FixMatch & Semi-supervised Learning Exploration

Adrián Campoy Rodríguez adriancr@kth.se
Gustavo Teodoro D. Beck gtdb@kth.se
Fernando García Sanz fegs@kth.se

October 28, 2020

Abstract

The amount of data available is virtually infinite, yet most of this data is unlabeled. The procedure of labeling data, normally, takes an enormous amount of time and must be reliable, since the nature of some data, like medical samples, can be extremely sensitive. Therefore, how to exploit the huge amount of available unlabeled data? Semi-supervised learning (SSL) methods, such as FixMatch [1], shown outstanding results. Thus, our project aimed at reproducing FixMatch, exploring the SSL domain, and providing new promising insights that may enhance the approach given to unlabeled data.

1 Introduction

Deep Neural Networks (DNNs) are very well known for its achievements in supervised learning. Image classification or diagnosing diseases using medical data, such as x-ray images or electrocardiograms, are tasks in which DNN has shown outstanding results [2]. However, it is not always the case that there is enough labeled data available for training a DNN or, even if it is labeled, there exists the possibility that labels are noisy which will affect the generalization capabilities of the network [3]. Moreover, having a labeled dataset implies that there has been human work which can be costly, for instance, in the example of medical data in which highly qualified professionals are required to provide labels [1]. As a consequence, there are many lines of research trying to leverage Semi-Supervised Learning (SSL) and unlabeled data, which are more abundant and cheaper to obtain, to better train DNNs [1].

Thereby, we focused on exploring the potential of FixMatch [1] and reproducing its results. Our implementation achieved very good results (90,7% of accuracy with 4000 labeled CIFAR-10 samples as can be observed in table 2), even though we simplified the training process by reducing the number of training steps of the data augmentation algorithm. The results and the source code of the methods discussed in this document can be found in https://github.com/fernando2393/FixMatch-Exploration.

2 Related Work

Additionally to traditional unsupervised learning methods, contrastive learning (semi-supervised) approaches received more attention in recent years. In essence, semi-supervised learning (SSL) methods consist of bolstering the performance of discriminative models by exploiting unlabeled data that assist the algorithms to learn the latent representations of the assigned dataset space. However, in this project, we propose a new perspective (5.3) that boosts SSL's performance by adding extra unlabeled images from different data distributions with some common features, while concurrently mapping the latent representation of the extra data distribution.

Primarily, we focused on reproducing FixMatch [1] and then, exploring the domain of pseudo-labeling and self-training techniques. Inherited from ReMixMatch [4], FixMatch implemented pseudo-labeling by generating a weakly-augmented example to assign artificial labels with high confidence (i.e. surpassing a threshold of 95% of confidence) and then consistently matching the predictions of strongly-augmented (using CTAugment 3.2) samples with those pseudo-labels.

Furthermore, in order to explore the SSL domain we proposed 3 experiments to comprehend the effect that the unlabeled data have in assessing the model to map latent representations of the target datasets: up-sampling and down-sampling particular difficult classes (5.1), changing the amount (ratio) of unlabeled data (5.2), and, finally, combining as unlabeled images dataset with same nature to assess the mapping of the latent representations (5.3).

3 Methods

3.1 FixMatch

Aiming at leveraging unlabeled data, Sohn et al [1] propose *FixMatch*, a combination of two common SSL methods: consistency regularization and pseudo-labeling. The novelty of FixMatch comes from combining both methods, as well as using a strong and a weak augmentation in the consistency regularization. **Consistency regularization** uses unlabeled data under the assumption that the model should output similar predictions when using a modified version of the input data. **Pseudo-labeling** considers that the model should be used to obtain artificial labels for unlabeled data by means of taking the `argmax` of the model's output, only retaining those in which the probability of belonging to a class is above a certain threshold (i.e. the model should provide high confidence predictions). As part of the algorithm, *FixMatch* applies first a weak augmentation in order to produce a weakly-augmented version of the unlabeled image, which is fed into the model for prediction (see the upper path in figure 1). Once the model assigns a probability of that image belonging to a specific class and if it is above a certain threshold (see dotted line in figure 1), the prediction is converted to a one-hot pseudo-label.

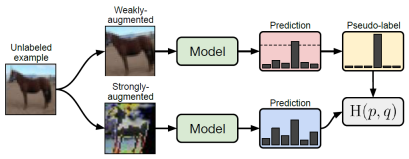

Figure 1: Diagram of *FixMatch* [1].

Then, a strong augmentation is performed to the original data sample (see the bottom path in figure 1). The model is then trained to make its prediction on the strongly augmented version match the pseudo-label created in the upper path of figure 1 by means of a standard cross-entropy loss ($\mathrm{H}(p, q)$ represents the cross-entropy between distributions p and q). The weak augmentation is standard flip-and-shift augmentations whereas the strong augmentations are based on AutoAugment, a procedure able to learn and apply augmentation strategies based on Python Libraries [5].

