A Survey on Semi-, Self- and Unsupervised Learning for Image Classification

Lars Schmarje*, Monty Santarossa, Simon-Martin Schröder, Reinhard Koch
Multimedia Information Processing Group, Kiel University, Germany
{las,msa,sms,rk}@informatik.uni-kiel.de

arXiv:2002.08721v3 [cs.CV] 24 Jul 2020

Abstract

While deep learning strategies achieve outstanding results in computer vision tasks, one issue remains: The current strategies rely heavily on a huge amount of labeled data. In many real-world problems, it is not feasible to create such an amount of labeled training data. Therefore, it is common to incorporate unlabeled data into the training process to reach equal results with fewer labels. Due to a lot of concurrent research, it is difficult to keep track of recent developments. In this survey, we provide an overview of often used ideas and methods in image classification with fewer labels. We compare 25 methods in detail. In our analysis, we identify three major trends. 1. State-of-the-art methods are scaleable to real-world applications based on their accuracy. 2. The degree of supervision which is needed to achieve comparable results to the usage of all labels is decreasing. 3. All methods share common ideas while only a few methods combine these ideas to achieve better performance. Based on all of these three trends we discover future research opportunities.

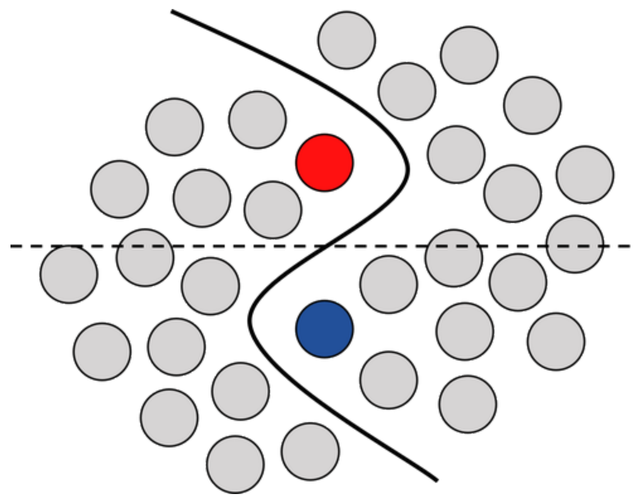


Figure 1: This image illustrates and simplifies the benefit of using unlabeled data during deep learning training. The red and dark blue circles represent labeled data points of different classes. The light grey circles represent unlabeled data points. If we have only a small number of labeled samples available we can only make assumptions (dotted line) over the underlying true distribution (solid line). This true distribution can only be determined if we also consider the unlabeled data points and clarify the decision boundary.

1. Introduction

Deep learning strategies achieve outstanding successes in computer vision tasks. They reach the best performance in a diverse range of tasks such as image classification, object detection or semantic segmentation.

The quality of a deep neural network is strongly influenced by the number of labeled/supervised images. ImageNet [29] is a huge labeled dataset with over one million images which allows the training of networks with impressive performance. Recent research shows that even larger datasets than ImageNet can improve these results [34]. However, in many real-world applications it is not possible to create labeled datasets with millions of images. A common strategy for dealing with this problem is transfer learning. This strategy improves results even on small and specialized datasets like medical imaging [43]. While this might be a practical workaround for some applications, the fundamental issue remains: Unlike humans, supervised learning needs enormous amounts of

*Corresponding author

labeled data.

For a given problem we often have access to a large dataset of unlabeled data. Xie et al. were among the first to investigate unsupervised deep learning strategies to leverage this data [52]. Since then, the usage of unlabeled data has been researched in numerous ways and has created research fields like semi-supervised, self-supervised, weakly-supervised or metric learning [26]. The idea that unifies these approaches is that using unlabeled data is beneficial during the training process (see Figure 1 for an illustration). It either makes the training with fewer labels more robust or in some rare cases even surpasses the supervised cases [24].

Due to this benefit, many researchers and companies work in the field of semi-, self- and unsupervised learning. The main goal is to close the gap between semi-supervised and supervised learning or even surpass these results. Considering presented methods like [57, 53] we believe that research is at the breaking point of achieving this goal. Hence, there is a lot of research ongoing in this field. This survey provides an overview to keep track of the major and recent developments in semi-, self- and unsupervised learning.

Most investigated research topics share a variety of common ideas while differing in goal, application contexts and implementation details. This survey gives an overview of this wide range of research topics. The focus of this survey is on describing the similarities and differences between the methods.

While we look at a broad range of learning strategies, we compare these methods only based on the image classification task. The addressed audience of this survey consists of deep learning researchers or interested people with comparable preliminary knowledge who want to keep track of recent developments in the field of semi-, self- and unsupervised learning.

1.1. Related Work

In this subsection, we give a quick overview of previous works and reference topics we will not address further to maintain the focus of this survey.

The research of semi- and unsupervised techniques in computer vision has a long history. A variety of research and even surveys have been published on this topic. Unsupervised cluster algorithms were researched before the breakthrough of deep learning and are still widely used [33]. There are already extensive surveys that describe unsupervised and semi-supervised strategies without deep learning [55, 61]. We will focus only on techniques including deep neural networks.

Many newer surveys focus only on self-, semi- or unsupervised learning [36, 25, 48]. Min et al. wrote an overview of unsupervised deep learning strategies [36]. They presented the beginning in this field of research from a network architecture perspective. The authors looked at a broad range of architectures. We focus ourselves on only one architecture which Min et al. refer to as "Clustering deep neural network (CDNN)-based deep clustering" [36]. Even though the work was published in 2018, it already misses the recent development in deep learning of the last years. We look at these more recent developments and show the connections to other research fields that Min et al. did not include.

Van Engelen and Hoos give a broad overview of general and recent semi-supervised methods [48]. While they cover some recent developments, the newest deep learning strategies are not covered. Furthermore, the authors do not explicitly compare the presented methods based on their structure or performance.

Jing and Tian concentrated their survey on recent developments in self-supervised learning [25]. Like us, the authors provide a performance comparison and a taxonomy. Their taxonomy distinguishes between different kinds of pretext task. We look at pretext tasks as one common idea and compare the methods based on these underlying ideas. Jing and Tian look at different tasks apart from classification but do not include semi- and unsupervised methods without a pretext task.

Qi and Luo are one of the few who look at self-, semi- and unsupervised learning in one survey [41]. However, they look at the different learning strategies separately and give comparisons only inside the respective learning strategy. We show that bridging these gaps leads to new insights, improved performance and future research approaches.

Some surveys focus not on the general overviews about semi-, self- and unsupervised learning but on special details. In their survey Cheplygina et al. present a variety of methods in the context of medical image analysis [9]. They include deep learning and older machine learning approaches but look at different strategies from a medical perspective. Mey and Loog focused on the underlying theoretical assumptions in semi-supervised learning [35]. We keep our survey limited to general image classification tasks and focus on their practical application.

Keeping the above-mentioned limitations in mind the topic of self-, semi- and unsupervised learning still includes a broad range of research fields. In this survey, we will focus on deep learning approaches for image classification. We will investigate the different learning strategies with a spotlight on loss functions. Therefore, topics like metric learning [26] and general adversarial networks will be excluded. Also, other applications like semantic segmentation [24] or other image sources like videos or sketches [54] are excluded.

1.2. Approach

The rest of the paper is structured in the following way. We will summarize common ideas that reappear in a broad range of methods. Furthermore, we will define three different training strategies to separate and structure the presented methods. All methods are presented with a short description and a reference to the common ideas. We compare these methods regarding their common ideas and performance. Based on this comparison we will discuss what trends and research opportunities arise.

2. Underlying Concepts

Throughout this survey, we use the terms training strategy, common idea and method in a specific meaning. The *training strategy* is the general type/approach of an algorithm. For further details see subsection 2.2. We call each algorithm proposed in a paper *method*. All methods are described in section 3. A method follows a training strategy and is based on several *common ideas*. We use the term common idea, or in short idea, for concepts and approaches that are shared between different methods. See subsection 2.1 for further information.

In the rest of this chapter, we will use a shared definition for the following variables. We define X_l and X_u as arbitrary sets of labeled and unlabeled images, respectively. For an image $x \in X_l$ the corresponding label is defined as $z_x \in Z$. All images form the set $X = X_l \cup X_u$. Let C be the number of classes for the labels Z. For a given neural network f and input $x \in X$ the output of the neural network is $f(x)$. For the below-defined formulations, f is an arbitrary network with arbitrary weights and parameters.

2.1. Common ideas

Different common ideas are used to train models in semi-, self- and unsupervised learning. In this section, we present a selection of these ideas that are used across multiple methods in the literature.

