Matching the Clinical Reality: Accurate OCT-Based Diagnosis From Few Labels

Valentyn Melnychuk\textsuperscript{a,b}, Evgeniy Faerman\textsuperscript{b}, Ilja Manakov\textsuperscript{b} and Thomas Seidl\textsuperscript{b}

\textsuperscript{a}Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany
\textsuperscript{b}Ludwig Maximilian University of Munich, Germany

Abstract
Unlabeled data is often abundant in the clinic, making machine learning methods based on semi-supervised learning a good match for this setting. Despite this, they are currently receiving relatively little attention in medical image analysis literature. Instead, most practitioners and researchers focus on supervised or transfer learning approaches. The recently proposed MixMatch and FixMatch algorithms have demonstrated promising results in extracting useful representations while requiring very few labels. Motivated by these recent successes, we apply MixMatch and FixMatch in an ophthalmological diagnostic setting and investigate how they fare against standard transfer learning. We find that both algorithms outperform the transfer learning baseline on all fractions of labelled data. Furthermore, our experiments show that Mean Teacher, which is a component of both algorithms, is not needed for our classification problem, as disabling it leaves the outcome unchanged.

Our code is available online: gitlab.com/Valentyn1997/oct_diagn_semi_supervised.

Keywords
Semi-supervised image classification, Transfer learning, Optical Coherence Tomography

1. Introduction

In recent years deep learning techniques have taken the field of AI by storm. Virtually all state-of-the-art systems in computer vision (CV) rely on some form of deep learning. This paradigm shift has sparked the imagination of many practitioners and researchers in the medical image analysis domain. Computer-aided diagnosis appeared to be next-in-line to benefit from the advancements made in CV, as the amount of data in clinical diagnostics is increasing rapidly. The research community has proposed a plethora of new algorithms and systems for the automated diagnosis of a wide range of diseases. However, clinical adoption has been slow. One crucial reason is that supervised learning, which forms the basis for the vast majority of deep learning approaches, is ill-suited to the medical domain.

This mismatch is two-fold. For one, the labelled data needed for supervised learning is prohibitively costly to generate for medical applications. With a shortage of medical practitioners, diverting medical experts’ time and energy to labelling efforts becomes exceedingly expensive. More fine-grained problem formulations (e.g. single-label vs. multi-label, volume level vs. slice level annotation, etc.) result in exponentially more labelling expenses. Additionally, most clinics lack the tools to label vast amounts of data. Secondly, and perhaps more fundamentally, there is an epistemic problem in generating accurate labels. For any given diagnostic problem, the inter-expert agreement is well below 100%. This discrepancy stems from the fact that medicine is complex and does not always fit neatly into a classification formulation. Additionally, each expert comes with his or her own set of experiences and knowledge.

Instead of solely relying on supervised learning, semi-supervised learning (SSL) should discover the bulk of the knowledge required for solving a diagnostic task on its own, with labels only serving as additional guidance. The idea of SSL is to train a machine learning algorithm on vast amounts of unlabeled data and a small set of labelled samples. SSL is a much better match for the clinical setting, as unlabeled data is often abundant since it is acquired as part of the clinical routine.

In this work, we apply two recently proposed SSL methods, MixMatch \cite{melnychenko2020mixmatch} and FixMatch \cite{sohn2020fixmatch}, to a diagnostic problem in ophthalmology. We test which performs better in classifying optical coherence tomography (OCT) b-scans into four classes (one healthy and three pathological) at different fractions of labelled data. We compare the two SSL methods to a baseline transfer learning approach, similar to \cite{melnychenko2019}. After going over related work in the next section, we explain the basis for our experiments in Section 3, covering MixMatch, FixMatch and the transfer learning baseline. In Section 4 we describe the dataset and present the results of our
investigation. We conclude with Section 5 by summarizing our findings and discussing how they apply to the clinical setting.