However, since the learning process of AutoAugment requires additional labeled data to train, it is a problem in SSL, thus the authors decide to perform the variants RandAugment and CTAugment, able to learn augmentation strategies without the need for additional labeled data. It was the latter, CTAugment, the one used during this project following the description of Berthelot et al [4]. Although both RandAugment and CTAugment were used by the authors, due to time constraints only CTAugment was used given that the results obtained with this method were slightly better.

Algorithm 1: FixMatch

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}\left(p_b, \alpha\left(x_b\right)\right)$ *// Cross-entropy loss for labeled data*

for b = *1* **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* u_b

  $q_b = p_{\mathrm{m}}\left(y \mid \alpha\left(u_b\right); \theta\right)$ *// Compute prediction after applying weak data augmentation of* u_b.

end

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left\{\max\left(q_b\right) > \tau\right\} \mathrm{H}\left(\arg\max\left(q_b\right), \tilde{u}_b\right)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

FixMatch loss function consists of two cross-entropy loss terms: a supervised loss ℓ_s and an unsupervised loss ℓ_u. ℓ_s is simply the standard cross-entropy given by $\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}\left(y_b, p_{\mathrm{m}}\left(p_b \mid \alpha\left(x_b\right)\right)\right)$, where ***B*** is the number of labeled samples, y_b is the label corresponding to data sample x_b and $p_{\mathrm{m}}\left(p_b \mid \alpha\left(x_b\right)\right)$ is the predicted class distribution produced by the model given the weakly augmented labeled sample $\alpha\left(x_b\right)$. On the other hand, for unlabeled data, the cross-entropy loss is given by $\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(p_{\mathrm{m}}\left(p_b \mid \alpha\left(x_b\right)\right)\right) \geq \tau\right) \mathrm{H}\left(\operatorname{argmax}(p_{\mathrm{m}}\left(p_b \mid \alpha\left(x_b\right)\right)), p_{\mathrm{m}}\left(y \mid \mathcal{A}\left(u_b\right)\right)\right)$, where μB is the number of unlabeled samples, τ is the aforementioned threshold, and $p_{\mathrm{m}}\left(y \mid \mathcal{A}\left(u_b\right)\right)$

is the predicted class distribution produced by the model given the strongly augmented unlabeled sample $\mathcal{A}(u_b)$. The final loss of *FixMatch* would be simply $\ell_s + \lambda_u \ell_u$, where λ_u is a fixed scalar hyperparameter that assigns a weight to the unlabeled loss.

3.2 Control Theory Augment (CTAugment)

As mentioned in the previous section, CTAugment was the method applied to perform the required strong augmentations. It is based on AutoAugment, a method for learning data augmentation policies (sequences of parameter magnitudes to be used with each transformation applied). These magnitudes are not applied randomly, but they require to be learned using 4,000 labels on CIFAR-10 and 1,000 labels on SVHN on a proxy task. Nonetheless, as SSL is a framework where labeled images are scarce, the fact that AutoAugment requires these additional labels to train is problematic. In order to tackle this, Berthelot et al [4] developed CTAugment, which randomly samples magnitudes for the parameters of each transformation. However, unlike RandAugment and AutoAugment, it learns the best magnitudes to be applied during the training process and without the need of a proxy task.

CTAugment applies different types of transformations and, for each one of them, it learns the likelihood of the transformed image to be classified as the correct label. Using these likelihoods, CTAugment will in the end sample the transformation hyperparameter values that will produce high likelihoods. In order to do so, first, the range of values that each of the parameters of the transformations may take is divided into bins, and weights are assigned to each of these bins (all the weights are initialized as 1). At each training step, for each image two transformations are sampled uniformly at random. Then, for each hyperparameter of the two transformations, another sampling process is performed in order to select the particular value that the hyperparameter will take (first a bin is selected with a probability proportional to its weight and then a value within the bin is sampled). The resulting transformations are applied to a labeled image x with label y in order to obtain the augmented image $\hat{x}$. Once this is done, the augmented version $\hat{x}$ is input in the model to perform a prediction and compute to which extent the model's prediction matches the one-hot encoding label y with the following formula $\omega = 1 - \frac{1}{2}\sum |p_{\text{model}}(\hat{y_i} \mid \hat{x}; \theta) - y|$, where $p_{\text{model}}(\hat{y} \mid \hat{x}; \theta)$ is the prediction of the model for the ith class given the hyperparameter value θ. With this process, a weight m_j value for each of the j hyperparameter bins is generated and updated with the following formula: $m_j = \rho m_j + (1 - \rho)\omega$ where ρ is a fixed decay parameter.

4 Data

In this project three different datasets are used: CIFAR-10, SVHN, and MNIST. All these three datasets consist of 10 different classes, being CIFAR-10 composed by images of different items while the other two are composed by images of digits from 0 to 9.