It is important to notice that our usage of common ideas is fuzzy and incomplete by definition. A common idea should not be an identical implementation or approximation but the underlying motivation. This fuzziness is needed for two reasons. Firstly, a comparison would not be possible due to so many small differences in the exact implementations. Secondly, they allow us to abstract some core elements of a method and therefore similarities can be detected. Also, not all details, concepts and motivations are captured by common ideas. As the name says only common usage qualifies an idea to be considered in this survey. We will limit ourselves to the common ideas described below due to the fact that new common ideas will arise, old ones will disappear and focus will shift to other ideas.

We sorted the ideas in alphabetical order. Since ideas might reference each other, you may have to jump to the corresponding entry if you would like to know more.

Cross-entropy (CE)
A common loss function for image classification is cross-entropy. It is commonly used to measure the difference between $f(x)$ and the corresponding label z_x for a given $x \in X_l$.

$$\begin{aligned} CE(f(x), z) &= \sum_{c=1}^{C} P_{f(x)}(c) log(P_z(c)) \\ &= H(P_z) + KL(P_z | P_{f(x)}) \end{aligned} \tag{1}$$

P is a probability distribution over all classes. H is the entropy of a probability distribution and KL is the Kullback-Leibler divergence. The distribution P can be approximated with the output of the neural network $f(x)$ or the given label z. It is important to note that cross-entropy is the sum of entropy over z and a Kullback-Leibler divergence between $f(x)$ and z. In general, the entropy $H(P_z)$ is zero due to the one-hot encoded label z.

Entropy Minimization (EntMin)
Grandvalet and Bengio proposed to sharpen the output predictions in semi-supervised learning by minimizing entropy [18]. They minimized the entropy $H(P_{f(x)})$ for all probability distributions $P_{f(x)}$ based on a certain neural output $f(x)$ for an image $x \in X$. This minimization only sharpens the predictions of a neural network and cannot be used on its own. If it would be used as a loss the predictions would degenerate.

Fine-tuning
Fine-tuning describes the improvement of a pretrained network on new or different labels. A pretrained network is a network in which the randomly initialized weights were already trained in some way. This training could be achieved for example in a supervised way on a different, large and generic dataset like ImageNet [29] or in an unsupervised way with a pretext task. A great variety of papers have shown that this knowledge transfer can benefit the generalisability and performance [43]. We distinguish between CE and fine-tuning to show the difference in using CE with other losses in one training stage or solely after a previous training stage.

Kullback-Leibler divergence (KL)
The Kullback-Leiber divergence is also commonly used in image classification due to the fact that it can be interpreted as a part of cross-entropy. In general, KL measures the difference between two given distributions and is therefore often used to define an auxiliary loss between the output for an image $x \in X$ and a given discrete probability distribution P over the classes C.

$$KL(f(x), P) = \sum_{c=1}^{C} P_{f(x)}(c) log(\frac{P_{f(x)}(c)}{P(c)}) \quad (2)$$

Mean Squared Error (MSE)
A distance measure between two neural network outputs $f(x), f(y)$ for images $x, y \in X$ is MSE. Instead of measuring the difference based on probability theory it uses the euclidean distance of the output vectors

$$MSE(f(x), f(y)) = ||f(x) - f(y)||_2^2 \quad (3)$$

The minimization of this measure can contract two outputs to each other. This distance measure can also be used on any intermediate output (feature space) of $f(x)$ and $f(y)$.

Mixup
Mixup creates convex combinations of images by blending them into each other. An illustration of the concept is given in Figure 2. The prediction of the convex combination of the corresponding labels turned out to be beneficial for supervised learning in general [58].

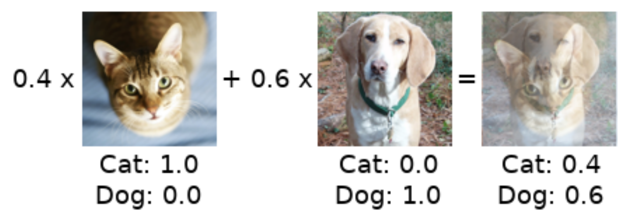


Figure 2: Illustration of mixup – The images of a cat and a dog are combined with a parametrized blending. The labels are also combined with the same parameterization. The shown images are taken from the dataset STL-10 [10]

Mutual Information (MI)
MI is defined for two probability distributions as the Kullback Leiber (KL) divergence between the joint distribution and the marginal distributions [11]. The loss function of several methods [22, 24, 2] is mainly motivated by this in contrast to CE. It is important to notice that the concrete usage of mutual information is different between these methods due to calculation issues. MI is not always easily calculable and different approximations have to be used. For further theoretical insights and one approximation see [3]. We sketch one example usage for images $x, y \in X$ and the corresponding probability distributions $P_{f(x)}, P_{f(y)}$ to illustrate the benefits of MI. We can maximize the mutual

information by minimizing the following:

$$-I(P_{f(x)}, P_{f(y)}) = -KL(P_{(f(x),f(y))}|P_{f(x)} * P_{f(y)}) = -H(P_{f(x)}) + H(P_{f(x)}|P_{f(y)}) \tag{4}$$

An alternative representation of mutual information is the separation in entropy $H(P_{f(x)})$ and conditional entropy $H(P_{f(x)}|P_{f(y)})$.
Ji et al. describe the benefits of using MI over CE in unsupervised cases [24]. One major benefit is the inherent property to avoid degeneration due to the separation in entropy and conditional entropy. MI balances the effects of maximizing the entropy with an uniform distribution for $P_{f(x)}$ and minimizing the conditional entropy by equalizing $P_{f(x)}$ and $P_{f(y)}$. Both cases lead on their own to a degeneration of a neural network.

Overclustering
Normally, if we have k classes in the supervised case we also use k clusters in the unsupervised case. Research showed that it can be beneficial to use more clusters than actual classes k exist [6, 24]. We call this idea *overclustering*.

Overclustering can be beneficial in semi-supervised or unsupervised cases due to the effect that neural networks can decide 'on their own' how to split the data. This separation can be helpful in noisy data or with intermediate classes that were sorted into adjacent classes randomly.

Pretext Task
A pretext task is a broad-ranged description of training a neural network on a different task before training on the actual task. No labels are used or the pretext task itself generates auxiliary target information for training. This task can be for example predicting the rotation of an image [17] or solving a jigsaw puzzle [38]. These pretext tasks are often used to learn representations. A small trained network is then fine-tuned based on these representations. In a semi-supervised context, some methods use this pretext task to define an additional loss during training [4].

Pseudo-Labels
A simple approach for estimating labels of unknown data is using Pseudo-Labels [31]. Lee proposed to classify unseen data with a neural network and use the predictions as labels. What sounds at first like a self-fulfilling assumption works reasonably well in real-world image classification tasks. Several modern methods are based on the same core idea of creating labels by predicting them on their own [45, 5].

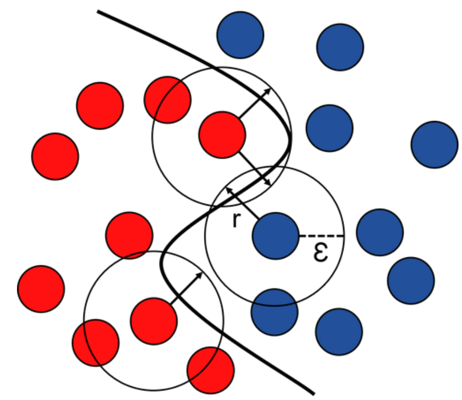
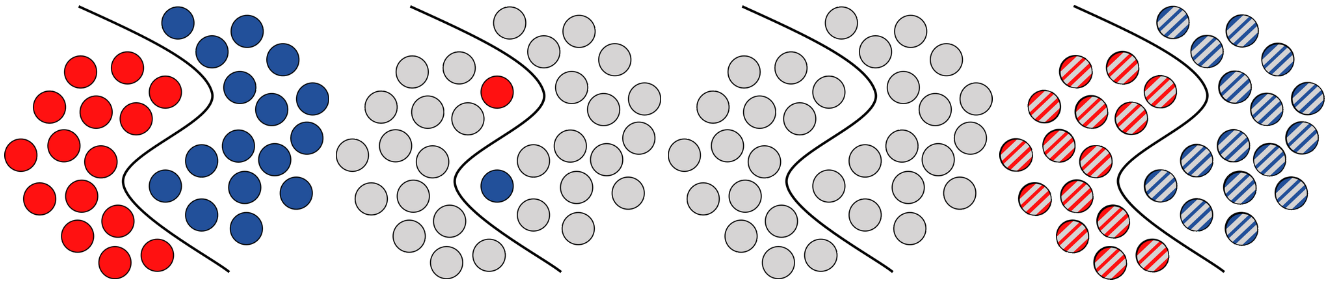


Figure 3: Illustration of the VAT concept - The blue and red circles represent two different classes. The line is the decision boundary between these classes. The ϵ spheres around the circles define the area of possible transformations. The arrows represent the adversarial change r which push the decision boundary away from any data point.

