IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 19, 2021, accepted May 18, 2021, date of publication May 27, 2021, date of current version June 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3084358

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

LARS SCHMARJE, MONTY SANTAROSSA, SIMON-MARTIN SCHRÖDER, AND REINHARD KOCH, (Member, IEEE)
Multimedia Information Processing Group, Kiel University, 24118 Kiel, Germany

Corresponding author: Lars Schmarje (las@informatik.uni-kiel.de)

This work was supported by Land Schleswig-Holstein through the Open Access Publikationsfonds Funding Program.

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 34 methods in detail based on their performance and their commonly used ideas rather than a fine-grained taxonomy. In our analysis, we identify three major trends that lead to future research opportunities. 1. State-of-the-art methods are scalable to real-world applications in theory but issues like class imbalance, robustness, or fuzzy labels are not considered. 2. The degree of supervision which is needed to achieve comparable results to the usage of all labels is decreasing and therefore methods need to be extended to settings with a variable number of classes. 3. All methods share some common ideas but we identify clusters of methods that do not share many ideas. We show that combining ideas from different clusters can lead to better performance.

INDEX TERMS Semi-supervised, self-supervised, unsupervised, image classification, deep learning, survey.

I. 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 [1]–[3], object detection [4], [5] or semantic segmentation [6], [7].

The quality of a deep neural network is strongly influenced by the number of labeled/supervised images [8]. ImageNet [1] 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 [9]. 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 [10]. This might be a practical workaround for some applications but the fundamental issue remains: Unlike humans, supervised learning needs enormous amounts of labeled data.

For a given problem we often have access to a large dataset of unlabeled data. How this unsupervised data could be used for neural networks has been of research interest for many years [11]. Xie *et al.* were among the first in 2016 to investigate unsupervised deep learning image clustering strategies to leverage this data [12]. Since then, the usage of unlabeled data has been researched in numerous ways and has created research fields like unsupervised, semi-supervised, self-supervised, weakly-supervised, or metric learning [13]. Generally speaking, unsupervised learning uses no labeled data, semi-supervised learning uses unlabeled and labeled while self-supervised learning generates labeled data on its own. Other research directions are even more different because weakly-supervised learning uses only partial information about the label and metric learning aims at learning a good distance metric. 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

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

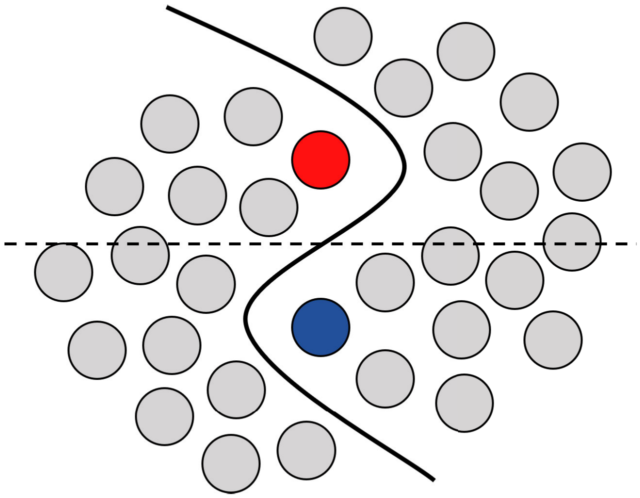


FIGURE 1. This image illustrates and simplifies the benefit of using unlabeled data during deep learning training. The red and dark blue circles represent labeled data points of different classes. The light grey circles represent unlabeled data points. If we have only a small number of labeled samples 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.

training with fewer labels more robust or in some rare cases even surpasses the supervised cases [14].

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 [15], [16] 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.

Whereas 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.

A. 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, surveys, and books has been published on this topic [17]–[21]. Unsupervised cluster algorithms were researched before the breakthrough of deep learning and are still widely used [22]. There are already extensive surveys that describe unsupervised and semi-supervised strategies without deep learning [18], [23]. We will focus only on techniques including deep neural networks.

Many newer surveys focus only on self-, semi- or unsupervised learning [19], [20], [24]. Min *et al.* wrote an overview of unsupervised deep learning strategies [24]. 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 on only one architecture which Min *et al.* refer to as ''Clustering deep neural network (CDNN)-based deep clustering'' [24]. Even though the work was published in 2018, it already misses the recent and major developments 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 [20]. They cover some recent developments but deep learning strategies such as [14], [25]–[28] 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 [19]. Like us, the authors provide a performance comparison and a taxonomy. Their taxonomy distinguishes between different kinds of pretext tasks. 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 [29]. 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 special details. In their survey, Cheplygina *et al.* present a variety of methods in the context of medical image analysis [30]. 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 [31]. We keep our survey limited to general image classification tasks and focus on their practical application.

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. We concentrate on recent methods because older one are already adequately addressed in previous literature [17]–[21]. Keeping the above-mentioned limitations in mind, the topic of self-, semi-, and unsupervised learning still includes a broad

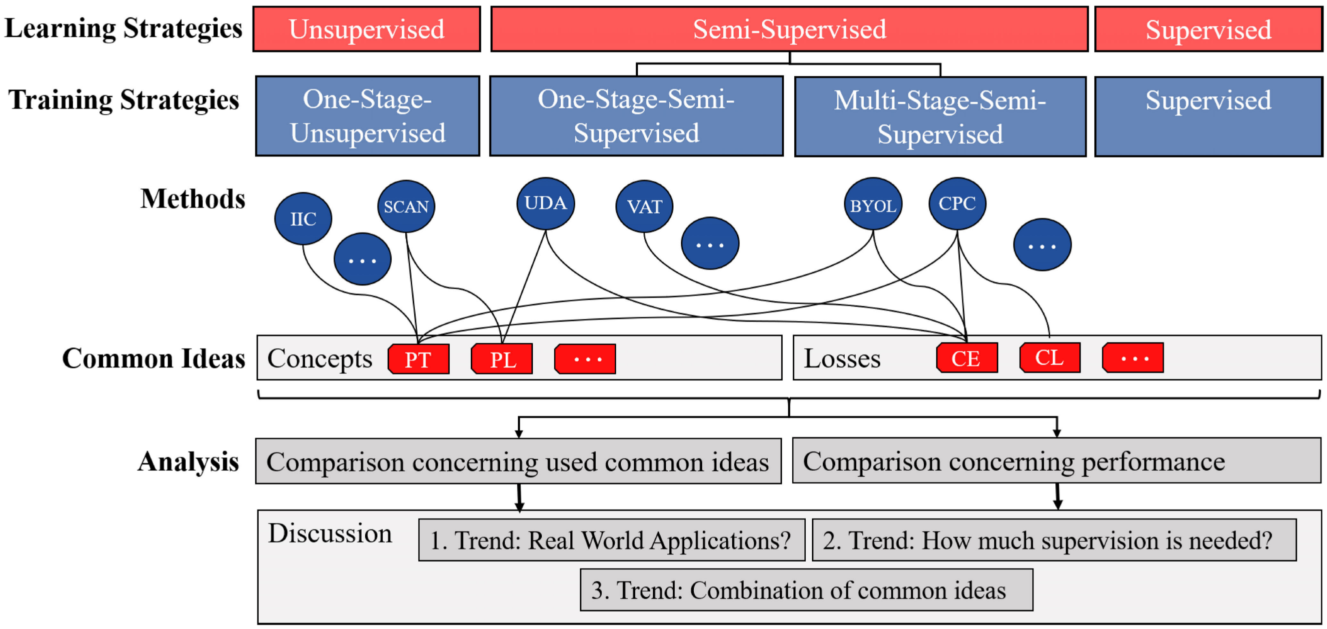

Learning Strategies
Unsupervised
Semi-Supervised
Supervised
Training Strategies
One-Stage-Unsupervised
One-Stage-Semi-Supervised
Multi-Stage-Semi-Supervised
Supervised
Methods
IIC
SCAN
UDA
VAT
BYOL
CPC
...
...
...
Common Ideas
Concepts
PT
PL
...
Losses
CE
CL
...
Analysis
Comparison concerning used common ideas
Comparison concerning performance
Discussion
1. Trend: Real World Applications?
2. Trend: How much supervision is needed?
3. Trend: Combination of common ideas