CIFAR-10 did not require any extra preprocessing when loaded in the model, besides the data augmentations of the method itself. Nevertheless, the structure of the SVHN images had to be modified in order to match the channels input format expected by the model, and MNIST had to be scaled from a 28x28 single channeled image to a triple channeled 32x32 image (where the original channel was replicated) in order to be processed by the system.

CIFAR-10 and SVHN results have been reported employing different models for the same goal than FixMatch. MixMatch, and its later improvement, ReMixMatch, some of the main models employed in this topic, provided these results.

Dataset	CIFAR-10			SVHN
	40 labels	250 labels	4000 labels	250 labels
MixMatch	52.46% ± 11.50	88.95% ± 0.86	93.58% ± 0.10	96.02 ± 0.23
ReMixMatch	80.9% ± 9.64	94.56% ± 0.05	95.28% ± 0.13	97.08 ± 0.48

Table 1: State-of-the-art performance on employed datasets.

5 Experiments and Findings

Initially, some experiments over CIFAR-10 and SVHN datasets were performed in order to reproduce the original paper FixMatch. The outcomes obtained can be seen in the following table.

Dataset	Labeled Samples	Test Acc.
CIFAR-10	4000	90.70%
CIFAR-10	250	85.44%
CIFAR-10	40	64.44%
SVHN	250	95.29%

Table 2: Performance of tested datasets with different amounts of labeled samples.

CIFAR-10 models were trained during 1024 epochs (approximately 2^{20} training steps), while the SVHN model was trained during 135 epochs. This number of epochs was chosen due to training time constraints and since it could be observed that the model already converged when the accuracy was plotted. In order to satisfy training time constraints too, CTAugment was employed every 10/15 training steps instead of in each one of them, which would correspond to training one model for more than 260 hours under a Nvidia Tesla - P100 GPU. Additionally, as stated in FixMatch, randomness has a big influence on the performance of SSL methods, which is also a factor that must be taken into account when trying to achieve similar results.

5.1 Unbalancing the supervised classes

The first one of the extra experiments performed consists of unbalancing one of the classes that compose the subset of the labeled sample in the SVHN dataset. This can be done either by oversampling or downsampling data belonging to a specific class, resulting in either an excess or a lack of samples compared to the other classes.

By training our model on SVHN (figure 2) and then reviewing some other experiments performed with this dataset, it can be seen that the digits 3 and 5 are normally those confused the most in between [6][7]. We consider that this may be caused due to the fact that the model might learn certain features of one class from several classes and changing the amount of samples of one specific number may have an effect on others. This is why it has been decided to modify the number of labeled samples of class 3, in order to analyze its effect over class 5.

Dataset	Scaling Factor	Test Acc. Class 3	Test Acc. Class 5
SVHN	1	92.44%	95.18%
SVHN	0.5	91.29%	96.35%
SVHN	1.5	95.62%	97.06%

Table 3: Performance of the most confused classes (3 and 5) in SVHN employing as base 250 labeled images (25 per class), but scaling the class 3 number of samples by different factors.

It can be observed that downsampling the number of labeled images of class 3 slightly improves class 5 classification, resulting in a trade-off with class 3 accuracy. Nevertheless, by using more labeled samples of class 3, the accuracy of both classes considerably improved. Therefore, it could be said that employing more samples of one of the confused classes improves the performance in both since the model managed to grasp the distinction between them and then enhance its discriminative ability.

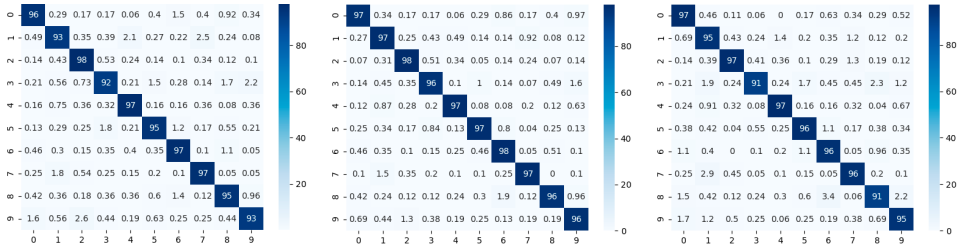

Figure 2: Confusion matrices for original, downsampled, and oversampled cases.

5.2 Labeled-Unlabeled data ratio

The next experiment performed was modifying the ratio of labeled and unlabeled data employed during training the CIFAR 10 dataset in order to observe how the performance of the model varied while changing the amount of labeled and unlabeled images. On one hand, the amount of unlabeled data was reduced for a fixed amount of labeled data. This has been tested employing only 40 labeled samples in total (4 per class), resulting in the following outcomes:

Dataset	Percentage of Total Unlabeled Data	Test Acc.
CIFAR-10	25%	49.00%
CIFAR-10	50%	51.23%
CIFAR-10	75%	57.93%
CIFAR-10	100%	64.44%

Table 4: Performance of the most confused classes (3 and 5) in SVHN employing as base 250 labeled images (25 per class), but scaling the class 3 number of samples by different factors.