2. Related work

Semi-supervised learning. State-of-the-art methods for image classification concentrate on finding the right combination of SSL paradigms. One of the early approaches – Mean-Teacher [4] – uses exponential moving average (EMA) of model parameters. Virtual Adversarial Training (VAT) [5], tries to find a minimal perturbation and fit a robust model against it. MixMatch [1] and RealMix [6] encompass mixing and overlaying labelled and unlabelled images to obtain consistent predictions. Unsupervised Data Augmentation (UDA) [7] uses strongly augmented images to force consistency among unlabeled images. ReMixMatch [8] uses so-called “augmentation anchoring”, i.e. strong and weak augmentations, to enforce consistency. Inspired by UDA and ReMixMatch, the authors of FixMatch [2] significantly simplify SSL by relying only on augmentations and pseudo-labelling with a confidence threshold. We provide a broader overview of applied SSL methods in Section 3.2.

Surprisingly, there exists only a little amount of literature on SSL applied to ophthalmological data. [9] and [10] utilize SSL for OCT segmentation. In the domain of automated diagnosis, [11] employ an autoencoder with an additional classification module on the latent code to detect retinopathy from colour fundus images. [12] tackle the same problem by extending the GAN framework [13] to one “fake” and six “real” classes, i.e. the labeled classes. Recent works [14] and [15] apply the same principle to the classification of OCT b-scans. Most recently, [16] applied SSL methods to glaucoma detection by imputing missing visual field (VF) measurements through nearest-neighbour identification in the latent space of a pre-trained classification CNN. Afterwards, [16] train a multi-task network jointly on glaucoma classification and VF measurement prediction. To the best of our knowledge, we are the first to apply consistency regularization based SSL techniques (see Section 3) to the problem of automated diagnosis in ophthalmology.

Transfer learning. Among numerous approaches existing in the deep transfer learning [17], we choose the fine-tuning or network-based transfer learning to be the most promising. [18, 19] proposed to use ImageNet [20] pre-trained CNN as the initialization for different visual recognition tasks with the limited amount of labels. Yosinski et al. [21] discovered how unfreezing different parts of the network while fine-tuning affects the target performance.

3. Approach

Transfer learning and semi-supervised learning are two main approaches for predictive modelling when dealing with data with few labels. Transfer learning approaches reuse knowledge from previously learned tasks. On the other hand, the SSL approaches allow learning with small labelled datasets by utilizing unlabeled data from the same distribution in the learning process. In the following, we first discuss our transfer learning baseline and afterwards describe the SSL approaches we have chosen for this study.

3.1. Transfer Learning

When applying transfer learning techniques, the user has to choose how to adapt the model from the auxiliary to the primary task. In our experiments, we use a network, which was pre-trained on ImageNet [20]. For adapting the model to OCT classification we try two common approaches. In the feature extraction approach, we freeze all parameters except for the final fully connected layer, analogous to [3]. Alternatively, we use the pre-trained network as initialization and allow all parameters to change. We refer to this as fine-tuning hereafter.

3.2. Semi-supervised Learning

In our study we compare two of recent state-of-the-art algorithms for SSL, MixMatch [1] and FixMatch [2]. Both algorithms combine several pre-existing techniques from SSL. In this chapter, we review the main ideas and compare their utilization in both algorithms. We refer the reader to Appendix A for the detailed algorithm descriptions.

Data Augmentation. Data Augmentation is a regularization technique which is often used in supervised learning. The goal is that the model’s prediction is not affected by the certain transformation of data instances. Therefore additional training data is added to the dataset by applying various perturbations to the data while keeping original labels. Most of the data augmentations are domain-specific and require domain knowledge.
MixMatch uses random flip-and-shift augmentations (horizontal flips and random crops) for both labelled and unlabeled data.

FixMatch distinguishes between weak and strong data augmentations. Flip-and-shift augmentations are considered as weak augmentations, whereas affine transformations and color-jittering are examples of strong augmentations (originally ~ 14 different transformations from RandAugment [22]).

Pseudo-Labelling. Pseudo-Labelling or self-training loss [23] is the process of using the trained model to obtain labels for unlabeled instances. The predicted labels are used to guide the further learning process, e.g., by using generated labels as new targets.

MixMatch applies different augmentations for an unlabeled instance and computes the class distribution for each augmentation. Therefore, instead of hard one hot label MixMatch defines a probability distribution as the target. To sharpen the distribution and to reduce its entropy, the temperature of distribution is adjusted [24].

