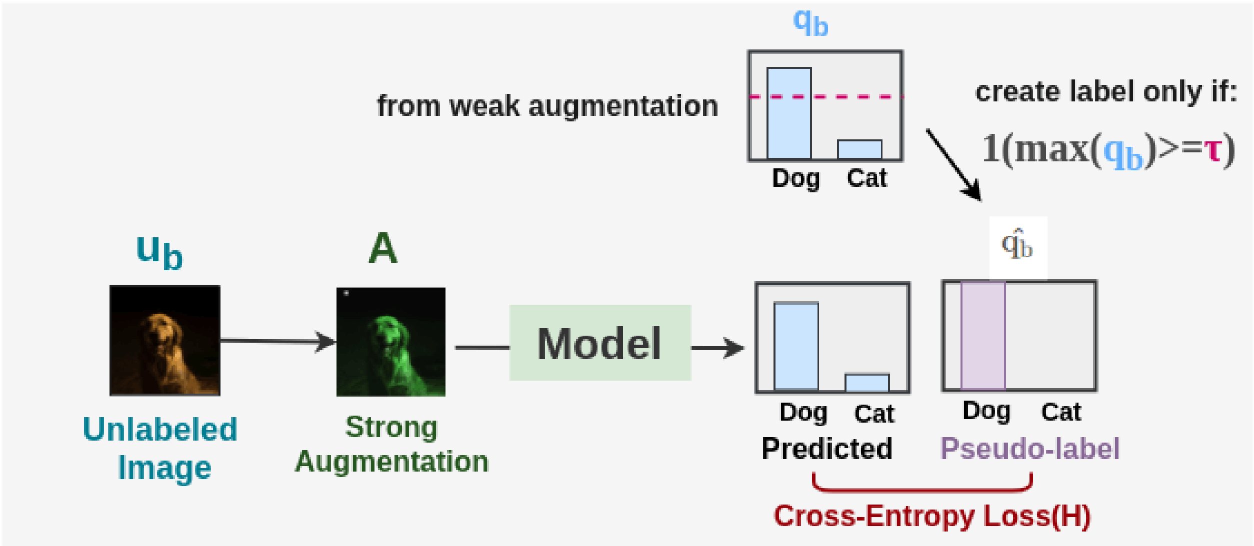


$$q_b = p_m(y|\alpha(u_b))$$

$$\hat{q}_b = argmax(q_b)$$

Step 4: Consistency Regularization

Consistency Regularization

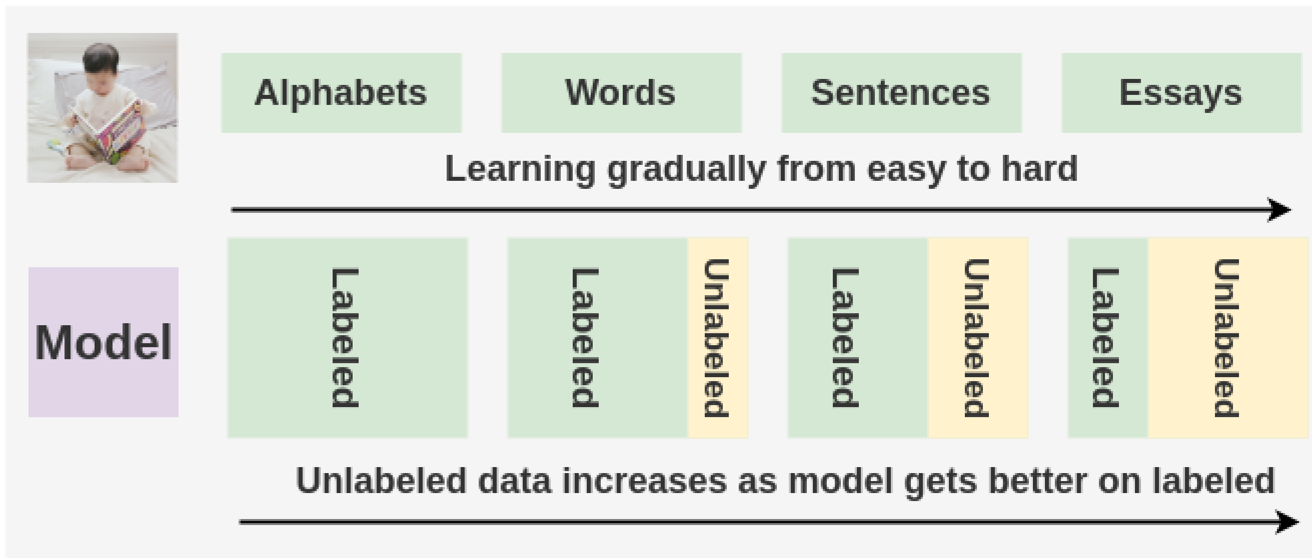


$$l_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} 1(max(q_b) >= \tau)\; H(\hat{q}_b, p_m(y|A(u_b)\;)$$

Step 5: Curriculum Learning

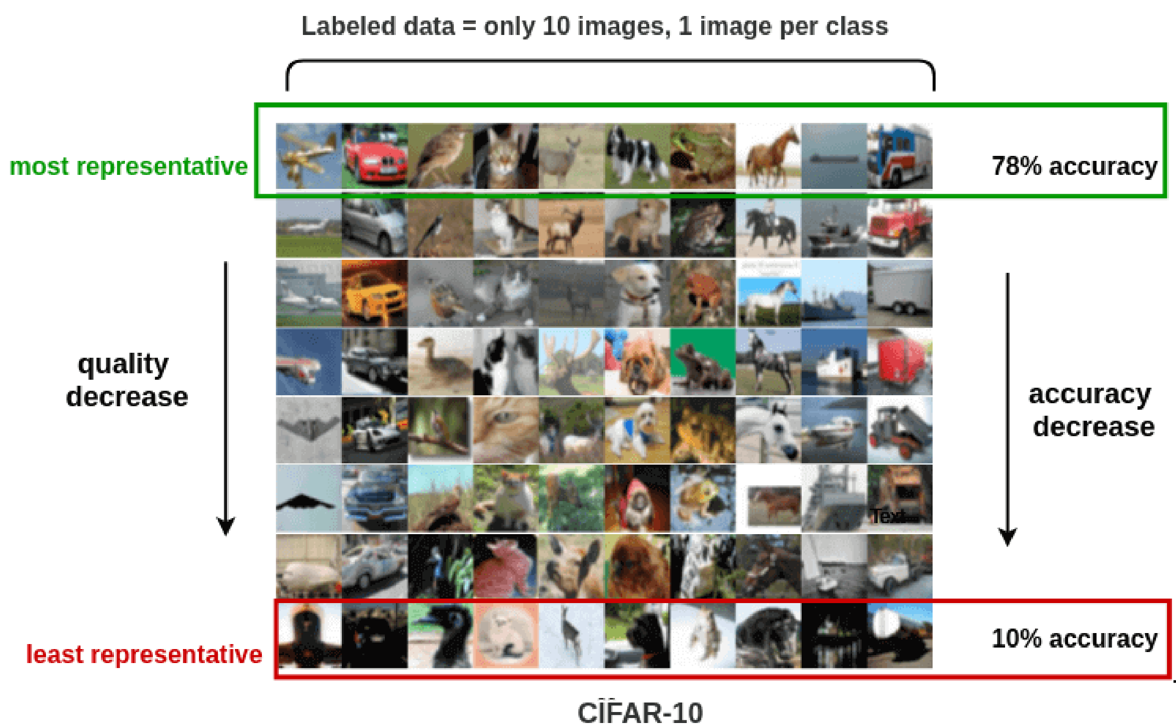
$$loss = l_s + \lambda_u l_u$$

Curriculum learning in FixMatch



Q. Can we learn with just one image per class?

- 8 training datasets with examples ranging from most representative to the least representative.
  - **Most representative bucket**: 78% median accuracy with a maximum accuracy of 84%
  - **Middle bucket**: 65% accuracy
  - **Outlier bucket**: Fails to converge completely with only 10% accuracy



Experiments

	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

Table 2: Error rates for CIFAR-10, CIFAR-100, SVHN and STL-10 on 5 different folds. FixMatch (RA) uses RandAugment [11] and FixMatch (CTA) uses CTAugment [3] for strong-augmentation. All baseline models (Π-Model [43], Pseudo-Labeling [25], Mean Teacher [51], MixMatch [4], UDA [54], and ReMixMatch [3]) are tested using the same codebase.

감사합니다.