For the cases with a lower percentage of unlabeled data, the best accuracies were reported in the past epochs. Which may infer that the models were still learning, but that it does it slower since the amount of unlabeled samples is reduced. It can be observed that the amount of unlabeled data has a great influence on the final accuracy, since it was observed that the bigger the percentage, the higher the accuracy. It is worth mentioning that the learning curves presented in Appendix A1 and A2 showcase that more unlabeled data assess the model to faster achieve better results, however after a while the model could not improve its confidence in the unlabeled data, which is clearer when observing the loss curves. We reason that this happened due to the lack of CTAugment iterations, since when performing more iterations the ratio of pseudo-labels that surpassed the threshold considerably increased. For instance, training the same model with CTAugment every 10 training steps, rather than every 15, increased the ratio of unlabeled data used from 40% to 60%. As a consequence, the overall performance of the model also increased given that it is able to learn better representations with more unlabeled samples that surpassed the threshold.

The second part of this extension involved changing the amount of labeled data while keeping the amount of unlabeled data fixed during training. Due to the long time required to train each model (see section 6), these experiments could not be performed. Thus, in order to do the required comparison, the experiments from table 2 were used for this purpose assuming that the difference in unlabeled data between each experiment has a minor impact in the resulting accuracy: the maximum difference in the amount of unlabeled data used was 3,960 images, which is a rather small amount considering that the training data set of CIFAR-10 is 50,000 images.

Finally, the relationships between *Labeled data* $vs.$ *Accuracy* and *Unlabeled data* $vs.$ *Accuracy* was plotted as it can be observed in Appendix A3 . Interestingly, it can be observed that the accuracy and the amount of labeled data have a logistic relationship. It changes very abruptly depending on the amount of labeled data used for very small amounts (for instance, when 210 labeled images are added to the experiment performed with 40 labeled images, the accuracy improves by 21.46% as observed in table 2). Then, for larger amounts of labeled data, more labeled examples cause smaller improvements in accuracy as happens when increasing the amount of labeled data from 250 to 4,000 images, where only a 5.26% increase was observed. On the other hand, the amount of unlabeled data and the accuracy follow a rather linear relationship. Linking these observations with the problems exposed in the Introduction section, identifying the relationships between the gain in terms of accuracy when increasing labeled data or unlabeled data can be very useful as both processes have different costs. For instance, taking once more the example of medical data, in a hypothetical scenario in which it should be decided whether to acquire more unlabeled medical data or hire highly qualified personnel to label already existing unlabeled data, it can be very helpful to know this relationship. This would allow to decide what is more convenient depending on the accuracy being achieved at the moment, the amount of labeled/unlabeled data that could be obtained, the cost of obtaining the data and its estimated effect on the accuracy.

5.3 Merging different unlabeled datasets

The last of the tests performed consist of employing samples of two different datasets, but of the same nature, as the unlabeled samples utilized during the model training. Our reasoning to perform such an experiment is based on the fact that real-world datasets may share certain similarities and we believed that it would be a waste of data and resources not combining this kind of datasets. As mentioned earlier labeled datasets are expensive and we wanted to prove that merging datasets can be an appropriate approach to optimize resources and achieve good performances. In this case, the datasets merged were SVHN and MNIST since both datasets represent the same kind of data (digits from 0 to 9). The results reported are the following:

Labeled Data Dataset	Unlabeled Data Datasets	Test Dataset	Test Acc.
SVHN	SVHN	SVHN	95.29%
SVHN	SVHN	MNIST	73.01%
SVHN	SVHN and MNIST	SVHN	96.51%
SVHN	SVHN and MNIST	MNIST	86.04%

Table 5: Performance of the most confused classes (3 and 5) in SVHN employing as base 250 labeled images (25 per class), but scaling the class 3 number of samples by different factors.

It can be seen that training with both SVHN and MNIST datasets as unlabeled samples not only slightly improved the performance in SVHN, but it also learned a better latent representation of MNIST, being able to increment the test accuracy by more than a 13%. Summarizing, a model trained with both SVHN and MNIST as unlabeled data has been able to perform better in both datasets than a model uniquely trained with SVHN data.

It is necessary to highlight that this model performed almost perfectly on MNIST dataset, except for classes 7 and 1. For 7, some samples were classified as fours, and for 1 almost all samples were wrongly classified as 7. A possible explanation for this is the shape of the ones in MNIST, being many of them just one-line shaped, probably making them more similar to SVHN sevens rather than ones. A similar thing could have happened in the case of the seven that were mistaken by four.