Virtual Adversarial Training (VAT)
VAT [37] tries to make predictions invariant to small transformations by minimizing the distance between an image and a transformed version of the image. Miyato et al. showed how a transformation can be chosen and approximated in an adversarial way. This adversarial transformation maximizes the distance between an image and a transformed version of it over all possible transformations. Figure 3 illustrates the concept of VAT. The loss is defined as

$$VAT(f(x)) = D(P_{f(x)}, P_{f(x+r_{adv})}) \\ r_{adv} = argmax_{r;||r|| \leq \epsilon} D(P_{f(x)}, P_{f(x+r)}) \tag{5}$$

In this equation, x is an image out of the dataset X and $f(x)$ is the output for a given neural network. P is the probability distribution over these outputs and D is a non-negative function that measures the distance. Two examples of used distance measures are cross-entropy [37] and Kullback-Leiber divergence [57].



(a) Supervised (b) One-Stage-Semi-Supervised (c) One-Stage-Unsupervised (d) Multi-Stage-Semi-Supervised

Figure 4: Illustrations of supervised learning and the three presented reduced training strategies - The red and dark blue circles represent labeled data points of different classes. The light grey circles represent unlabeled data points. The black lines define the underlying decision boundaries between the classes. The striped circles represent data points that ignore and use the label information at different stages of the training process.

2.2. Training strategies

Terms like supervised, semi-supervised and self-supervised are often used in literature. A precise definition that clearly separates semi-, self- and unsupervised is rarely given. In most cases a rough consensus about the meaning is sufficient but we noticed a high variety of definitions in borderline cases. Even if precise and well structure taxonomies are given [41], future research comes up with new combinations that were not thought of before. We consider it impossible to create the perfect taxonomy which can include all future approaches.

Nevertheless, we need to structure the methods in some way to keep an overview, allow comparison and acknowledge the difference of research foci. Therefore, we call all semi-, self- and unsupervised (learning) strategies together *reduced supervised* (learning) strategies. We separate and compare the methods using the above mentioned common ideas and a simple separation based on the stages during training.

We classify approaches with reduced supervision into One-Stage-Semi-Supervised, One-Stage-Unsupervised, Multi-Stage-Semi-Supervised. In the following paragraphs, we will define and describe each training strategy. Moreover, we will shortly cover the common supervised learning strategy. We will also mention what kind of reduced learning strategies are common for these training strategies to give a better understanding. Figure 4 illustrates the supervised and the three presented reduced supervised strategies.

We will see later in subsection 4.3 that this separation leads to a clear clustering of the methods regarding the common ideas. This clustering shows the benefit of the below-defined training strategies further.

2.2.1 Supervised Learning

Supervised learning is the most common strategy in image classification with deep neural networks. Only the labeled data X_l and its corresponding labels Z are used. The goal is to minimize a loss function between the output of the network $f(x)$ and the expected label $z_x \in Z$ for all $x \in X_l$. We do not distinguish this learning strategy by the number of training stages in comparison to the following training strategies.

2.2.2 One-Stage-Semi-Supervised Training

All methods which follow the one-stage-semi-supervised training strategy are trained in one training procedure with the usage of X_l, X_u and Z. For simplicity, we treat pretraining a network for example on ImageNet as weight initialization instead of counting it as a separate stage. Due to this the main difference to many supervised learning strategies is the usage of the additional unlabeled data X_u. A common way to integrate the unlabeled data is to add one or many unsupervised losses to the supervised loss. Many authors call their methods therefore semi-supervised.

2.2.3 One-Stage-Unsupervised Training

All methods which follow the one-stage-unsupervised training strategy are trained in one training procedure with the usage of only the unlabeled samples X_u. Therefore, many authors in this training strategy call their method unsupervised. A variety of loss functions exist for unsupervised learning [7, 24, 52]. In most cases, the problem is rephrased in such a way that all inputs for the loss can be generated, e.g. reconstruction loss in autoencoders [52]. Due to this self-supervision, some call these methods also self-supervised. We want to point out one major difference to many self-supervised methods following the multi-stage-semi-supervised training strategy below. One-Stage-Unsupervised methods give image classifications without any further usage of labeled data.

2.2.4 Multi-Stage-Semi-Supervised Training

All methods which follow the multi-stage-semi-supervised training strategy are trained in at least two training procedures with the usage of X_l, X_u and the corresponding labels Z. Many methods that are called self-supervised by their authors fall into this strategy. Commonly a pretext task is used to learn representations on unlabeled data X_u. In the second stage, these representations are fine-tuned to image classification on X_l. There are a variety of multi-stage-semi-supervised methods that have more than two stages or use the different datasets in a mixed way. An important difference to a one-stage method is that these methods return useable classifications only after an additional training stage.

3. Methods

In the following, we give a short overview of all methods in this survey in alphabetical order and separated by their training strategy. Since they may reference each other, you may have to jump to the corresponding entry if you would like to know more. This list does not claim to be complete. We included methods that were referenced often in related work, which are comparable to the other methods and which are complementary to the presented methods.

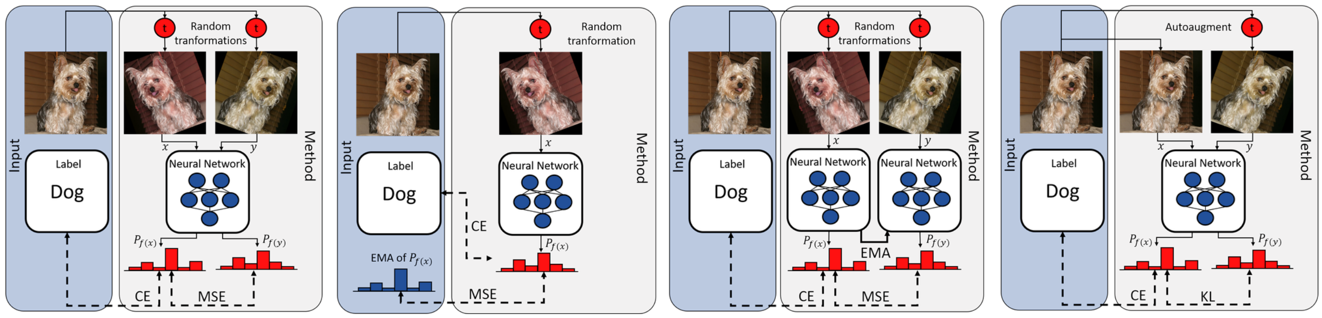
3.1. One-Stage-Semi-Supervised

Ensemple AutoEndocing Transformation (EnAET)
EnAET [50] combines the self-supervised pretext task AutoEncoding Transformations [59] with MixMatch [5]. Wang et al. use spatial transformations, such as translations and rotations, and non-spatial transformations, such as color distortions, on input images. The transformations are estimated with the original and augmented image given in contrast to other pretext tasks where the estimation is often based on only the augmented image. The loss is used together with the loss of MixMatch and is extended with the Kullback Leiber divergence between the predictions of the original and the augmented image.

Interpolation Consistency Traning (ICT)
ICT [49] uses linear interpolations of unlabeled data points to regularize the consistency. Verma et al. use a combination of supervised cross-entropy and an unsupervised loss. The unsupervised loss is MSE between the prediction for the interpolation of two images and the interpolation of their Pseudo-Labels. The interpolation is generated with the mixup [58] algorithm from two unlabeled data points. For these unlabeled data points, the Pseudo-Labels are predicted by a Mean Teacher [45] network.

Fast-Stochastic Weight Averaging (fast-SWA)
In contrast to other semi-supervised methods Athiwaratkun et al. do not change the loss but the optimization algorithm [1]. They analyzed the learning process based on ideas and concepts of SWA [23], π-model [30] and Mean Teacher [45]. Athiwaratkun et al. show that averaging and cycling learning rates are beneficial in semi-supervised learning by stabilizing the training. They call their improved version of SWA fast-SWA due to faster convergence and lower performance variance [1]. The architecture and loss is either copied from π-model [30] or Mean Teacher [45].

Mean Teacher
With Mean Teacher Tarvainen & Valpola present a student-teacher-approach for semi-supervised learning [45]. They develop their approach based on the π-model and Temporal Ensembling [30]. Therefore, they also use MSE as a consistency loss between two predictions but create these predictions differently. They argue that Temporal Ensembling incorporates new information too slowly into predictions. The reason for this is that the exponential moving average (EMA) is only updated once per epoch. Therefore, they propose



(a) π-model (b) Temporal Ensembling (c) Mean Teacher (d) UDA

Figure 5: Illustration of four selected one-stage-semi-supervised methods – The used method is given below each image. The input is given in the blue box on the left side. On the right side, an illustration of the method is provided. The process is organized in general from top to bottom. At first, the input images are preprocessed by none or two different random transformations. Autoaugment [12] is a special augmentation technique. The following neural network uses these preprocessed images (x, y) as input. The calculation of the loss (dotted line) is different for each method but shares common parts. All methods use the cross-entropy (CE) between label and predicted distribution $P_{f(x)}$ on labeled examples. All methods also use a consistency regularization between different predicted output distributions $(P_{f(x)}, P_{f(y)})$. The creation of these distributions differ for all methods and the details are described in the corresponding entry in section 3. EMA stands for exponential moving average. The other abbreviations are defined above in subsection 2.1.

to use a teacher based on average weights of a student in each update step. Tarvainen & Valpola show for their model that the KL-divergence is an inferior consistency loss than MSE. An illustration of this method is given in Figure 5.