FixMatch uses a “classic” version of pseudo-labelling with hard labels and fixed confidence. The class probability distribution is taken from model outputs after a weak augmentation. If the probability of the most probable class exceeds a predefined threshold the label is assigned to a strongly augmented version of the same instance and used in the loss calculation.

Consistency Regularization. Consistency regularization [25] imposes the constraint that the model should make similar predictions for the same instance under different data augmentations. Both MixMatch and FixMatch apply data augmentation on labelled and unlabeled data and enforce similar prediction for the same instance under different augmentations. For the unlabeled instances, the pseudo-label is used as a target. FixMatch uses soft augmentations to compute pseudo-labels for hard augmentations of the same training sample.

Mean Teacher. Another popular consistency requirement in SSL is a similar prediction over time or punishing the behaviour when the model changes its decisions rapidly. The Mean Teacher algorithm [4] maintains two models. The teacher model stores an exponential moving average of student’s parameters and is used to make the predictions to compute the pseudo-labels. Therefore pseudo-labels computed by the teacher can be considered as a weighted combination of decisions of previous models. The student model makes the predictions for the training data and is updated based on the training loss. Both MixMatch and FixMatch employ Mean Teacher for the computation of pseudo-labels. Note, that keeping a second model in memory and updating its parameters results in higher memory requirements and computation costs.

MixUp. MixUp [26] is another regularization technique to avoid overfitting. MixUp linearly combines training instance pairs and their prediction targets. Therefore it tries to impose linear behaviour between training samples. MixMatch does not differentiate between pseudo targets predicted for the unlabeled instances and ground truth labels and mixes all possible target pairs. Therefore a resulting instance used in training may be a combination of two pseudo targets, two ground truth labels or of pseudo-target with ground truth label.

4. Experiments

Our work follows the principles of the fair SSL evaluation framework, defined by [27]. The authors highlight the importance of using the same classifying model structure for comparison. The evaluation is also meaningful for the real use-case if SSL methods are compared with well-fine-tuned transfer learning and fully supervised models.

For the evaluation we use the UCSD dataset published by Kermany et al. [3]. It contains 84,495 optical coherence tomography (OCT) b-scans pertaining to four categories: normal, drusenoid (DRUSEN), choroidal neovascularization (CNV) and diabetic macular edema (DME). The images vary in size, where the median image has a size of 496×512 pixels. The height of the images ranges between 496 and 512 and the width between 384 and 1536. The dataset is also obtainable through Kaggle. For better comparability, we use the same train/validation/test split as in the Kaggle challenge. There are several images for each patient in the dataset and splits are done patient-wise, there are no images of the same patients in different splits. Test and validation are balanced, there are 8 and 242 images per class respectively (see Fig. 1). In our experiments, we vary the number of labelled data, which we sample randomly from the training subset. We sample the same number of labelled training instances from each class. For SSL approaches the rest of the train set is used as unlabeled data.

We compare the performance of transfer learning and SSL models using the same Wide ResNet-50-2

backbone. Since the images are monochrome we duplicate the channel three times for RGB channels.

For each model, we perform hyperparameter search, described in Appendix B.1 and B.2. For all experiments, we report the model performance on test data in the epoch with the lowest validation loss.

### 4.1. Comparison of transfer learning and SSL approaches

First, in Table 1 we compare the performance of our backbone model trained with all labelled instances to the results reported previously in the literature for the same UCSD dataset. As we can see, the backbone model achieves almost perfect performance when trained with enough labels.

Next, in Fig. 2b we compare two transfer learning approaches. Note, that the hyperparameter search was done for each number of labels for each approach. We discover that, contrary to our expectations, the fine-tuning variant outperforms feature extraction approach in all label settings. We believe that a thorough selection of hyperparameters with representative validation set reduces the risk of overfitting. Furthermore, since the original models are trained on the dataset with RGB channels, we believe that the model can better adapt to the monochrome setting when all model weights are allowed to be changed.