FIGURE 2. Overview of the structure of this survey – The learning strategies unsupervised, semi-supervised and supervised are commonly used in the literature. Because semi-supervised learning is incorporating many methods we defined training strategies which subdivides semi-supervised learning. For details about the training and learning strategies (including self-supervised learning) see subsection II-A. Each method belongs to one training strategy and uses several common ideas. A common idea can be a concept such as a pretext task or a loss such as cross-entropy. The definition of methods and common ideas is given in section II. Details about the common ideas are defined in subsection II-B. All methods in this survey are shortly described and categorized in section III. The methods are compared with each other based on this information concerning their used common ideas and their performance in subsection IV-C. The results of the comparisons and three resulting trends are discussed in subsection IV-D.

range of research fields. We have to exclude some related topics from this survey to keep the focus of this work for example because other research have a different aim or are evaluated on different datasets. Therefore, topics like metric learning [13] and meta learning such as [32] will be excluded. More specific networks like general adversarial networks [33] and graph networks such as [34] will be excluded. Also, other applications like pose estimation [35] and segmentation [36] or other image sources like videos or sketches [37] are excluded. Topics like few-shot or zero-shot learning methods such as [38] are excluded in this survey. However, we will see in subsection IV-D that topics like few-shot learning and semi-supervised can learn from each other in the future like in [39].

B. OUTLINE

The rest of the paper is structured in the following way. We define and explain the terms which are used in this survey such as method, training strategy and common idea in section II. A visual representation of the terms and their dependencies can be seen before the analysis part in Figure 2. All methods are presented with a short description, their training strategy and common idea in section III. In section IV, we compare the methods based their used ideas and their performance across four common image classification datasets. This section also includes a description of the datasets and evaluation metrics. Finally, we discuss the results of the comparisons in subsection IV-D and identify three trends and research opportunities. In Figure 2, a complete overview of the structure of this survey can be seen.

II. 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 for using the unsupervised data during training. The training strategies are similar to the terms semi-supervised, self-supervised, or unsupervised learning but provide a definition for corner cases that the other terms do not. We will explain the differences and similarities in detail in subsection II-A. The papers we discuss in detail in this survey propose different elements like an algorithm, a general idea, or an extension of previous work. To be consistent in this survey, we call the main algorithm, idea, or extension in each paper a *method*. All methods are briefly described in section III. 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. We roughly sort the methods based on their training strategy but compare them in detail based on the used common ideas. See subsection II-B for further information about common ideas.

In the rest of this chapter, we will use a shared definition for the following variables. For an arbitrary set of images X we define X_l and X_u with $X = X_l \dot{\cup} X_u$ as the labeled and unlabeled images, respectively. For an image $x \in X_l$ the corresponding label is defined as $z_x \in Z$. An image

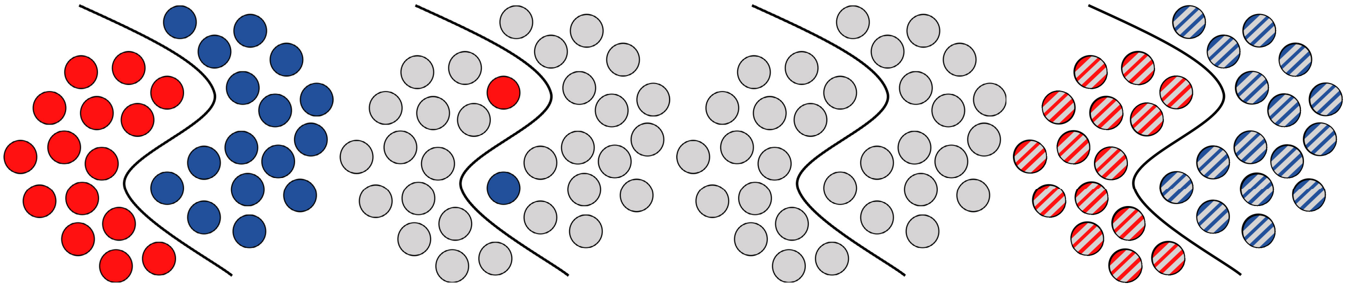

(a) Supervised (b) One-Stage-Semi-Supervised (c) One-Stage-Unsupervised (d) Multi-Stage-Semi-Supervised

FIGURE 3. Illustrations of supervised learning (a) and the three presented reduced training strategies (b-d) - 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 do not use the label information in the first stage and can access this information in a second stage. For more details on stages and the different learning strategies see subsection II-A.

$x \in X_u$ has no label otherwise it would belong to X_l. For the distinction between X_u and X_l, only the usage of the label information during training is important. For example, an image $x \in X$ might have a label that can be used during evaluation but as long as the label is not used during training we define $x \in X_u$. The learning strategy LS_X for a dataset X is either unsupervised ($X = X_u$), supervised ($X = X_l$) or semi-supervised ($X_u \cap X_l \neq \emptyset$). During different phases of the training, different image datasets $X_1, X_2, \ldots X_n$ with $n \in \mathbb{N}$ could be used. Two consecutive datasets X_i and X_{i+1} with $i \leq n$ and $i \in \mathbb{N}$ are different as long as different images ($X_i \neq X_{i+1}$) or different labels ($X_{L_i} \neq X_{L_{i+1}}$) are used. The learning strategy LS_i up to the dataset X_i during the training is calculated based on $X_u = \cup_{j=1}^{i} X_{u_j}$ and $X_l = \cup_{j=1}^{i} X_{l_j}$. Consecutive phases of the training are grouped into *stages*. The stage changes during consecutive datasets X_i and X_{i+1} iff the learning strategy is different ($LS_{X_i} \neq LS_{X_{i+1}}$) and the overall learning strategy changes ($LS_i \neq LS_{i+1}$). Due to this definition, only two stages can occur during training and the seven possible combinations are visualized in Figure 4. For more details see subsection II-A. 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.

A. TRAINING STRATEGIES

Terms like semi-supervised, self-supervised, and unsupervised learning are often used in literature but have overlapping definitions for certain methods. We will summarize the general understanding and definition of these terms and highlight borderline cases that are difficult to classify. Due to these borderline cases, we will define a new taxonomy based on the stages during training for a precise distinction of the methods. In subsection IV-C, we will see that this taxonomy leads to a clear clustering of the methods regarding the common ideas which further justifies this taxonomy. A visual comparison between the learning-strategies semi-supervised and unsupervised learning and the training strategies can be found in Figure 4.