MixMatch
MixMatch [5] uses a combination of a supervised and an unsupervised loss. Berthelot et al. use CE as the supervised loss and MSE between predictions and generated Pseudo-Labels as their unsupervised loss. These Pseudo-Labels are created from previous predictions of augmented images. They propose a novel sharping method over multiple predictions to improve the quality of the Pseudo-Labels. Furthermore, they extend the algorithm mixup [58] to semi-supervised learning by incorporating the generated labels.

π-model and Temporal Ensembling
Laine & Aila present two similar learning methods with the names π-model and Temporal Ensembling [30]. Both methods use a combination of the supervised CE loss and the unsupervised consistency loss MSE. The first input for the consistency loss in both cases is the output of their network from a randomly augmented input image. The second input is different for each method. In the π-model an augmentation of the same image is used. In Temporal Ensembling an exponential moving average of previous predictions is evaluated. Laine & Aila show that Temporal Ensembling is up to two times faster and more stable in comparison to the π-model [30]. Illustrations of these methods are given in Figure 5.

Pseudo-Labels
Pseudo-Labels [31] describes a common technique in deep learning and a learning method on its own. For the general technique see above in subsection 2.1. In contrast to many other semi-supervised methods, Pseudo-Labels does not use a combination of an unsupervised and a supervised loss. The Pseudo-Labels approach uses the predictions of a neural network as labels for unknown data as described in the general technique. Therefore, the labeled and unlabeled data are used in parallel to minimize the CE loss.

ReMixMatch
ReMixMatch [4] is an extension of MixMatch with

distribution alignment and augmentation anchoring. Berthelot et al. motivate the distribution alignment with an analysis of mutual information. While they use entropy minimization via ”sharpening” they do not use any prediction equalization like in mutual information. They argue that an equal distribution is also not desirable since the distribution of the unlabeled data could be skewed. Therefore, they align the predictions of the unlabeled data with a marginal class distribution over the seen examples. Berthelot et al. exchange the augmentation scheme of MixMatch with augmentation anchoring. Instead of averaging the prediction over different slight augmentations of an image they only use stronger augmentations as regularization. All augmented predictions of an image are encouraged to result in the same distribution with CE instead of MSE. Furthermore, a self-supervised loss based on the rotation pretext task [17] was added.

Unsupervised Data Augmentation (UDA)
Xie et al. present with UDA a semi-supervised learning algorithm which concentrates on the usage of state-of-the-art augmentation [53]. They use a supervised and an unsupervised loss. The supervised loss is CE while the unsupervised loss is the Kullback Leiber divergence between output predictions. These output predictions are based on an image and an augmented version of this image. For image classification, they propose to use the augmentation scheme generated by AutoAugment [12] in combination with Cutout [13]. AutoAugment uses reinforcement learning to create useful augmentations automatically. Cutout is an augmentation scheme where randomly selected regions of the image are masked out. Xie et al. show that this combined augmentation method achieves higher performance in comparison to previous methods on their own like Cutout, Cropping or Flipping. In addition to the different augmentation, they propose to use a variety of other regularization methods. They proposed Training Signal Annealing which restricts the influence of labeled examples during the training process to prevent overfitting. They use EntMin [18] and a kind of Pseudo-Labeling [31]. We use the term kind of Pseudo-Labeling because they do not use the predictions as labels but they use them to filter unsupervised data for outliers. An illustration of this method is given in Figure 5.

Virtual Adversarial Training (VAT)
VAT [37] is not just the name for a comon idea but it is also a one-stage-semi-supervised method. Miyato et al. used a combination of VAT on unlabeled data and CE on labeled data [37]. They showed that the adversarial transformation leads to a lower error on image classification than random transformations. Furthermore, they showed that adding EntMin [18] to the loss increased accuracy even more.

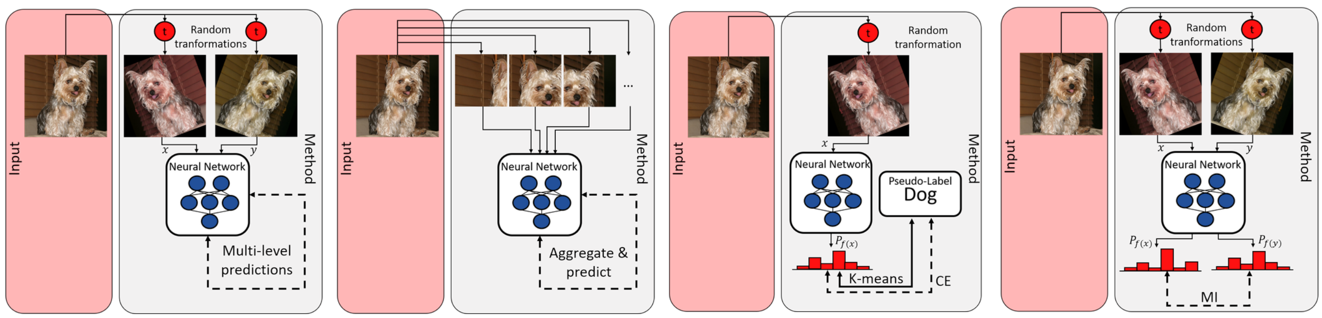
3.2. One-Stage-Unsupervised

Deep Adaptive Image Clustering (DAC)
DAC [7] reformulates unsupervised clustering as a pairwise classification. Similar to the idea of Pseudo-Labels Chang et al. predict clusters and use these to retrain the network. The twist is that they calculate the cosine distance between all cluster predictions. This distance is used to determine whether the input images are similar or dissimilar with a given certainty. The network is then trained with binary CE on these certain similar and dissimilar input images. During the training process, they lower the needed certainty to include more images. As input Chang et al. use a combination of RGB and extracted HOG features. Additionally, they use an auxiliary loss in their source code which is not reported in the paper.[1]

Invariant Information Clustering (IIC)
IIC [24] is described below as a multi-stage-semi-supervised method. In comparison to other presented methods, IIC creates usable classifications without fine-tuning the model on labeled data. The reason for this is that the pretext task is constructed in such a way that label predictions can be extracted directly from the model. This leads to the conclusion that IIC can also be interpreted as an unsupervised learning method.

Information Maximizing Self-Augmented Training (IMSAT)
IMSAT [22] maximizes MI between the input and output of the model. As a consistency regularization Hu et al. use CE between an image prediction and an augmented image prediction. They show that the best augmentation of the prediction can be calculated with VAT [37]. The maximization of MI directly on the image

[1] https://github.com/vector-1127/DAC/blob/master/STL10/stl.py



(a) AMDIM (b) CPC (c) DeepCluster (d) IIC

Figure 6: Illustration of four selected multi-stage-semi-supervised methods – The used method is given below each image. The input is given in the red box on the left side. On the right side, an illustration of the method is provided. The fine-tuning part is excluded. The process is organized in general from top to bottom. At first, the input images are either preprocessed by one or two random transformations or are split up. The following neural network uses these preprocessed images (x, y) as input. The calculation of the loss (dotted line) is different for each method. AMDIM and CPC use internal elements of the network to calculate the loss. DeepCluster and IIC use the predicted output distribution $(P_{f(x)}, P_{f(y)})$ to calculate a loss. For further details see the corresponding entry in section 3.

input leads to a problem. For datasets like CIFAR-10, CIFAR-100 [28] and STL-10 [10] the color information is too dominant in comparison to the actual content or shape. As a workaround Hu et al. use the features generated by a pretrained CNN on ImageNet [29] as input.

3.3. Multi-Stage-Semi-Supervised

Augmented Multiscale Deep InfoMax (AMDIM)
AMDIM [2] maximizes the MI between inputs and outputs of a network. It is an extension of the method DIM [21]. DIM usually maximizes MI between local regions of an image and a representation of the image. AMDIM extends the idea of DIM in several ways. Firstly, the authors sample the local regions and representations from different augmentations of the same source image. Secondly, they maximize MI between multiple scales of the local region and the representation. They use a more powerful encoder and define mixture-based representations to achieve higher accuracies. Bachman et al. fine-tune the representations on labeled data to measure their quality. An illustration of this method is given in Figure 6.

Constrastive Multiview Coding (CMC)
CMC [46] generalizes CPC [47] to an arbitrary collection of views. Tian et al. try to learn an embedding that is different for contrastive samples and equal for similar images. Like Oord et al. they train their network by identifying the correct prediction out of multiple negative ones [47]. However, Tian et al. take different views of the same image such as color channels, depth and segmentation as similar images. For common image classification datasets like STL-10 they use patch-based similarity. After this pretext task, the representations are fine-tuned to the desired dataset.