In the Fig. 2a we present the results of both SSL algorithms and compare them with the best performing transfer learning setting. We find that the SSL approaches outperform transfer learning on all fractions of labelled data. The gap between SSL and transfer learning widens significantly for smaller fractions of labelled data. With only 10 labelled representatives per class, the FixMatch achieves an accuracy of over 86%, while transfer learning reaches only 59%. We also see, that with about 2000-4000 labels all methods achieve almost perfect performance. The Fix-Match algorithm also outperforms Mix-Match in almost all settings and with only 50 labelled points per class achieves the accuracy of 98.14%. We also observe a small SSL performance drop for 25 labelled images per class – mainly because methods require even more epochs to fit (we employ a heuristical formula for defining the maximum number of epochs based on the number of labels, see Appendix B.2, 3).

Finally, since practitioners have often to deal with the resource constraints and actual running times are rarely reported in the literature, we report them in Table 2. Note, that all methods are implemented in the same framework and the experiments are done on the same machine with two Tesla V100 Nvidia GPUs. To use the same batch size as recommended in the original publications, we have used both GPUs to train Fix-Match. Other models are trained on a single GPU.

<table>
<thead>
<tr>
<th>Method</th>
<th>$n_l$</th>
<th>Accuracy</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kermany et al. [3]</td>
<td>All</td>
<td>96.6%</td>
<td>Original paper</td>
</tr>
<tr>
<td>Alqudah [29]</td>
<td>All</td>
<td>97.1%</td>
<td>Extended UCSD with 5 classes</td>
</tr>
<tr>
<td>Wu et al. [30]</td>
<td>All</td>
<td>97.5%</td>
<td></td>
</tr>
<tr>
<td>Chetoui et al. [31]</td>
<td>All</td>
<td>98.46%</td>
<td></td>
</tr>
<tr>
<td>Tsuji et al. [32]</td>
<td>All</td>
<td>99.6%</td>
<td></td>
</tr>
<tr>
<td>WideResNet-50-2 (our backbone)</td>
<td>All</td>
<td>99.69%</td>
<td>With EMA decay ($\beta_{EMA} = 0.999$)</td>
</tr>
<tr>
<td>He et al. [14]</td>
<td>835</td>
<td>87.25±1.44%</td>
<td>*Average precision</td>
</tr>
</tbody>
</table>

Table 1

Reported test accuracies for UCSD dataset. Methods have different backbones and thus are not fully comparable with the proposed SLL methods. Nevertheless, our best fully-supervised model outperforms previously reported methods.
Table 2
Training time comparison between the best models of each approach for varying number of labels $n_l$. We do not include the time, spent on hyperparameter search and report only the training time of single models.

<table>
<thead>
<tr>
<th>Model</th>
<th>40</th>
<th>100</th>
<th>200</th>
<th>800</th>
<th>2000</th>
<th>4000</th>
<th>20000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer Learning</td>
<td>10m</td>
<td>9m</td>
<td>12m</td>
<td>15m</td>
<td>24m</td>
<td>39m</td>
<td>1h 39m</td>
</tr>
<tr>
<td>Mix-Match</td>
<td>1d</td>
<td>16h</td>
<td>5m</td>
<td>6h</td>
<td>13m</td>
<td>2h 30m</td>
<td>2h 37m</td>
</tr>
<tr>
<td>FixMatch</td>
<td>5d</td>
<td>9h</td>
<td>36m</td>
<td>1d</td>
<td>40m</td>
<td>9h 58m</td>
<td>10h 40m</td>
</tr>
</tbody>
</table>

5. Conclusion

In this work, we have demonstrated the efficacy of MixMatch and FixMatch, when applied to an ophthalmological diagnostic problem on OCT data. The two algorithms were able to attain high accuracy, achieving well over 80% on as little as 40 labelled samples (i.e. ten per class). Both algorithms outperformed transfer learning in the few labelled data settings. This study emphasizes the use of SSL methods in the clinical adoption of AI. Although both MixMatch and FixMatch are more computationally expensive than transfer learning, the amount of labelling effort saved by using them is immense. With labelling being one of the biggest factors hindering clinical use of AI methodology, we argue that smarter use of the abundance of unlabeled data already present at the clinic will be a major strategy for overcoming this hurdle.