Unsupervised learning describes the training without any labels. However, the goal can be a clustering (e.g. [14], [27]) or good representation (e.g. [25], [40]) of the data. Some methods combine several unsupervised steps to achieve firstly a good representation and then a clustering (e.g. [41]). In most cases, this unsupervised training is achieved by generating its own labels, and therefore the methods are called self-supervised. A counterexample for an unsupervised method without self-supervision would be k-means [22]. Often, self-supervision is achieved on a pretext task on the same or a different dataset and then the pretrained network is fine-tuned on a downstream task [19]. Many methods that follow this paradigm say their method is a form of representation learning [25], [40], [42]–[44]. In this survey, we focus on image classification, and therefore most self-supervised or representation learning methods need to fine-tune on labeled data. The combination of pretraining and fine-tuning can neither be called unsupervised nor self-supervised as external labeled information are used. Semi-supervised learning describes methods that use labeled and unlabeled data. However, semi-supervised methods like [16], [26], [45]–[49] use the labeled and unlabeled data from the beginning in comparison to representation learning methods like [25], [40], [42]–[44] which use them in different stages of their training. Some methods combine ideas from self-supervised learning, semi-supervised learning and unsupervised learning [15], [27] and are even more difficult to classify.

From the above explanation, we see that most methods are either unsupervised or semi-supervised in the context of image classification. The usage of labeled and unlabeled data in semi-supervised methods varies and a clear distinction in the common taxonomy is not obvious. Nevertheless, we need to structure the methods in some way to keep an overview, allow comparisons and acknowledge the difference of research foci. We decided against providing a fine-grained taxonomy as in previous literature [29] because we believe future research will come up with new combinations that were not thought of before. We separate the methods only based on a rough distinction when the labeled or unlabeled data is used during the training. For detailed comparisons, we distinct the methods based on their common ideas that are defined above and described in detail in subsection II-B. We call all semi-, self-, and unsupervised (learning) strategies together *reduced supervised* (learning) strategies.

We defined *stages* above (see section II) as the different phases/time intervals during training when the different learning strategies supervised ($X = X_l$), unsupervised ($X = X_u$) or semi-supervised ($X_u \cap X_l \neq \emptyset$) are used. For example, a method that uses a self-supervised pretraining on X_u and then fine-tunes on the same images with labels has two stages. A method that uses different algorithms, losses, or datasets during the training but only uses unsupervised data X_u has one stage (e.g. [41]). A method which uses X_u and X_l during the complete training has one stage (e.g. [26]). Based on the definition of stages during training, we classify reduced supervised methods into the training strategies: One-Stage-Semi-Supervised, One-Stage-Unsupervised, and Multi-Stage-Semi-Supervised. An overview of the stage combinations and the corresponding training strategy is given in Figure 4. As we concentrate on reduced supervised learning in this survey, we will not discuss any methods which are completely supervised.

Due to the above definition of stages a fifth combination of data usage between the stages exists. This combination would use only labeled data in the first stage and unlabeled data in the second stage. In the rest of the survey, we will exclude this training strategy for the following reasons. The case that a stage of complete supervision is followed by a stage of partial or no supervision is an unusual training strategy. Due to this unusual usage, we only know of weight initialization followed by other reduced supervised training steps where this combination could occur. We see the initialization of a network with pretrained weights from a supervised training on a different dataset (e.g. Imagenet [1]) as an architectural decision. It is not part of the reduced supervised training process because it is used mainly as a more sophisticated weight initialization. If we exclude weight initialization for this reason, we know of no method which belongs to this stage.

In the following paragraphs, we will describe all other training strategies in detail and they are illustrated in Figure 3.

1) SUPERVISED LEARNING

Supervised learning is the most common strategy in image classification with deep neural networks. These methods only use labeled data X_l and its corresponding labels Z. 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 $x \in X_l$.

2) ONE-STAGE-SEMI-SUPERVISED TRAINING

All methods which follow the one-stage-semi-supervised training strategy are trained in one stage with the usage of X_l, X_u, and Z. The main difference to all 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 more unsupervised losses to the supervised loss.

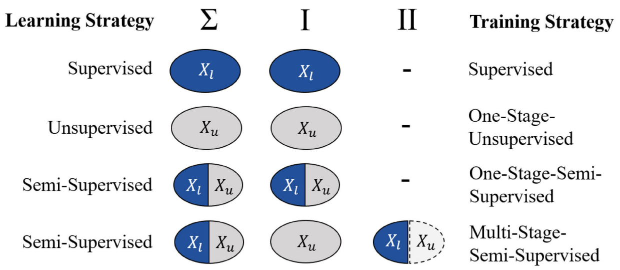


FIGURE 4. Illustration of the different training strategies – Each row stands for a different combination of data usage during the first and second stage (defined in section II). The first column states the common learning strategy name in the literature for this usage whereas the last column states the training strategy name used in this survey. The second column represents the used data overall. The third and fourth column represent the used data in stage one or two. The blue and grey (half-) circles represent the usage of the labeled data X_l and the unlabeled data X_u respectively in each stage or overall. A minus means that no further stage is used. The dashed half circle in the last row represents that this dashed part of the data can be used.

3) ONE-STAGE-UNSUPERVISED TRAINING

All methods which follow the one-stage-unsupervised training strategy are trained in one stage 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 [12], [14], [50]. 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 [12]. Due to this self-supervision, some call also these methods 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.

4) MULTI-STAGE-SEMI-SUPERVISED TRAINING

All methods which follow the multi-stage-semi-supervised training strategy are trained in two stages with the usage of X_u in the first stage and X_l and maybe X_u in the second stage. 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. An important difference to a one-stage method is that these methods return useable classifications only after an additional training stage.

B. 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. We will limit ourselves to the common ideas described below since we believe they are enough to characterize all recent methods. At the same time, we know that these ideas need to be extended in the future as new common ideas will arise, old ones will disappear, and focus will shift to other ideas. In contrast to detailed taxonomies, these new ideas can easily be integrated as new tags.

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

LOSS FUNCTIONS

CROSS-ENTROPY (CE)

A common loss function for image classification is cross-entropy [51]. It is commonly used to measure the difference between $f(x)$ and the corresponding label z_x for a given $x \in X_l$. The loss is defined in Equation 1 and the goal is to minimize the difference.

$$\begin{aligned} CE(z_x, f(x)) &= -\sum_{c=1}^{C} P(c|z_x)log(P(c|f(x))) \\ &= -\sum_{c=1}^{C} P(c|z_x)log(P(c|z_x)) \\ &\quad -\sum_{c=1}^{C} P(c|z_x)log(\frac{P(c|f(x))}{P(c|z_x)}) \\ &= H(P(\cdot|z_x)) \\ &\quad + KL(P(\cdot|z_x) \mid\mid P(\cdot|f(x)) \end{aligned} \tag{1}$$

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

The loss function CE could also be used with a different probability distribution than P based on the ground-truth label. These distributions could be for example be based on Pseudo-Labels or other targets in a self-supervised pretext task. We abbreviate the used common idea with CE* if not the ground-truth labels are used to highlight this specialty.