Contrastive Predictive Coding (CPC)
CPC [47, 20] is a self-supervised method that predicts representations of local image regions based on previous image regions. The authors determine the quality of these predictions by identifying the correct prediction out of randomly sampled negative ones. They use the CE loss and adopt the loss to the summation over the complete image. This adaption results in their loss InfoNCE [47]. Van den Oord et al. showed that minimizing InfoNCE maximizes the lower bound for MI between the previous image regions and the predicted image region [47]. An illustration of this method is given in Figure 6.

DeepCluster
DeepCluster [6] is a self-supervised method that gen-

erates labels by k-means clustering. Caron et al. iterate between clustering of predicted labels to generate Pseudo-Labels and training with cross-entropy on these labels. They show that it is beneficial to use overclustering in the pretext task. After the pretext task, they fine-tune the network on all labels. An illustration of this method is given in Figure 6.

Deep InfoMax (DIM)
DIM [21] maximizes the MI between local input regions and output representations. Hjelm et al. show that maximizing over local input regions rather than the complete image is beneficial for image classification. In addition, they use a discriminator to match the output representations to a given prior distribution. In the end, they fine-tune the network with an additional small fully-connected neural network.

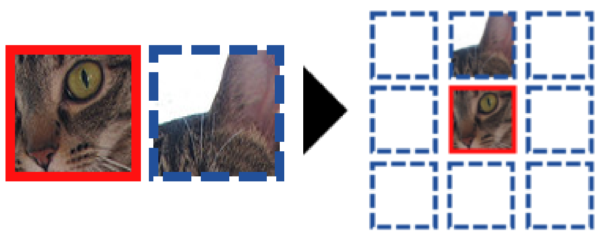


Figure 7: Illustration of the Context pretext task – A central patch and an adjacent patch from the same image are given. The task is to predict one of the 8 possible relative positions of the second patch to the first one. In the example, the correct answer is upper center. The illustration is inspired by [14].

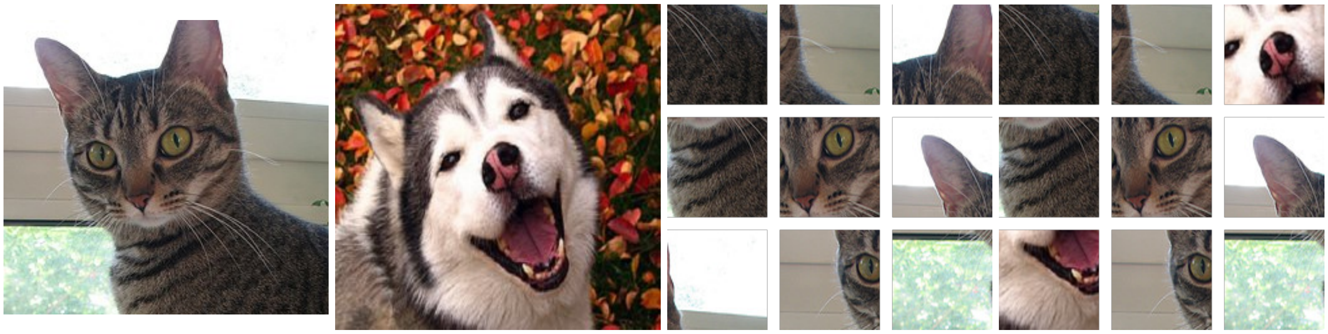
Deep Metric Transfer (DMT)
DMT [32] learns a metric as a pretext task and then propagates labels onto unlabeled data with this metric. Liu et al. use self-supervised image colorization [60] or unsupervised instance discrimination [51] to calculate a metric. In the semi-supervised case, they propagate labels to unlabeled data with spectral clustering. In the end, the network is finetuned with the new Pseudo-Labels. Additionally, they show that their approach is complementary to previous methods. If they use the most confident Pseudo-Labels for methods such as Mean Teacher [45] or VAT [37], they can improve the accuracy with very few labels by about 30%.

Invariant Information Clustering (IIC)
IIC [24] maximizes the MI between augmented views of an image. The idea is that images should belong to the same class regardless of the augmentation. The augmentation has to be a transformation to which the neural network should be invariant. The authors do not maximize directly over the output distributions but over the class distribution which is approximated for every batch. Ji et al. use auxiliary overclustering on a different output head to increase their performance in the unsupervised case. This idea allows the network to learn subclasses and handle noisy data. Ji et al. use Sobel filtered images as input instead of the original RGB images. Additionally, they show how to extend IIC to image segmentation. Up to this point, the method is completely unsupervised. To be comparable to other semi-supervised methods they fine-tune their models on a subset of available labels. An illustration of this method is given in Figure 6.

Representation Learning - Context
Doersch et al. propose to use context prediction as a pretext task for visual representation learning [14]. A central patch and an adjacent patch from an image are used as input. The task is to predict one of the 8 possible relative positions of the second patch to the first one. An illustration of the pretext task is given in Figure 7. Doersch et al. argue that this task becomes easier if you recognize the content of these patches. The authors fine-tune their representations for other tasks and show their superiority in comparison to random initialization. Aside from fine-tuning, Doersch et al. show how their method could be used for Visual Data Mining.

Representation Learning - Exemplar
Dosovitskiy et al. were one of the first to propose a self-supervised pretext task with additional fine-tuning [15]. They randomly sample patches from different images and augment these patches heavily. Augmentations can be for example rotations, translations, color changes or contrast adjustments. The classification task is to map all augmented versions of a patch to the correct original patch.

Representation Learning - Jigsaw
Noroozi and Favaro propose to solve Jigsaw puzzles as



(a) main image (b) different image (c) Jigsaw (d) Jigsaw++

Figure 8: Illustrations of the pretext task Jigsaw and Jigsaw++ – The Jigsaw pretext task consists of solving a simple Jigsaw puzzle generated from the main image. Jigsaw++ augments the Jigsaw puzzle by adding in parts of a different image. The illustrations are inspired by [39].

a pretext task [38]. The idea is that a network has to understand the concept of a presented object to solve the puzzle. They prevent simple solutions that only look at edges or corners by including small random margins between the puzzle patches. They fine-tune on supervised data for image classification tasks. Noroozi et al. extended the Jigsaw task by adding image parts of a different image [39]. They call the extension Jigsaw++. An example of Jigsaw and Jigsaw++ is given in Figure 8.

Representation Learning - Rotation
Gidaris et al. use a pretext task based on image rotation prediction [17]. They propose to randomly rotate the input image by 0, 90, 180 or 270 degrees and let the network predict the chosen rotation degree. In their work, they also evaluate different numbers of rotations but four rotations score the best result. For image classification, they fine-tune on labeled data.

Simple Framework for Contrastive Learning of Visual Representation (SimCLR)
SimCLR [8] maximizes the agreement between two different augmentations of the same image. The method is similiar to CPC [47] and IIC [24]. In comparison to CPC Chen et al. do not use the different inner representations. Contrary to IIC they use normalized temperature-scaled cross-entropy (NT-Xent) as their loss.

Based on the cosine similarity of the predictions NT-Xent measures if positive pairs are similar and negative pairs are dissimilar. Augmented versions of the same image are treated as positive pairs and pairs with any other image as negative pair. The system is trained with large batch sizes of up to 8192 instead of a memory bank to create enough negative examples.

Self-Supervised Semi-Supervised Learning (S^4L)
S^4L [57] is, as the name suggests, a combination of self-supervised and semi-supervised methods. Zhai et al. split the loss in a supervised and an unsupervised part. The supervised loss is CE while the unsupervised loss is based on the self-supervised techniques using rotation and exemplar prediction [17, 15]. The authors show that their method performs better than other self-supervised and semi-supervised techniques [15, 17, 37, 18, 31]. In their *Mix Of All Models* (MOAM) they combine self-supervised rotation prediction, VAT, entropy minimization, Pseudo-Labels and fine-tuning into a single model with multiple training steps. Since we discuss the results of their MOAM we identify S^4L as a multi-stage-semi-supervised method.

4. Comparison

In this chapter, we will analyze which common ideas are shared or differ between methods. We will compare the performance of all methods with each other on common deep learning datasets.

(a) CIFAR-10 (b) STL-10 (c) ILSVRC-2012

Figure 9: Examples of four random cats in the different datasets to illustrate the difference in quality

4.1. Datasets

In this survey, we compare the presented methods on a variety of datasets. We selected four datasets that were used in multiple papers to allow a fair comparison. An overview of example images is given in Figure 9.