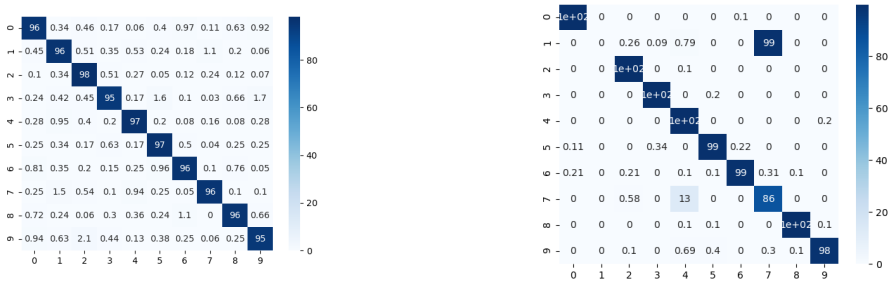

Figure 3: Confusion matrices for SVHN and MNIST.

6 Challenges

The authors of Fixmatch wrote in general an article that allowed the reader to understand the general concepts easily and without the need for many external sources. However, during the implementation, we found several details that were explained in a very succinct way or not specified. Therefore, there were parts of the implementation that probably required a better explanation for proper reproducibility.

As an example, regarding the unlabeled data, there were several details that should have been better specified. One of them is the amount of unlabeled data used in each experiment. It may happen that by going through the paper, the reader may understand that the unlabeled data set had the same size over different experiments and it was just the amount of labeled data modified (thus, it could be thought that there is a fixed-size unlabeled data set and a fixed-size labeled data set from which the labeled images were sampled). However, after reviewing the authors' code it was observed that the

unlabeled data was not fixed but it was the result of taking the labeled images out of the original data set and using the rest as unlabeled data.

Moreover, the CTAugment was very briefly introduced in the paper and directly referenced to the ReMixMatch paper [4]. However, even the explanation provided in this paper seems a bit short as to fully understand how it works, how to update the policies, and also when to apply CTAugment within Fixmatch. We found this last part to be very important for the performance of the models as there was a significant difference - in terms of evaluation and time performance - depending on the number of times that the CTAugment was executed. This, in addition, leads to another challenge we found, which was the amount of time required to train the models. With the available hardware (Nvidia Tesla K80, Nvidia Tesla P100, and Nvidia GeForce RTX2070 GPUs) training the experiments could take from 25 to 40 hours when doing the CTAugment every 10-15 training steps. Training the same experiments but with CTAugment being applied every training step was estimated to take more than 260 hours. Thus, our experiments had to be run with configurations in which less CTAugment training steps were performed in order to meet the deadlines at the cost of decreasing our performance. Finally, the authors briefly mention they apply Exponential Moving Average (EMA) of model parameters however no clear explanation on how they applied the EMA was provided.

7 Conclusion

FixMatch has proven to be a prime method in the field of semi-supervised learning. By means of just a few labeled images, it is possible to achieve great results when combined with unlabeled data. CTAugment has also demonstrated its efficacy. Nevertheless, it is a really heavy technique to employ which requires lots of computing power. Therefore, in order to achieve the best results possible, it is necessary to count on enough resources as explained in the original FixMatch paper. Furthermore, in order to obtain the best of the models, choosing the proper hyperparameters has also proven to be crucial, resulting in a huge variability in the retrieved results.

From the experiments performed, the merging of SVHN and MNIST datasets may be the most interesting one. Further research in this area, combining datasets of similar nature in order to enhance the model performance might be an appealing topic of study.

8 Societal Impact

As mentioned before, unlabeled data is abundant, yet expensive to label. As shown in section 5.2, it is possible to estimate the trade-off between expenses of acquiring labeled and unlabeled data and the performance of semi-supervised learning methods. Thus, medical images diagnosis proves to be an ideal scenario to apply SSL methods due to its amount of unlabeled images and the cost of hiring professionals to correctly label them.

Valentyn Melnychuk et al. [8], recently published their studies with regards to retinal optical coherence tomography (OCT) medical images and the employment of MixMatch and FixMatch. Their results demonstrated how powerful SSL can be since they outperformed transfer learning techniques and achieved very similar results as fully supervised approaches. Therefore, due to its scalability, SSL can provide a huge societal impact once applied to the health care domain. For instance, it could be used to enhance the diagnoses of lung diseases, considering that it is a complex problem difficult to be spotted even by professional human eyes [9] and due to the number of different diseases that the same patient can exhibit concurrently (multi-labeled problem).

9 Self-Assessment

In general, the project fulfilled its purpose in providing an implementation in PyTorch of FixMatch and CTAugment, presenting different experiments and new insights, such as the trade-off between expending more in labeled data in favor of better performance and going beyond the scope of SSL by merging similar datasets to improve the latent representations. In addition, we believe that the structure of the report is coherent and easy to understand, which instigates the reader to comprehend our experiments and how to extrapolate them to other analyses. Finally, the project was finished and submitted on the first stipulated deadline. For all these reasons we consider this project is eligible for an A grade.