As part of future work, we propose to also compare SSL approach with the few-shot deep learning.

References

[4] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,
<table>
<thead>
<tr>
<th>$n_l$</th>
<th>$\beta_{EMA}$</th>
<th>40</th>
<th>0.0 0.999</th>
<th>100</th>
<th>0.0 0.999</th>
<th>200</th>
<th>0.0 0.999</th>
<th>800</th>
<th>0.0 0.999</th>
<th>2000</th>
<th>0.0 0.999</th>
<th>4000</th>
<th>0.0 0.999</th>
<th>20000</th>
<th>0.0 0.999</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer Learning</td>
<td>58.96</td>
<td>78.65</td>
<td>90.52</td>
<td>95.00</td>
<td>97.71</td>
<td>98.23</td>
<td>99.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mix-Match</td>
<td>83.02</td>
<td>55.00</td>
<td>75.73</td>
<td>81.98</td>
<td>92.50</td>
<td>88.85</td>
<td>94.69</td>
<td>96.88</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FixMatch</td>
<td>86.33</td>
<td>72.66</td>
<td>82.91</td>
<td>75.88</td>
<td>92.50</td>
<td>88.85</td>
<td>97.07</td>
<td>98.34</td>
<td>98.02</td>
<td>98.23</td>
<td>99.02</td>
<td>98.23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3
Best test accuracy based on several runs (8 runs with varying learning rate, weight decay and backbone unfreezing for Transfer Learning, 2 runs with a varying total number of batches for MixMatch and FixMatch) for varying number of labels $n_l$. We do not observe any profit of using / not using EMA decay ($\beta_{EMA}$) for all methods, unlike previously reported results.


[22] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, arXiv: Com-


Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A and by the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology through the Center for Analytics-Data-Applications (ADA-Center) within the framework of “BAYERN DIGITAL II”. The authors of this work take full responsibilities for its content.
Appendix

A. MixMatch & FixMatch — algorithm details

The foundation of both MixMatch and FixMatch is consistency regularization — the idea that augmentations of the same data point should yield the same label. In this way, the model regularizes itself based on its predictions.

Let $\mathcal{X} = \{(x_b, y_b), b \in (1,...B)\}$ be the batch of labeled examples. $\alpha(\cdot)$ denotes the set of weak augmentations and $\mathcal{A}(\cdot)$ — strong augmentations. $\hat{y} = f_M(x; \theta)$ is the prediction of the model classifier, parametrized by $\theta$. $H(\cdot, \cdot)$ denotes categorical cross-entropy and $\lambda_u$ is unsupervised loss weight.

MixMatch employs only weak augmentations $\alpha(\cdot)$ and MixUp [26]. Let $\mathcal{U} = \{u_b, b \in (1,...,B)\}$ be the unlabeled data batch. The model outputs of $K$ random weak augmentations $\alpha(\cdot)$ of the same unlabeled sample are treated as soft pseudo-labels $q_b$. These soft pseudo-labels are averaged and sharpened with the temperature $T$ for each image in $\mathcal{U}$ to yield a pseudo-label for that image. Then, images from both randomly augmented $\hat{\mathcal{X}}$ and $\mathcal{U}$ are concatenated and shuffled, resulting in set $\mathcal{W}$. Afterwards, samples in $\mathcal{X}$ and $\mathcal{U}$ are weakly augmented and linearly interpolated with samples from $\mathcal{W}$. This results in $\hat{\mathcal{X}}$ and $\hat{\mathcal{U}}$ — “mixed-up” versions of augmented labelled and $K$ unlabelled batches. Coefficients of MixUp are sampled from Beta($\alpha, \alpha$) distribution. The final loss is the sum of categorical cross-entropy for images from $\hat{\mathcal{X}}$ (supervised part) and Brier score for $\hat{\mathcal{U}}$ images (unsupervised part):

$$\mathcal{L} = \frac{1}{B} \sum_{(x, y) \in \mathcal{X}} H(y, f_M(x; \theta)) + \frac{\lambda_u}{KB} \sum_{(u, q) \in \mathcal{U}} \| \hat{q} - f_M(u; \theta) \|^2. \quad (1)$$

MixMatch linearly ramps up $\lambda_u$ from 0 to its maximum after each batch to reduce the influence of unsupervised part during early stages of training.