CIFAR-10 and CIFAR-100 are large datasets of tiny color images with size 32x32 [28]. Both datasets contain 60,000 images belonging to 10 or 100 classes respectively. The 100 classes in CIFAR-100 can be combined into 20 superclasses. Both sets provide 50,000 training labels and 10,000 validation labels. The presented results are only trained with 4,000 labels for CIFAR-10 and 10,000 labels for CIFAR-100 to represent a semi-supervised case. If a method uses all labels this is marked independently.

STL-10 is dataset designed for unsupervised and semi-supervised learning [10]. The dataset is inspired by CIFAR-10 [28] but provides fewer labels. It only consists of 5,000 training labels and 8,000 validation labels. However, 100,000 unlabeled example images are also provided. These unlabeled examples belong to the training classes and some different classes. The images are 96x96 color images and were acquired in combination with their labels from ImageNet [29].

ILSVRC-2012 is a subset of ImageNet [29]. The training set consists of 1.2 million images while the validation and the test set includes 150,000 images. These images belong to 1000 object categories. Due to this large number of categories, it is common to report Top-5 and Top-1 accuracy. Top-1 accuracy is the classical accuracy where one prediction is compared to one ground-truth label. Top-5 accuracy checks if a ground truth label is in a set of at most five predictions. For further details on accuracy see subsection 4.2. The presented results are only trained with 10% of labels to represent a semi-supervised case. If a method uses all labels this is marked independently.

4.2. Evaluation metrics

We compare the performance of all methods based on their classification score. This score is defined differently for unsupervised and all other settings. We follow standard protocol and use the classification accuracy in most cases. For unsupervised learning, we use cluster accuracy because we need to handle the missing labels during the training. We need to find the best one-to-one permutations from the network cluster predictions to the ground-truth classes.

For vectors $x, y \in \mathbb{Z}^N$ with $N \in \mathbb{N}$ the accuracy is defined as follows:

$$ACC(x, y) = \frac{\sum_{i=1}^{N} \mathbb{1}_{y_i = x_i}}{N} \tag{6}$$

For the cluster accuracy we additionally maximize over all possible one-to-one permutations σ.

$$ACC(x, y) = \max_{\sigma} \frac{\sum_{i=1}^{N} \mathbb{1}_{y_i = \sigma(x_i)}}{N} \tag{7}$$

4.3. Comparison of methods

In this subsection, we will compare the methods concerning their used common ideas and performance. We will summarize the presented results and discuss the underlying trends in the next subsection.

Comparison with regard to used common ideas

In Table 1 we present all methods and their used common ideas. Following our definition of common ideas in subsection 2.1 we evaluate only ideas that were used frequently in different papers. Special details such as the different optimizer for fast-SWA or the used approximation for MI are excluded. Please see section 3 for further details.

One might expect that common ideas are used equally between methods and training strategies. We rather see a tendency that common ideas differ between training strategies. We will step through all common ideas based on the significance of differentiating the training strategies.

A clear separation between one-stage-semi-supervised and multi-stage-semi-supervised training can be based on CE, fine-tuning and pretext tasks. All one-stage-semi-supervised methods use a cross-entropy loss during training while only two use additional losses based on pretext tasks. All multi-stage-semi-supervised methods use a pretext task and fine-tune in the end while only some methods use CE before the last stage. All one-stage-semi-supervised methods use no fine-tuning. The supervision from ground-truth labels in one-stage-semi-supervised and multi-stage-semi-supervised is solely introduced by using CE and fine-tuning. Due to our definition of the training strategies this grouping is expected.

However, further clusters of the common ideas are visible. We notice that some common ideas are (almost) solely used by one strategy. These common ideas are EntMin, KL, MSE and Mixup for one-stage-semi-supervised methods and MI and overclustering for multi-stage-semi-supervised methods. We hypothesize that this shared and different usage of ideas exists due the different usage of unlabeled data. On-stage-semi-supervised methods use the unlabeled and labeled data in the same stage. This encourages the usage of different ideas than the multi-stage-sem-supervised strategy. For example mixup interpolates labels but in many multi-stage-semi-supervised methods the labeled data is only used in an additional stage.

If we compare multi-stage-semi-supervised and one-stage-unsupervised training we notice that MI, overclustering and pretext tasks are used often in both. All three of them are not used often with one-stage-semi-supervised training as stated above. We hypothesize that this similarity arises because most multi-stage-semi-supervised methods have an unsupervised stage followed by a supervised stage. For the method IIC the authors even proposed to fine-tune the unsupervised method to surpass purely supervised results.

Pseudo-Labels and VAT are used in several different methods. Due to their simple and complementary idea, they can be used in a variety of different methods. UDA for example uses this technique to filter the unlabeled data for useful images.

All in all, we see that the defined training strategies share common ideas inside each strategy and differ in the usage of ideas between them. We conclude that the definition of the training strategies is not only logical sensible but is also support by their usage of common ideas.

We compared the training strategies but what about the individual algorithms? We want to highlight EnAET, ReMixMatch, S^4L which stand out between these methods. They all use more than 5 common ideas or in general use a broad range of ideas. This also supported by the fact that they use uncommon ideas with regard to their training strategy. EnAET and ReMixMatch both use an additional loss based on a pretext task while S^4L uses supervision not only pure fine-tuning stage. A possible implication on performance is given in the next sections.

Comparison with regard to performance

We compare the performance of the different methods based on their respective reported results or cross-references in other papers. For better comparability, we would have liked to recreate every method in a unified setup but this was not feasible. While using reported values might be the only possible approach, it leads to drawbacks in the analysis.

Kolesnikov et al. showed that changes in the architecture can lead to significant performance boost or drops [27]. They state that 'neither [...] the ranking of architectures [is] consistent across differ-

Table 1: Overview of the methods and their used common ideas — On the left-hand side, the reviewed methods from section 3 are sorted by the training strategy. The top row lists the common ideas. Details about the ideas and their abbreviations are given in subsection 2.1. The last column and some rows sum up the usage ideas per method or per training strategy. *Legend*: * An idea which uses labels from the ground-truth Z. (X) The idea is only used indirectly. The individual explanations are given by the indicated number. [1] MixMatch does entropy minimization implicitly by sharpening the predictions [5]. [2] ReMixMatch uses CE as a supervised loss between output and label and as a self-supervised loss between outputs of augmented images [4]. [3] ReMixMatch motivates the distribution alignment with MI [4]. [4] UDA predicts Pseudo-Labels for filtering the unsupervised data. [5] Minimize mutual information objective as a pretext task e.g. between views [2] or layers [20]. [6] InfoNCE and NT-Xent are based on CE [47, 8]. [7] The loss InfoNCE maximizes the mutual information indirectly [47]. [8] Deep Cluster uses K-Means to calculate Pseudo-Labels and optimizes the assignment as a pretext task. [9] DMT learns a metric for label propagation as a pretext task [32]. [10] SimCLR uses the normalized temperature-scaled cross-entropy loss between different augmentations as pretext task [8]. [11] DAC uses the cosine distance between elements to estimate similar and dissimilar items. One could say DAC creates Pseudo-Labels for the similarity problem.

	CE	EntMin	Fine-tuning	KL	MSE	Mixup	MI	Over-Clustering	Pretext Task	Pseudo-Labels	VAT	*Overall Sum*
One-Stage-Semi-Supervised												
EnAET [50]	X*	$(X)^1$		X	X	X			AET	X		7
ICT [49]	X*				X	X				X		4
fast-SWA [1]	X*				X							2
Mean Teacher [45]	X*				X							2
MixMatch [5]	X*	$(X)^1$			X	X				X		5
π model [30]	X*				X							2
Pseudo-Labels [31]	X*									X		2
ReMixMatch [4]	X* / $(X)^2$	$(X)^1$				X	$(X)^3$		Rotation	X		6
Temporal Ensembling [30]	X*				X							2
UDA [53]	X*	X		X						$(X)^4$		4
VAT [37]	X*										X	2
VAT + EntMin [37]	X*	X									X	3
Sum	12	5	0	2	7	4	1	0	2	6	2	41
Multi-Stage-Semi-Supervised												
AMDIM [2]			X*				X		$(X)^5$			3
Context [14]			X*						Context			2
CMC [46]	$(X)^6$		X*				$(X)^7$		$(X)^5$			4
CPC [47, 20]	$(X)^6$		X*				$(X)^7$		$(X)^5$			4
DeepCluster [6]	X		X*					X	$(X)^8$	$(X)^8$		5
DMT [32]			X*		X				$(X)^9$	X		4
DIM [21]			X*				X		$(X)^5$			3
Exemplar [15]			X*						Augmentation			2
IIC [24]			X*				X	X	$(X)^5$			4
Jigsaw [38]			X*						Jigsaw			2
Rotation [17]			X*						Rotation			2
SimCLR [8]	$(X)^6$		X*						$(X)^{10}$			3
S^4L [57]	X*	X	X*						Rotation	X	X	6
Sum	5	1	13	0	1	0	5	2	13	3	1	44
One-Stage-Unsupervised												
DAC [7]										$(X)^{11}$		1
IIC [24]							X	X	$(X)^5$			3
IMSAT [22]							X				X	2
Sum	0	0	0	0	0	0	2	1	1	1	1	6
Overall Sum	17	6	13	2	8	4	8	3	16	10	4	91

ent methods, nor is the ranking of methods consistent across architectures' [27]. While most methods try to achieve comparability with previous ones by a similar setup, over time small differences still aggregate and lead to a variety of used architectures. Some methods use only early convolutional networks such as AlexNet [29] but others use more modern architectures like Wide ResNet-Architecture [56] or Shake-Shake-Regularization [16].