References

[1] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[2] A. H. Ribeiro, M. H. Ribeiro, G. M. Paixao, D. M. Oliveira, P. R. Gomes, J. A. Canazart, M. P. Ferreira, C. R. Andersson, P. W. Macfarlane, M. Wagner Jr, *et al.*, "Automatic diagnosis of the 12-lead ecg using a deep neural network," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.

[3] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," *arXiv preprint arXiv:2002.07394*, 2020.

[4] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[5] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123, 2019.

[6] M. Tipari, "Digit classification in images using convolutional neural network," 2019.

[7] A. Sharma, "Svhn-cnn." `https://github.com/aditya9211/SVHN-CNN/blob/master/svhn_model.ipynb`.

[8] V. Melnychuk, E. Faerman, I. Manakov, and T. Seidl, "Matching the clinical reality: Accurate oct-based diagnosis from few labels," 2020.

[9] M.-Y. Ng, E. Lee, J. Yang, F. Yang, X. Li, H. Wang, M. Lui, C. Lo, B. S. T. Leung, P. Khong, C. Hui, K.-y. Yuen, and M. Kuo, "Imaging profile of the covid-19 infection: Radiologic findings and literature review," *Radiology: Cardiothoracic Imaging*, vol. 2, p. e200034, 02 2020.

Appendices

A Experiments and Findings Plots

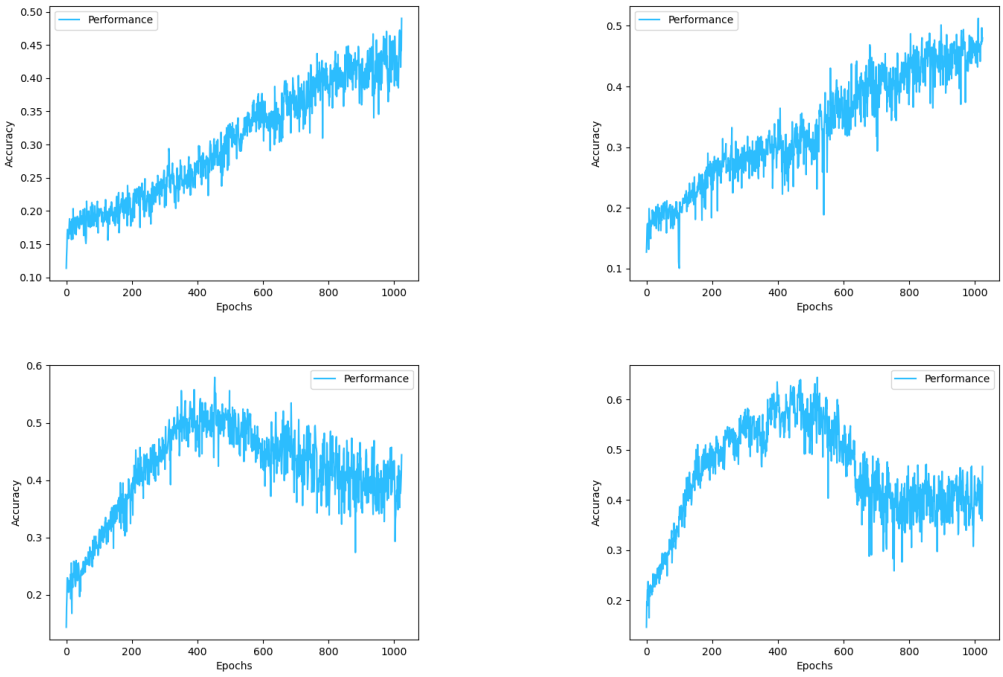

Figure A1: CIFAR-10 accuracy employing 40 labels and 25%, 50%, 75%, and 100% of unlabeled data.