CONTRASTIVE LOSS (CL)

A contrastive loss tries to distinguish positive and negative pairs. The positive pair could be different views of the same image and the negative pairs could be all other pairwise combinations in a batch [25]. Hadsell *et al.* proposed to learn representations based on contrasting [53]. In recent years, the idea has been extended by self-supervised visual representation learning methods [25], [54]–[57]. Examples of contrastive loss functions are NT-Xent [25] and InfoNCE [55] and both are based on Cross-Entropy. The loss NT-Xent is computed across all positive pairs (x_i, x_j) in a fixed subset of X with N elements e.g. a batch during training. The definition of the loss for a positive pair is given in Equation 2. The similarity *sim* between the outputs is measured with a normalized dot product, τ is a temperature parameter and the batch consists of N image pairs.

$$l_{x_i,x_j} = -log\frac{exp(sim(f(x_i), f(x_j))/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} exp(sim(f(x_i), f(x_k))/\tau)} \tag{2}$$

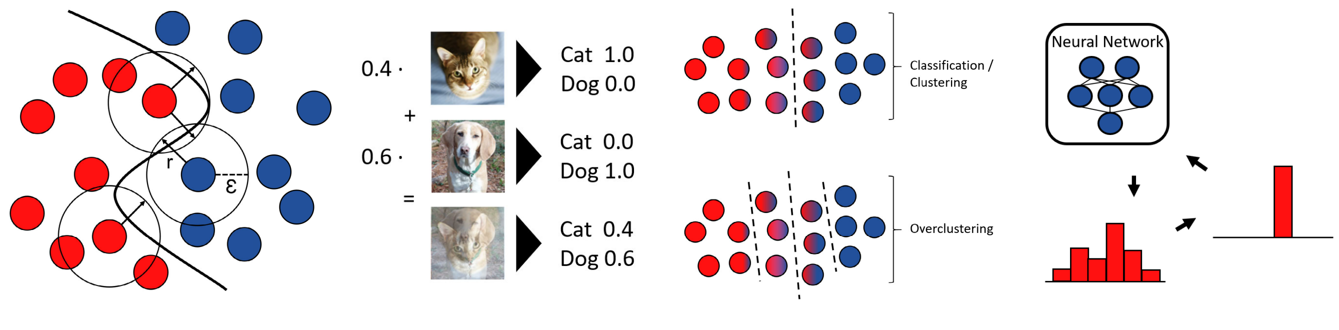
Chen and Li generalize the loss NT-Xent into a broader family of loss functions with an alignment and a distribution part [58]. The alignment part encourages representations of positive pairs to be similar whereas the distribution part "encourages representations to match a prior distribution" [58]. The loss InfoNCE is motivated like other contrastive losses by maximizing the agreement / mutual information between different views. Van der Oord *et al.* showed that InfoNCE is a lower bound for the mutual information between the views [55]. More details and different bounds for other losses can be found in [59]. However, Tschannen *et al.* show evidence that these lower bounds might not be the main reason for the successes of these methods [60]. Due to this fact, we count losses like InfoNCE as a mixture of the common ideas contrastive loss and mutual information.

ENTROPY MINIMIZATION (EM)

Grandvalet and Bengio noticed that the distributions of predictions in semi-supervised learning tend to be distributed over many or all classes instead of being sharp for one or few classes [61]. They proposed to sharpen the output predictions or in other words to force the network to make more confident predictions by minimizing entropy [61]. They minimized the entropy $H(P(\cdot|f(x)))$ for a probability distribution $(P(\cdot|f(x))$ based on a certain neural output $f(x)$ and an image $x \in X$. This minimization leads to sharper / more confident predictions. If this loss is used as the only loss the network/predictions would degenerate to a trivial minimization.

KULLBACK-LEIBLER DIVERGENCE (KL)

The Kullback-Leiber divergence is also commonly used in image classification since it can be interpreted as a part of cross-entropy. In general, KL measures the difference between two given distributions [62] and is therefore often used to define an auxiliary loss between the output $f(x)$ for an image $x \in X$ and a given secondary discrete probability distribution Q over the classes C. The definition is given in Equation 3. The second distribution could be another network output distribution, a prior known distribution, or a ground-truth distribution depending on the goal of



(a) VAT (b) Mixup (c) Overclustering (d) Pseudo-Label

FIGURE 5. Illustration of four selected common ideas – (a) 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 vector *r* which pushes the decision boundary away from any data point. (b) 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 [52] (c) Each circle represents a data point and the coloring of the circle the ground-truth label. In this example, the images in the middle have fuzzy ground-truth labels. Classification can only draw one arbitrary decision boundary (dashed line) in the datapoints whereas overclustering can create multiple subregions. This method could also be applied to outliers rather than fuzzy labels. (d) This loop represents one version of Pseudo-Labeling. A neural network predicts an output distribution. This distribution is cast into a hard Pseudo-Label which is then used for further training the neural network.

the minimization.

$$KL(Q \,||\, P(\cdot|f(x)) = -\sum_{c=1}^{C} Q(c) log(\frac{P(c|f(x))}{Q(c)}) \qquad (3)$$

MEAN SQUARED ERROR (MSE)

MSE measures the Euclidean distance between two vectors e.g. two neural network outputs $f(x), f(y)$ for the images $x, y \in X$. In contrast to the loss CE or KL, MSE is not a probability measure and therefore the vectors can be in an arbitrary Euclidean feature space (see Equation 4). The minimization of the MSE will pull the two vectors or as in the example the network outputs together. Similar to the minimization of entropy, this would lead to a degeneration of the network if this loss is used as the only loss on the network outputs.

$$MSE(f(x), f(y)) = ||f(x) - f(y)||_2^2 \qquad (4)$$

MUTUAL INFORMATION (MI)