Oliver et al. proposed guidelines to ensure more comparable evaluations in semi-supervised learning [40]. They showed that not following these guidelines may lead to changes in the performance [40]. While some methods try to follow these guidelines, we cannot guarantee that all methods do so. This impacts comparability further. Considering the above-mentioned limitations, we do not focus on small differences but look for general trends and specialities instead.

Table 2 shows the collected results for all presented methods. We also provide results for the respective supervised baselines reported by the authors. To keep fair comparability we did not add state-of-the-art baselines with more complex architectures. Table 3 shows the results for even fewer.

In general, the used architectures become more complex and the accuracies rise over time. This behavior is expected as new results are often improvements of earlier works. The changes in architecture may have led to these improvements. However, many papers include ablation studies and comparisons to only supervised methods to show the impact of their method. We believe that a combination of more modern architecture and more advanced methods lead to improvements.

For the CIFAR-10 dataset, almost all multi- or one-stage-semi-supervised methods reach about or over 90% accuracy. The best methods MixMatch and ReMixMatch reach an accuracy of about 95% and are roughly two percent worse than the fully supervised baseline. For the CIFAR-100 dataset, fewer results are reported. MixMatch is with about 74% on this dataset the best method in comparison to the fully supervised baseline of about 80%. Newer methods provide also results for 1000 or even 250 labels instead of 4000 labels. Especially EnAET and ReMixMatch stick out since they achieve only 1-2% worse results with 250 labels instead of with 4000 labels.

For the STL-10 dataset, most methods report a better result than the supervised baseline. These results are possible due to the unlabeled part of the dataset. The unlabeled data can only be utilized by semi-, self- or unsupervised methods. EnAET achieves the best results with about 95%. ReMixMatch reports an accuracy of about 94% with only 1000 labels. This is more than most methods achieve with 5000 labels.

The ILSVRC-2012 dataset is the most difficult dataset based on the reported Top-1 accuracies. Most methods achieve only a Top-1 accuracy which is roughly 20% worse than the reported supervised baseline with around 79%. Only the methods SimCLR, S^4L and UDA achieve an accuracy that is less than 10% worse than the baseline. SimCLR achieves the best accuracy with a Top-1 accuracy of about 76.5% and a Top-5 accuracy of around 93%. For fewer used labels only two results are reported. Therefore, a comparison is difficult. A Top-5 accuracy of about 86% with only 1% of the labels from the method SimCLR sounds promising.

The unsupervised methods are separated from the supervised baseline by a clear margin of up to 50%. IIC achieves the best results of about 61% on CIFAR-10 and STL-10. IMSAT reports an accuracy of about 94% on STL-10. Since IMSAT uses pretrained ImageNet features, a superset of STL-10, the results are not directly comparable.

4.4. Discussion

In this subsection, we discuss the presented results of the previous subsection. We divide our discussion into three major trends which we identified. All these trends lead to possible future research opportunities.

1. Trend: Real World Applications

Previous methods were not scaleable to real-world images and applications and used workarounds e.g. extracted features [22] to process real-world images. Many methods can report a result of over 90% on CIFAR-10, a simple low-resolution dataset. Only three methods are able to achieve a Top-5 accuracy of over 90% on ILSVRC-2012, a high-resolution dataset. We conclude that most methods are not scalable to real-world image classification problems. However, the best-reported methods like ReMixMatch, SimCLR

Table 2: Overview of the reported accuracies — The first column states the used method. For the supervised baseline, we used the best-reported results which were considered as baselines in the referenced papers. The original paper is given in brackets after the score. The architecture and their reference are given in the second column. The third column shows the year of publication or the release year of the preprint. The last four columns report the Top-1 accuracy score in % for the respective dataset (See subsection 4.2 for further details). If the results are not reported in the original paper, the reference is given after the result. A blank entry represents the fact that no result was reported. *Legend*: * Top-5 accuracy instead of Top-1 is reported. † 100% of the labels are used instead of the default value defined in subsection 4.1. ‡ Multilayer perceptron used for fine-tuning instead of one fully connected layer. Remarks on special architectures and evaluations: [1] Architecture includes Shake-Shake regularization. [2] Network uses wider hidden layers. [3] Method uses ten random classes out of the default 1000 classes. [4] Network only predicts 20 superclasses instead of the default 100 classes. [5] Inputs are pretrained ImageNet features. [6] Method uses different copies of the network for each input.

	Architecture	Publication	CIFAR-10	CIFAR-100	STL-10	ILSVRC-2012
Supervised (100% labels)	Best reported	–	97.14[45]	79.82[2]	68.7 [21]	78.57 [57] / 94.10* [57]
One-Stage-Semi-Supervised						
EnAET [50]	Wide ResNet-28 [40]	2019	94.65	73.07	95.48	
ICT [49]	Wide ResNet-28 [40]	2019	92.34			
ICT [49]	CONV-13 [30]	2019	92.71			
fast-SWA [1]	CONV-13 [30]	2019	90.95	66.38		
fast-SWA [1]	ResNet-26[1] [16]	2019	93.72			
Mean Teacher [45]	CONV-13 [30]	2017	87.69			
Mean Teacher [45]	Wide ResNet-28 [40]	2017	89.64			
MixMatch [5]	Wide ResNet-28 [40]	2019	95.05	74.12	94.41	
π model [30]	CONV-13 [30]	2017	87.64			
Pseudo-Label [31]	ResNet50v2 [19]	2013				82.41* [57]
ReMixMatch [4]	Wide ResNet-28 [40]	2019	94.86		93.82	
Temporal Ensembling [30]	CONV-13 [30]	2017	87.84			
UDA [53]	Wide ResNet-28 [40]	2019	94.7			68.66 / 88.52*
VAT [37]	CONV-13 [30]	2018	88.64			
VAT [37]	ResNet50v2 [19]	2018				82.78* [57]
VAT [37] + EntMin [18]	CONV-13 [30]	2018	89.45 [37]			
VAT [37] + EntMin [18]	ResNet50v2 [19]	2018	86.41 [57]			83.39* [57]
Multi-Stage-Semi-Supervised						
AMDIM [2]	ResNet18 [19]	2019	91.3† / 93.6†‡	70.2† / 73.8†‡	93.6 / 93.8‡	60.2† / 60.9†‡
Context [14]	ResNet50 [19]	2015				51.4† [27]
CMC [46]	AlexNet [29]	2019			86.88‡	
CMC [46]	ResNet-50[6] [19]	2019				70.6/89.7*
CPC [47, 20]	ResNet-170 [20]	2019	77.45† [21]		77.81† [21]	61.0 / 84.88*
DeepCluster [6]	AlexNet [29]	2018			73.4 [24]	41†
DMT [32]	Wide ResNet-28 [40]	2019	88.70			
DIM [21]	AlexNet [29]	2019			72.57‡	
DIM [21]	GAN Discriminator [42]	2019	75.21†‡	49.74†‡		
Exemplar [15]	ResNet50 [19]	2016				46.0† [27] / 81.01* [57]
IIC [24]	ResNet34 [19]	2019			88.8‡	
Jigsaw [38]	AlexNet [29]	2016				44.6† [27]
Rotation [17]	AlexNet [29]	2018				55.4† [27]
Rotation [17]	ResNet50v2 [19]	2018				78.53* [57]
SimCLR [8]	ResNet50v2[2] [27]	2020				76.5† / 93.2†* / 92.6*
S^{4}L [57]	ResNet50v2[2] [27]	2019				73.21 / 91.23*
One-Stage-Unsupervised						
DAC [7]	All-ConvNet [44]	2017	52.18	23.75	46.99	52.72[3]
IIC [24]	ResNet34 [19]	2019	61.7	25.7[4]	61.0	
IMSAT [22]	Autoencoder[5]	2017	45.6	27.5	94.1	

Table 3: Overview of the reported accuracies with fewer labels - The first column states the used method. The last seven columns report the Top-1 accuracy score in % for the respective dataset and amount of labels. The number are either given in absolute numbers or as percent. A blank entry represents the fact that no result was reported. *Legend*: * Top-5 accuracy instead of Top-1 is reported.