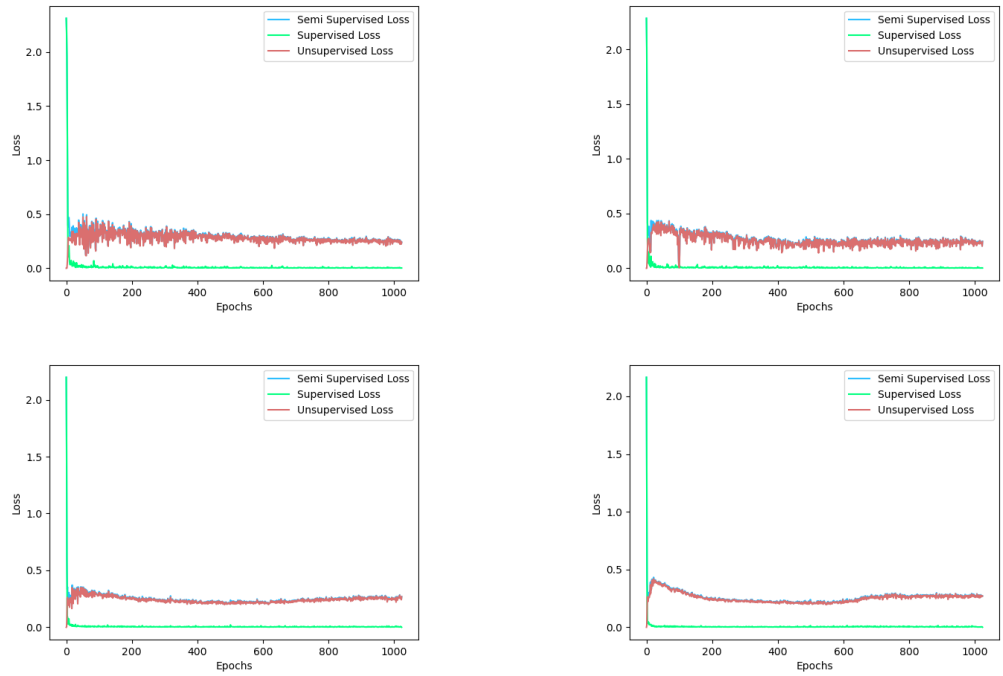

Figure A2: CIFAR-10 loss curves employing 40 labels and 25%, 50%, 75%, and 100% of unlabeled data.

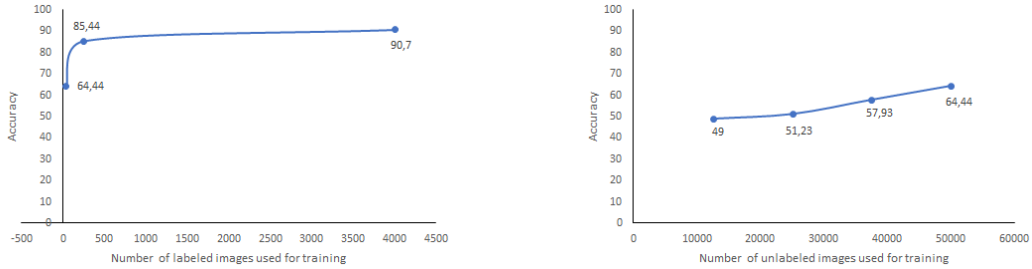

Figure A3: Plotted relationship between accuracy and amount of labeled data and unlabeled data.

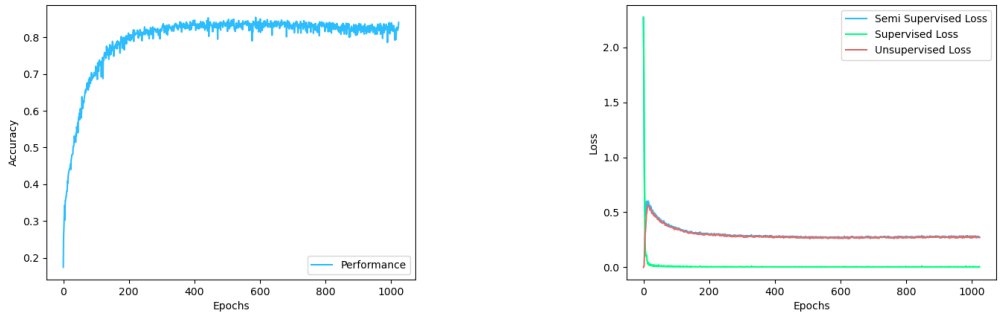

Figure A4: CIFAR-10 reproduction accuracy and loss curves employing 250 labeled data.

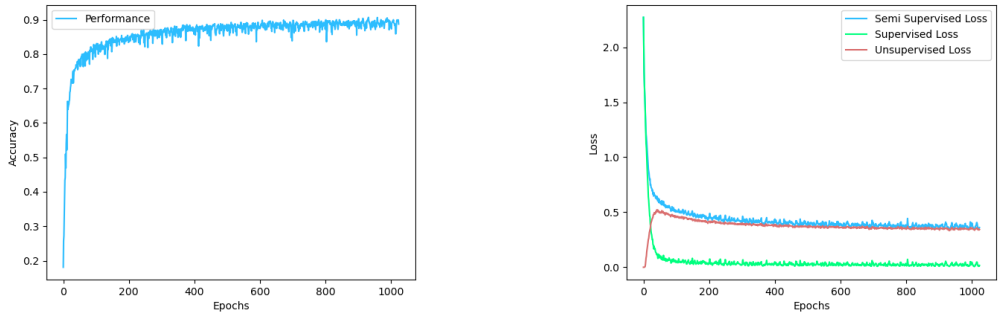

Figure A5: CIFAR-10 reproduction accuracy and loss curves employing 4000 labeled data.