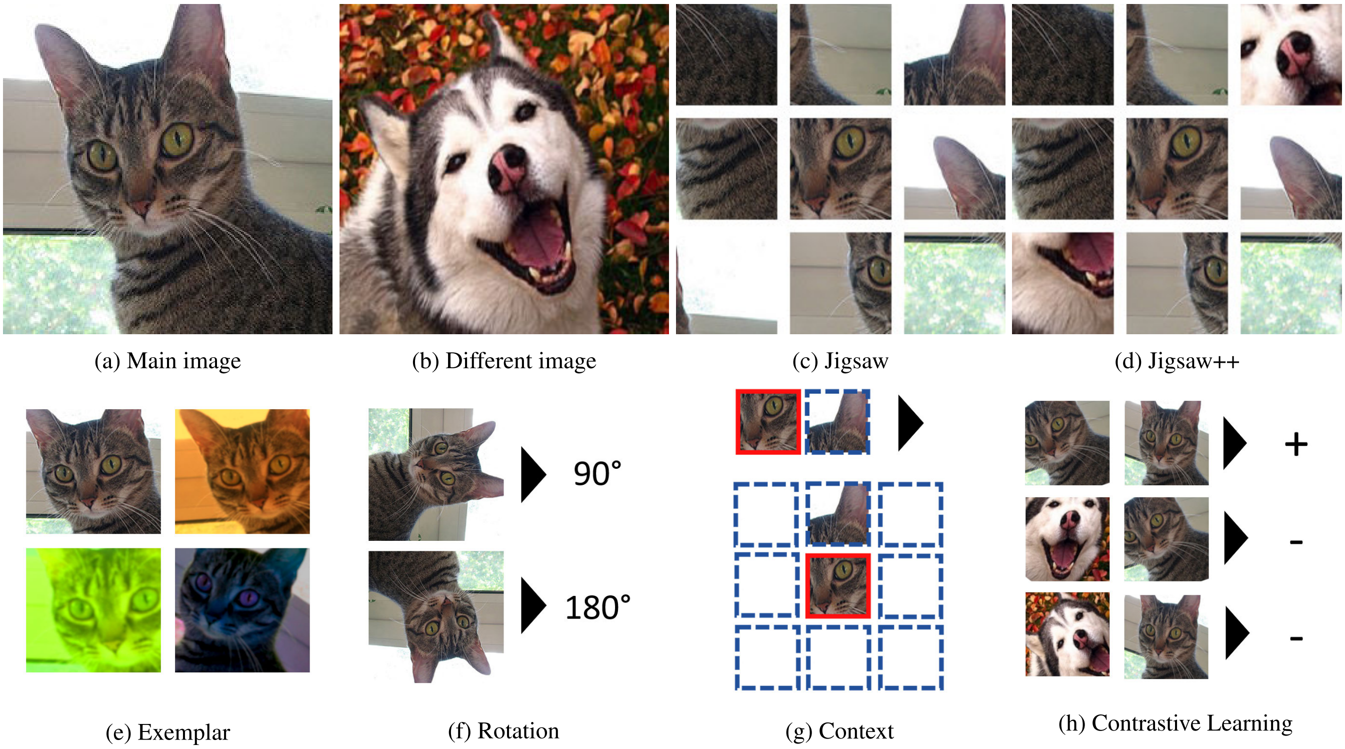
MI is defined for two probability distributions P, Q as the Kullback Leiber (KL) divergence between the joint distribution and the marginal distributions [63]. In many reduced supervised methods, the goal is to maximize the mutual information between the distributions. These distributions could be based on the input, the output, or an intermediate step of a neural network. In most cases, the conditional distribution between P and Q and therefore the joint distribution is not known. For example, we could use the outputs of a neural network $f(x), f(y)$ for two augmented views x, y of the same image as the distributions P, Q. In general, the distributions could be dependent as x, y could be identical or very similar and the distributions could be independent if x, y they are crops of distinct classes e.g. the background sky and the foreground object. Therefore, the mutual information needs to be approximated. The used approximation varies depending on the method and the definition of the distributions P, Q. For further theoretical insights and several approximations see [59], [64].

We show the definition of the mutual information between two network outputs $f(x), f(y)$ for images $x, y \in X$ as an example in Equation 5. This equation also shows an alternative representation of mutual information: the separation in entropy $H(P(\cdot|f(x)))$ and conditional entropy $H(P(\cdot|f(x)) \mid P(\cdot|f(y)))$. Ji *et al.* argue that this representation illustrates the benefits of using MI over CE in unsupervised cases [14]. A degeneration is avoided because MI balances the effects of maximizing the entropy with a uniform distribution for $P(\cdot|f(x))$ and minimizing the conditional entropy by equalizing $P(\cdot|f(x))$ and $P(\cdot|f(y))$. Both cases lead to a degeneration of the neural network on their own.

$$\begin{aligned} &I(P(\cdot|f(x), P(\cdot|f(y)) \\ &\quad = KL(P(\cdot|f(x), f(y)) \,||\, P(\cdot|f(x) * P(\cdot|f(y)))) \\ &\quad = \sum_{c=1,c'=1}^{C} P(c, c'|f(x), f(y)) \\ &\qquad log(\frac{P(c, c'|f(x), f(y))}{P(c|f(x) * P(c'|f(y)))}) \\ &\quad = H(P(\cdot|f(x)) + H(P(\cdot|f(x)) \mid P(\cdot|f(y))) \qquad (5) \end{aligned}$$

VIRTUAL ADVERSARIAL TRAINING (VAT)

VAT [65] 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. The loss is defined in Equation 6 with an image $x \in X$ and the output of a given



(a) Main image (b) Different image (c) Jigsaw (d) Jigsaw++

(e) Exemplar (f) Rotation (g) Context (h) Contrastive Learning

FIGURE 6. Illustrations of 8 selected pretext tasks – (a) Example image for the pretext task (b) Negative/different example image in the dataset or batch (c) The Jigsaw pretext task consists of solving a simple Jigsaw puzzle generated from the main image. (d) Jigsaw++ augments the Jigsaw puzzle by adding in parts of a different image. (e) In the exemplar pretext task, the distributions of a weakly augmented image (upper right corner) and several strongly augmented images should be aligned. (f) An image is rotated around a fixed set of rotations e.g. 0, 90, 180, and 270 degrees. The network should predict the rotation which has been applied. (g) 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. (h) The network receives a list of pairs and should predict the positive pairs. In this example, a positive pair consists of augmented views from the same image. Some illustrations are inspired by [40], [42], [44].

neural network $f(x)$.

$$
\begin{aligned}
VAT(f(x)) &= D(P(\cdot|f(x), P(\cdot|f(x+r_{adv})) \\
r_{adv} &= \underset{r;||r||\leq\epsilon}{\operatorname{argmax}} D(P(\cdot|f(x), P(\cdot|f(x+r)) \qquad (6)
\end{aligned}
$$

P is the probability distribution over the outputs of the neural network and D is a non-negative function that measures the distance. As illustrated in Figure 5a r is a vector and ϵ the maximum length of this vector. Two examples of used distance measures are cross-entropy [65] and Kullback-Leiber divergence [15].

CONCEPTS

MIXUP (MU)

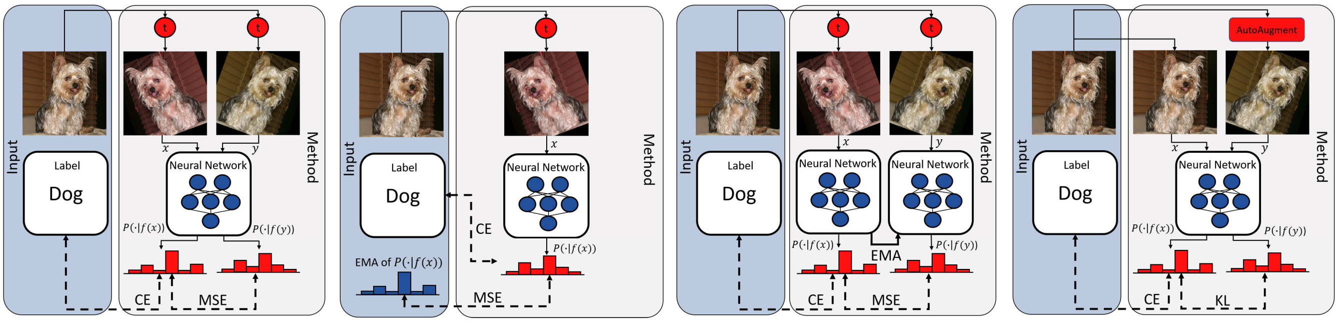
Mixup creates convex combinations of images by blending them into each other. An illustration of the concept is given in Figure 5b. The prediction of the convex combination of the corresponding labels turned out to be beneficial because the network needs to create consistent predictions for intermediate interpolations of the image. This approach has been beneficial for supervised learning in general [66] and is therefore also used in several semi-supervised learning algorithms [26], [45], [46].