	CIFAR-10			STL-10		ILSVRC-2012	
	4000	1000	250	5000	1000	10%	1%
One-Stage-Semi-Supervised							
EnAET [50]	94.65	93.05	92.4	95.48	91.96		
ICT [49]	92.71	84.52	61.4 [5]				
Mean Teacher [45]	89.64	82.68	52.68				
MixMatch [5]	93.76	92.25	88.92	94.41	89.82		
ReMixMatch [4]	94.86	94.27	93.73		93.82		
Multi-Stage-Semi-Supervised							
DMT [32]	88.70		80.3				58.6
SimCLR [8]						$92.6^\star$	$85.8^\star$

and S^4L surpassed the point of only scientific usage and can be applied to real-world applications.

This conclusion applies to real-world image classification tasks with balanced and clearly separated classes. This conclusion also implicates which real-world issues need to be solved in future research. Class imbalance or noisy labels are not treated by the presented methods. Datasets with also few unlabeled data points are not considered.

2. Trend: Required supervision is decreasing

We see that the gap between reduced supervised and supervised methods is shrinking. For CIFAR-10, CIFAR-100 and ILSVRC-2012 we have a gap of less than 5% left between total supervised and reduced supervised learning. For STL-10 the reduced supervised methods even surpass the total supervised case by about 30% due to the additional set of unlabeled data. We conclude that reduced supervised learning reaches comparable results while using only roughly 10% of the labels.

A lot of newly proposed methods are semi- or self-supervised in comparison to unsupervised ones. Unsupervised methods like IIC still reach results of over 60% and show that this kind of training can be beneficial for semi-supervised learning [24]. However, the results are still surpassed by semi- or self-supervised methods by a large margin e.g. over 30% on CIFAR-10. The integration of the knowledge of some labels into the training process seems to be crucial.

In general, we considered a reduction from 100% to 10% of all labels. However, we see that methods like ReMixMatch and SimCLR achieve comparable results with even fewer labels such as the usage of 1% of all labels. For ILSVRC-2012 this is equivalent to about 13 images per class. We expect that future research will concentrate on achieving comparable results for only 1% or even fewer of all labels. In the end research fields like few-shot, single-shot and semi-supervised learning might even merge.

We assume that in parallel to the reduction of required labels, the usage of solely unsupervised methods will decrease further. However, they might be included into multi-stage methods as initialization. The benefit of even some labels as a guiding reference in form of archetypes for many real-world applications is important. This will lead to a shift in the corresponding research efforts.

3. Trend: Combination of common ideas

In the comparison, we identified that few common ideas are shared by one-stage-semi-supervised and multi-stage-semi-supervised methods.

We believe there is only little overlap between these methods due to the different aims of the respective authors. Many multi-stage-semi-supervised papers focus on creating good representations. They fine-tune their results only to be comparable. One-stage-semi-supervised papers aim for the best accuracy scores with as few labels as possible.

The comparison showed that EnAET, ReMixMatch, S^{4}L, SimCLR are the best methods or, if we consider the above-mentioned limitations due to architecture differences, one of the best. Three methods out of these stood out in the comparison of common ideas. They used a broad range of ideas and ideas uncommon for their respective training strategy. S^4L calls their combined approach even "Mix of all models" [57]. We assume that this combination is one reason for their superior performance. This assumption is supported by the included comparisons in the original papers. For example S^{4}L showed the impact of each method separately as well as the combination of all [57].

IIC is the only method that can be used as an unsupervised or self-supervised method with fine-tuning. This flexibility allows approaches with a smooth transition between no and small supervision.

The comparison showed that some ideas such as Pseudo-Labels can be applied to a variety of methods. However, only a few methods use this idea.

We identified that some common ideas are not often combined and that the combination of broad range and unusual methods is beneficial. We believe that the combination of different common idea is a promising future research field because many reasonable combinations are yet not explored.

5. Conclusion

In this paper, we provided an overview of semi-, self- and unsupervised methods. We analyzed their difference, similarities and combinations based on 25 different methods. This analysis led to the identification of several trends and possible research fields.

We based our analysis on the definition of the different training strategies and common ideas in these strategies. We showed how the methods work in general, which ideas they use and provide a simple classification. Despite the difficult comparison of the methods' performances due to different architectures and implementations, we identified three major trends.

Results of over 90% Top-5 accuracy on ILSVRC-2012 with only 10% of the labels show that semi-supervised methods are applicable to real-world problems. However, issues like class imbalance are not considered. Future research has to address these issues.

The performance gap between supervised and semi- or self-supervised methods is closing. For one dataset it is even surpassed by about 30%. The number of labels to get comparable results to fully supervised learning is decreasing. Future research could lower the number of required labels even further. We noticed that, as time progresses, unsupervised methods are used less often. These two conclusions lead us to the assumption that unsupervised methods will lose significance for real-world image classification in the future.

We concluded that one-stage-semi-supervised and multi-stage-semi-supervised training mainly use a different set of common ideas. Both strategies use a combination of different ideas but there are few overlaps in these techniques. EnAET, ReMixMatch and S^{4}L are the only presented method which gap this separation. We identified the trend that a combination of different techniques is beneficial to the overall performance. In combination with the small overlap between the ideas, we identified possible future research opportunities.

References

[1] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2019. 7, 15, 17

[2] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019. 4, 10, 15, 17

[3] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540, 2018. 4

[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, 2019. 5, 8, 15, 17, 18

[5] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019. 5, 7, 8, 15, 17, 18

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 5, 10, 15, 17

[7] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, 2017. 7, 9, 15, 17

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 12, 15, 17, 18

[9] V. Cheplygina, M. de Bruijne, and J. P. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280296, May 2019. 2

[10] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011. 4, 10, 13

[11] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 4

[12] 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*, pages 113–123, 2019. 8, 9

[13] T. Devries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *ArXiv*, abs/1708.04552, 2017. 9

[14] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015. 11, 15, 17

[15] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):17341747, Sep 2016. 11, 12, 15, 17

[16] X. Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 16, 17

[17] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 5, 9, 12, 15, 17

[18] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 4, 9, 12, 17

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 17

[20] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. *Proceedings of the37thInternational Conference on MachineLearning*, 2020. 10, 15, 17

[21] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. 10, 11, 15, 17

[22] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1558–1567. JMLR. org, 2017. 4, 9, 15, 16, 17

[23] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights

leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 2018. 7

[24] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019. 2, 3, 4, 5, 7, 9, 11, 12, 15, 17, 18

[25] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey, 2020. 2

[26] M. Kaya and H. Ş. Bilge. Deep metric learning: a survey. *Symmetry*, 11(9):1066, 2019. 2, 3

[27] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019. 14, 16, 17

[28] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 10, 13

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 4, 10, 13, 16, 17

[30] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 7, 8, 15, 17

[31] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013. 5, 8, 9, 12, 15, 17

[32] B. Liu, Z. Wu, H. Hu, and S. Lin. Deep metric transfer for label propagation with limited annotated data. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct 2019. 11, 15, 17, 18

[33] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. 2

[34] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. *Lecture Notes in Computer Science*, page 185201, 2018. 1

[35] A. Mey and M. Loog. Improvability through semi-supervised learning: A survey of theoretical results. *arXiv preprint arXiv:1908.09574*, 2019. 3

[36] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018. 2

[37] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 5, 9, 11, 12, 15, 17

[38] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 5, 12, 15, 17

[39] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018. 12

[40] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018. 16, 17

[41] G.-J. Qi and J. Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods, 2019. 2, 6

[42] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 17

[43] L. Schmarje, C. Zelenka, U. Geisen, C.-C. Glüer, and R. Koch. 2d and 3d segmentation of uncertain local collagen fiber orientations in shg microscopy. In *DAGM GCPR*, pages 374–386, 2019. 1, 4

[44] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *ICLR (workshop track)*, 2015. 17

[45] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 5, 7, 11, 15, 17, 18

[46] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding, 2019. 10, 15, 17

[47] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2018. 10, 12, 15, 17

[48] J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, pages 1–68, 2019. 2

[49] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Aug 2019. 7, 15, 17, 18

[50] X. Wang, D. Kihara, J. Luo, and G.-J. Qi. Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning. *arXiv preprint arXiv:1911.09265*, 2019. 7, 15, 17, 18

[51] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 11

[52] J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016. 2, 7

[53] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. Unsupervised data augmentation for consistency training, 2019. 2, 9, 15, 17

[54] P. Xu, Z. Song, Q. Yin, Y.-Z. Song, and L. Wang. Deep self-supervised representation learning for free-hand sketch, 2020. 3

[55] R. Xu and D. C. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16:645–678, 2005. 2

[56] S. Zagoruyko and N. Komodakis. Wide residual networks. *Procedings of the British Machine Vision Conference 2016*, 2016. 16

[57] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019. 2, 5, 12, 15, 17, 19

[58] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 4, 7, 8

[59] L. Zhang, G.-J. Qi, L. Wang, and J. Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 7

[60] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 11

[61] X. Zhu. Semi-supervised learning literature survey. *Comput Sci, University of Wisconsin-Madison*, 2, 07 2008. 2