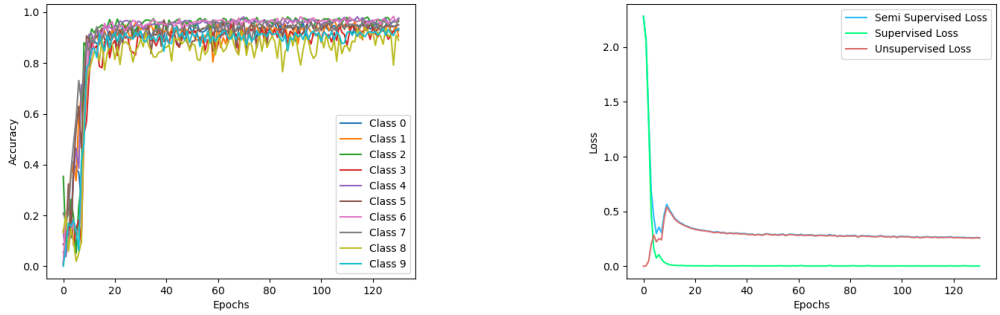

Figure A6: SVHN reproduction accuracy and loss curves employing 250 labeled data.

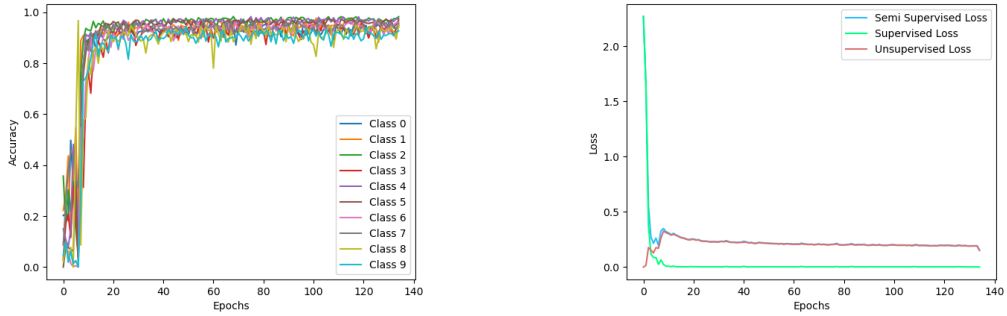

Figure A7: SVHN merged with MNIST accuracy and loss curves employing 250 labeled data of SVHN.

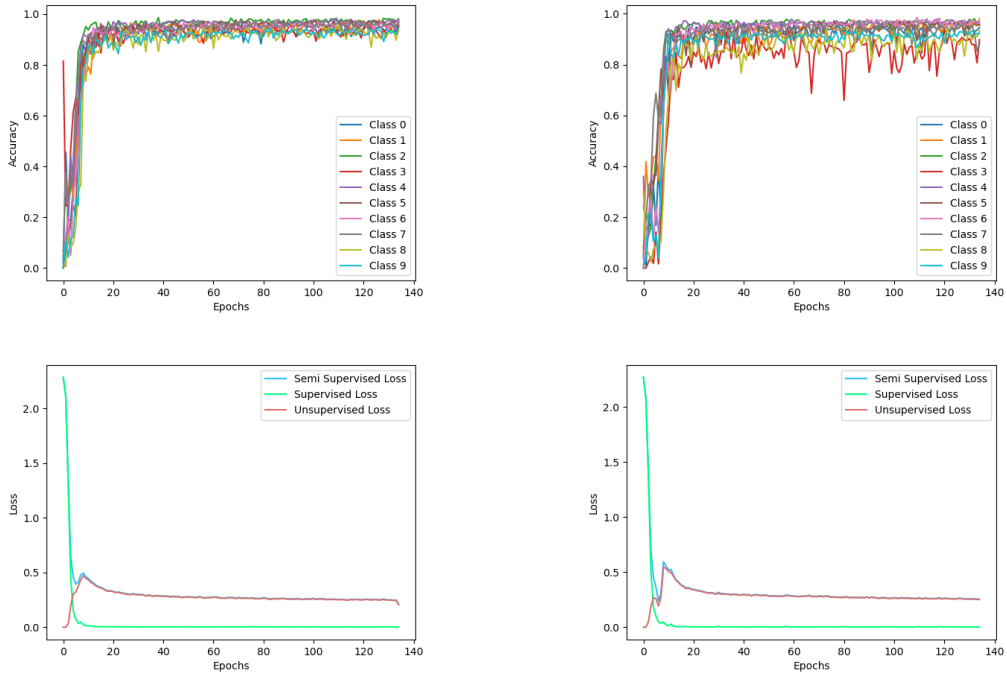

Figure A8: SVHN upsampling (right) and downsampling (left) class 3 accuracies and loss curves employing 250 labeled data.