OVERCLUSTERING (OC)

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 [14], [27], [67]. 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/fuzzy data or with intermediate classes that were sorted into adjacent classes randomly [27]. An illustration of this idea is presented in Figure 5c

PRETEXT TASK (PT)

A pretext task is a broad-ranged description of self-supervised training a neural network on a different task than the target task. This task can be for example predicting the rotation of an image [40], solving a jigsaw puzzle [43], using a contrastive loss [25], [55] or maximizing mutual information [14], [27]. An overview of most pretext task in this survey is given in Figure 6 and a complete overview is given in Table 1. In most cases the self-supervised, pretext task is used to learn representations which can then be fine-tuned for image classification [25], [40], [42]–[44],



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

FIGURE 7. Illustration of four selected one-stage-semi-supervised methods – The used method is given below each image. The input including label information is given in the blue box on the left side. On the right side, an illustration of the method is provided. In general, the process is organized from top to bottom. At first, the input images are preprocessed by none or two different random transformations *t*. Special augmentation techniques like Autoaugment [69] are represented by a red box. 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 $P(\cdot|f(x))$ on labeled examples. Details about the methods can be found in the corresponding entry in section III whereas abbreviations for common methods are defined in subsection II-B. EMA stands for the exponential moving average.

[55], [68]. In a semi-supervised context, some methods use this pretext task to define an additional loss during training [45].

PSEUDO-LABELS (PL)

A simple approach for estimating labels of unknown data is using Pseudo-Labels [47]. Lee proposed to classify unseen data with a neural network and use the predictions as labels. This process is illustrated in Figure 5d. What sounds at first like a self-fulfilling assumption works reasonably well in real-world image classification tasks. It is important to notice that the network needs additional information to prevent total random predictions. This additional information could be some known labels or a weight initialization of other supervised data or unsupervised on a pretext task. Several modern methods are based on the same core idea of creating labels by predicting them on their own [46], [48].

III. METHODS

This section shorty summarizes all methods in the survey in roughly chronological order and separated by their training strategy. Each summary states the used common ideas, explains their usage, and highlights special cases. The abbreviations for the common ideas are defined in subsection II-B. We include a large number of recent methods but we do not claim this list to be complete.

A. ONE-STAGE-SEMI-SUPERVISED

PSEUDO-LABELS

Pseudo-Labels [47] describes a common idea in deep learning and a learning method on its own. For the description of the common idea see above in subsection II-B. 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 common idea. Therefore, the labeled and unlabeled data are used in parallel to minimize the CE loss. *Common ideas: CE, CE*, PL*

π-MODEL AND TEMPORAL ENSEMBLING

Laine & Aila present two similar learning methods with the names π-model and Temporal Ensembling [49]. 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 [49]. Illustrations of these methods are given in Figure 7. *Common ideas: CE, MSE*

MEAN TEACHER

With Mean Teacher Tarvainen & Valpola present a student-teacher-approach for semi-supervised learning [48]. They develop their approach based on the π-model and Temporal Ensembling [49]. 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 to use a teacher based on the 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 7. *Common ideas: CE, MSE*

VIRTUAL ADVERSARIAL TRAINING (VAT)

VAT [65] is not just the name for a common 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 [65]. They showed that the adversarial transformation leads to a lower error on image classification than random transformations. Furthermore, they showed that adding EntMin [61] to the loss increased accuracy even more. *Common ideas: CE, (EM), VAT*

INTERPOLATION CONSISTENCY TRAINING (ICT)

ICT [70] uses linear interpolations of unlabeled data points to regularize the consistency between images. Verma *et al.* use a combination of the supervised loss CE and the unsupervised loss MSE. The unsupervised loss is measured between the prediction of the interpolation of two images and the interpolation of their Pseudo-Labels. The interpolation is generated with the mixup [66] algorithm from two unlabeled data points. For these unlabeled data points, the Pseudo-Labels are predicted by a Mean Teacher [48] network. *Common ideas: CE, MSE, MU, PL*

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 [71]. They analyzed the learning process based on ideas and concepts of SWA [72], π-model [49] and Mean Teacher [48]. 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 [71]. The architecture and loss is either copied from π-model [49] or Mean Teacher [48]. *Common ideas: CE, MSE*

MixMatch

MixMatch [46] 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. This sharpening also enforces implicitly a minimization of the entropy on the unlabeled data. Furthermore, they extend the algorithm mixup [66] to semi-supervised learning by incorporating the generated labels. *Common ideas: CE, (EM), MSE, MU, PL*

ENSEMPLE AutoEndocing TRANSFORMATION (EnAET)

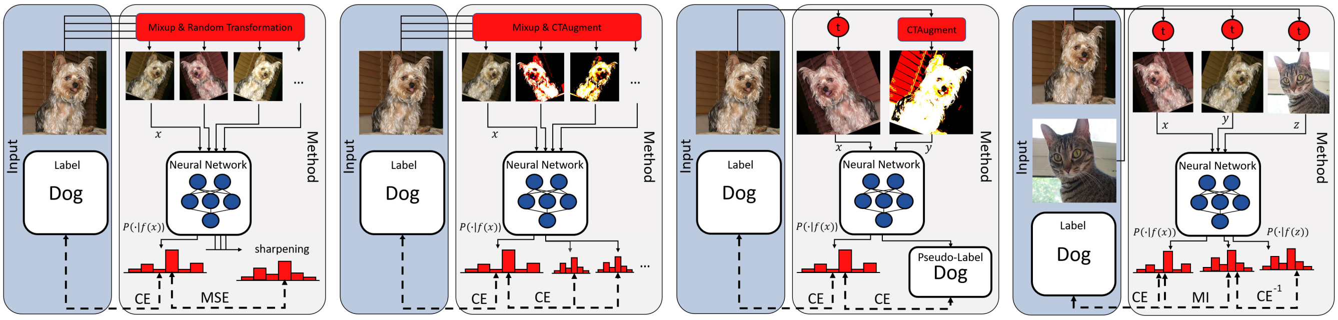
EnAET [73] combines the self-supervised pretext task AutoEncoding Transformations [74] with MixMatch [46]. Wang *et al.* apply spatial transformations, such as translations and rotations, and non-spatial transformations, such as color distortions, on input images in the pretext task. The transformations are then estimated with the original and augmented image given. This is a difference to other pretext tasks where the estimation is often based on the augmented image only [40]. 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. *Common ideas: CE, (EM), KL, MSE, MU, PL, PT*

UNSUPERVISED DATA AUGMENTATION (UDA)

Xie *et al.* present with UDA a semi-supervised learning algorithm that concentrates on the usage of state-of-the-art augmentation [16]. They use a supervised and an unsupervised loss. The supervised loss is CE whereas 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 [69] in combination with Cutout [75]. 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 [61] and a kind of Pseudo-Labeling [47]. 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 7. *Common ideas: CE, EM, KL, (PL)*

SELF-PACED MULTI-VIEW CO-TRAINING (SpamCo)