FixMatch is a more simplified method. Unlabeled data batch $\mathcal{U} = \{u_b, b \in (1,...,\mu B)\}$ is now $\mu$-times bigger. Given the model’s prediction $q_b$ for a weakly augmented unlabelled sample $u_b$, method yields hard pseudo-labels $\hat{q}_b = \text{argmax}(q_b)$ and $\mathcal{U} = \{(u_b, q_b, \hat{q}_b)\}$. Afterwards, the model predicts labels for both a batch of weakly augmented labelled images and a batch of strongly augmented unlabelled images. Only the confident predictions for unlabelled samples are used in the final supervised part of the loss. They are filtered with the threshold $\tau$. The loss of FixMatch is then the sum of two categorical cross-entropies for labelled and unlabelled images:

$$\mathcal{L} = \frac{1}{B} \sum_{(x, y) \in \mathcal{X}} H(y, f_M(x; \theta)) + \frac{\lambda_u}{\mu B} \sum_{(u, q, \hat{q}) \in \mathcal{U}} \| \max(q) - \tau \| H(\hat{q}, f_M(u; \theta)) \quad (2)$$

As we use $\tau$ for filtering confident pseudo-labels, we do not need the linear ramp-up for $\lambda_u$.

B. Experiments

B.1. Transfer learning

We took a version of Wide ResNet-50-2 pre-trained on ImageNet from PyTorch. Transfer learning was fine-tuned for every individual $n_l$, as it did not require much computational budget:

- learning rate $\in \{1 \cdot 10^{-3}, 5 \cdot 10^{-4}\}$
- optimizer weight decay $\in \{0.0, 0.0001\}$
- layers freezing $\in \{\text{Fine-tuning, Feature extraction}\}$ (see Section 3.1)

Further hyperparameters are kept fixed, namely we use Adam optimizer [33], $B = 32$, number of epochs = 50. Additionally, early stopping with the patience of 25 epochs was applied to avoid overfitting.

B.2. MixMatch & FixMatch

Hyperparameter fine-tuning for both SSL methods was two-fold: firstly, we fine-tuned more general parameters on 200 labelled samples ($n_l = 200$) with respect to the validation loss (see Table 4). Secondly, for each specific $n_l$, we tuned subset-size-dependent parameters.

The labeled batch size was $B = 16$ for both algorithms. Additionally, we fix $\mu = 4$, $\tau = 0.7$ for FixMatch. We omit using cosine learning rate decay.

Regarding secondary fine-tuning, after the increase of $n_l$, each epoch becomes proportionally longer. Thus, we propose the following inverse formula to define the number of epochs:

$$\text{Number of epochs} = \text{round} \left( \frac{n_B}{n_l \text{ div } B} \right).$$

https://pytorch.org/hub/pytorch_vision_wide_resnet/.
<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>MixMatch</th>
<th>FixMatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>{0.01, 0.001}</td>
<td>{0.03}</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>Adam, SGD</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>{500, 1000}</td>
<td>{1000, 2000}</td>
</tr>
<tr>
<td>(T)</td>
<td>{0.25, 0.5, 0.75, 0.9}</td>
<td>—</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>{0.25, 0.5, 0.75, 0.9}</td>
<td>—</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>{12.5, 25, 50, 100, 150}</td>
<td>{50, 25.0}</td>
</tr>
<tr>
<td>Grid-search size</td>
<td>320</td>
<td>8</td>
</tr>
</tbody>
</table>

**Table 4**

Primary hyperparameter search grid for SSL methods. Best value is marked with bold font. SGD – stochastic gradient descent with momentum \((\beta = 0.9)\) \[34\]. An epoch is defined by maximum number of batches in labelled subset.

Where \(n_B\) denotes total number of labelled batches, used while training.

While secondary fine-tuning, we vary:

- \(\beta_{EMA} \in \{0.0, 0.999\}\) (EMA decay)
- \(n_B \in \{12000, 15000\}\) for MixMatch /
  \(n_B \in \{24000, 30000\}\) for FixMatch