Ma *et al.* propose a general framework for co-training across multiple views [76]. In the context of image classification, different neural networks can be used as different views. The main idea of the co-training between different views is similar to using Pseudo-Labels. The main differences in SpamCo are that the Pseudo-Labels are not used for all samples and they influence each other across views. Each unlabeled image has a weight value for each view. Based on an age parameter, more unlabeled images are considered in each iteration. At first only confident Pseudo-Labels are used and over time also less confident ones are allowed. The proposed hard or soft co-regularizers also influence the weighting of the unlabeled images. The regularizers encourage to select unlabeled images for training across views. Without this regularization the training would degenerate to an independent training of the different views/models. CE is used as loss on the labels and Pseudo-Labels with additional L_2 regularization. Ma *et al.* show further applications including text classification and object detection. *Common ideas: CE, CE*, MSE, PL*



(a) MixMatch (b) ReMixMatch (c) FixMatch (d) FOC

FIGURE 8. Illustration of four selected methods – The used method is given below each image. The input including label information is given in the blue box on the left side. On the right side, an illustration of the method is provided. For FOC the second stage is represented. In general, the process is organized from top to bottom. At first, the input images are preprocessed by none or two different random transformations *t*. Special augmentation techniques like CTAugment [45] are represented by a red box. The following neural network uses these preprocessed images (e.g. *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 $P(\cdot|f(x))$ on labeled examples. Details about the methods can be found in the corresponding entry in section III whereas abbreviations for common methods are defined in subsection II-B.

ReMixMatch

ReMixMatch [45] is an extension of MixMatch with distribution alignment and augmentation anchoring. Berthelot *et al.* motivate the distribution alignment with an analysis of mutual information. They use entropy minimization via "sharpening" but 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 [40] was added. *Common ideas: CE, CE* (EM), (MI), MU, PL, PT*

FixMatch

FixMatch [26] is building on the ideas of ReMixMatch but is dropping several ideas to make the framework more simple while achieving a better performance. FixMatch is using the cross-entropy loss on the supervised and the unsupervised data. For each image in the unlabeled data, one weakly- and one strongly-augmented version is created. The Pseudo-Label of the weakly-augmented version is used if a confidence threshold is surpassed by the network. If a Pseudo-Label is calculated the network output of the strongly-augmented version is compared with this hard label via cross-entropy which implicitly encourages low-entropy predictions on the unlabeled data [26]. Sohn *et al.* do not use ideas like Mixup, VAT, or distribution alignment but they state that they can be used and provide ablations for some of these extensions. *Common ideas: CE, CE*, (EM), PL*

B. MULTI-STAGE-SEMI-SUPERVISED

EXEMPLAR

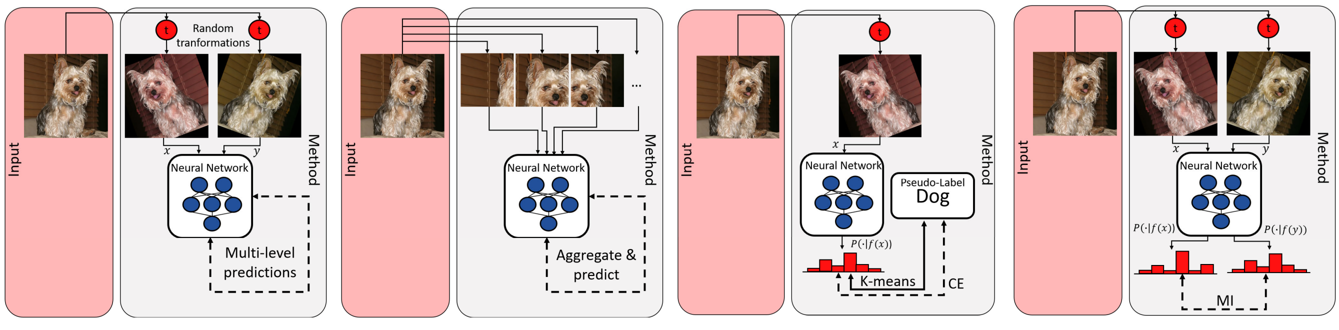
Dosovitskiy *et al.* proposed a self-supervised pretext task with additional fine-tuning [68]. They randomly sample patches from different images and augment these patches heavily. Augmentations can be for example rotations, translations, color changes, or contrast adjustments. The classification task is to map all augmented versions of a patch to the correct original patch using cross-entropy loss. *Common ideas: CE, CE*, PT*

CONTEXT

Doersch *et al.* propose to use context prediction as a pretext task for visual representation learning [42]. 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 using cross-entropy loss. An illustration of the pretext task is given in Figure 6. 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 the random initialization. Aside from fine-tuning, Doersch *et al.* show how their method could be used for Visual Data Mining. *Common ideas: CE, CE*, PT*

JIGSAW

Noroozi and Favaro propose to solve Jigsaw puzzles as a pretext task [43]. The idea is that a network has to understand the concept of a presented object to solve the puzzle using the classification loss cross-entropy. 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 [44]. They call the extension Jigsaw++.



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

FIGURE 9. 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 and only the first stage/pretext task is represented. In general, the process is organized from top to bottom. At first, the input images are either preprocessed by one or two random transformations t 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 distributions $(P(\cdot|f(x)), P(\cdot|f(y)))$ to calculate a loss. Details about the methods can be found in the corresponding entry in section III whereas abbreviations for common methods are defined in subsection II-B.

Examples for a Jigsaw or Jigsaw++ puzzle are given in Figure 6. *Common ideas: CE, CE*, PT*

DeepCluster

DeepCluster [67] is a self-supervised method that generates 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 9. *Common ideas: CE, OC, PL, PT*

ROTATION

Gidaris *et al.* use a pretext task based on image rotation prediction [40]. They propose to randomly rotate the input image by 0, 90, 180, or 270 degrees and let the network predict the chosen rotation degree. They train the network with cross-entropy on this classification task. 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. *Common ideas: CE, CE*, PT*

CONTRASTIVE PREDICTIVE CODING (CPC)

CPC [55], [56] 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 with a contrastive loss which identifies the correct prediction out of randomly sampled negative ones. They call their loss InfoNCE which is cross-entropy for the prediction of positive examples [55]. 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 [55]. An illustration of this method is given in Figure 9. The representations of the pretext task are then fine-tuned. *Common ideas: CE, (CE*), CL, (MI), PT*

CONSTRASTIVE MULTIVIEW CODING (CMC)

CMC [54] generalizes CPC [55] 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 [55]. 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. *Common ideas: CE, (CE*), CL, (MI), PT*

DEEP InfoMax (DIM)

DIM [77] 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. Also, 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. *Common ideas: CE, MI, PT*

AUGMENTED MULTISCALE DEEP InfoMax (AMDIM)

AMDIM [78] maximizes the MI between inputs and outputs of a network. It is an extension of the method DIM [77]. 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 9. *Common ideas: CE, MI, PT*

DEEP METRIC TRANSFER (DMT)

DMT [79] 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 [80] or unsupervised instance discrimination [81] to calculate a metric. In the second stage, they propagate labels to unlabeled data with spectral clustering and then fine-tune the network 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 [48] or VAT [65], they can improve the accuracy with very few labels by about 30%. *Common ideas: CE, CE*, PL, PT*

INVARIANT INFORMATION CLUSTERING (IIC)

IIC [14] 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 9. The first unsupervised stage can be seen as a self-supervised pretext task. In contrast to other pretext tasks, this task already predicts representations which can be seen as classifications. *Common ideas: CE, MI, OC, PT*

SELF-SUPERVISED SEMI-SUPERVISED LEARNING (S^4L)

S^4L [15] is, as the name suggests, a combination of self-supervised and semi-supervised methods. Zhai *et al.* split the loss into a supervised and an unsupervised part. The supervised loss is CE whereas the unsupervised loss is based on the self-supervised techniques using rotation and exemplar prediction [40], [68]. The authors show that their method performs better than other self-supervised and semi-supervised techniques [40], [47], [61], [65], [68]. 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. *Common ideas: CE, CE*, EM, PL, PT, VAT*

SIMPLE FRAMEWORK FOR CONTRASTIVE LEARNING OF VISUAL REPRESENTATION (SimCLR)

SimCLR [25] maximizes the agreement between two different augmentations of the same image. The method is similar to CPC [55] and IIC [14]. 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 whether 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. *Common ideas: CE, (CE*), CL, PT*

FUZZY OVERCLUSTERING (FOC)

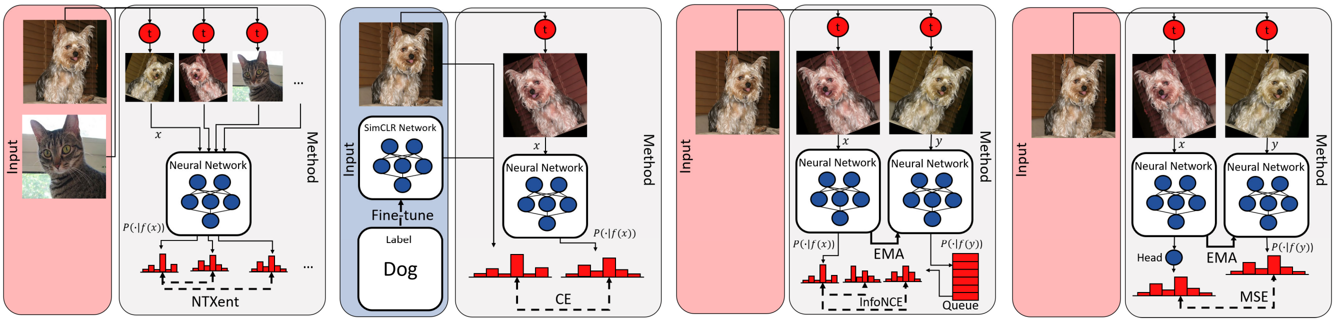
Fuzzy Overclustering [27] is an extension of IIC [14]. FOC focuses on using overclustering to subdivide fuzzy labels in real-world datasets. Therefore, it unifies the used data and losses proposed by IIC between the different stages and extends it with new ideas such as the novel loss Inverse Cross-Entropy (CE^{-1}). This loss is inspired by Cross-Entropy but can be used on the overclustering results of the network where no ground truth labels are known. FOC is not achieving state-of-the-art results on a common image classification dataset. However, on a real-world plankton dataset with fuzzy labels, it surpasses FixMatch and shows that 5-10% more consistent predictions can be achieved. Like IIC, FOC can be viewed as a multi-stage-semi-supervised and an one-stage-unsupervised method. In general, FOC is trained in one unsupervised and one semi-supervised stage and can be seen as a multi-stage-semi-supervised method. Like IIC, it produces classifications already in the unsupervised stage and can therefore also be seen as an one-stage-unsupervised method. *Common ideas: CE, (CE*) MI, OC, PT*

MOMENTUM CONTRAST (MoCo)

He *et al.* propose to use a momentum encoder for contrastive learning [82]. In other methods [25], [55]–[57], the negative examples for the contrastive loss are sampled from the same mini-batch as the positive pair. A large batch size is needed to ensure a great variety of negative examples. He *et al.* sample their negative examples from a queue encoded by another network whose weights are updated with an exponential moving average of the main network. They solve the pretext task proposed by [81] with negative examples samples from their queue and fine-tune in a second stage on labeled data. Chen *et al.* provide further ablations and baseline for the MoCo Framework e.g. by using a MLP head for fine-tuning [83]. *Common ideas: CE, CL, PT*

BOOTSTRAP YOU OWN LATENT (BYOL)

Grill *et al.* use an online and a target network. In the proposed pretext task, the online network predicts the image representation of the target network for an image [28]. The difference between the predictions is measured with MSE. Normally, this approach would lead to a degeneration of the network as a constant prediction over all images would also



(a) SimCLR (b) SimCLRv2 (c) MoCo (d) BYOL

FIGURE 10. Illustration of four selected multi-stage-semi-supervised methods – The used method is given below each image. The input is given in the red (not using labels) or blue (using labels) box on the left side. On the right side, an illustration of the method is provided. The fine-tuning part is excluded and only the first stage/pretext task is represented. For SimCLRv2 the second stage or distillation step is illustrated. In general, the process is organized from top to bottom. At first, the input images are either preprocessed by one or two random transformations *t* or are split up. The following neural network uses these preprocessed images (x, y) as input. Details about the methods can be found in the corresponding entry in section III whereas abbreviations for common methods are defined in subsection II-B. EMA stands for the exponential moving average.

achieve the goal. In contrastive learning, this degeneration is avoided by selecting a positive pair of examples from multiple negative ones [25], [55]–[57], [82], [83]. By using a slow-moving average of the weights between the online and target network, Grill *et al.* show empirically that the degeneration to a constant prediction can be avoided. This approach has the positive effect that BYOL performance is depending less on hyperparameters like augmentation and batch size [28]. In a follow-up work, Richemond *et al.* show that BYOL even works when no batch normalization which might have introduced kind of a contrastive learning effect in the batches is used [84]. *Common ideas: MSE, PT*

SIMPLE FRAMEWORK FOR CONTRASTIVE LEARNING OF VISUAL REPRESENTATION (SimCLRv2)

Chen *et al.* extend the framework SimCLR by using larger and deeper networks and by incorporating the memory mechanism from MoCo [57]. Moreover, they propose to use this framework in three steps. The first is training a contrastive learning pretext task with a deep neural network and the SimCLRv2 method. The second step is fine-tuning this large network with a small amount of labeled data. The third step is self-training or distillation. The large pretrained network is used to predict Pseudo-Labels on the complete (unlabeled) data. These (soft) Pseudo-Labels are then used to train a smaller neural network with CE. The distillation step could be also performed on the same network as in the pretext task. Chen *et al.* show that even this self-distillation leads to performance improvements [57]. *Common ideas: CE, (CE*), CL, PL, PT*

C. ONE-STAGE-UNSUPERVISED

DEEP ADAPTIVE IMAGE CLUSTERING (DAC)

DAC [50] 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. One can interpret these similarities and dissimilarities as Pseudo-Labels for the similarity classification task. 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. *Common ideas: PL*

INFORMATION MAXIMIZING SELF-AUGMENTED TRAINING (IMSAT)

IMSAT [85] 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 [65]. The maximization of MI directly on the image input leads to a problem. For datasets like CIFAR-10, CIFAR-100 [86] and STL-10 [52] 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 [1] as input. *Common ideas: MI, VAT*

INVARIANT INFORMATION CLUSTERING (IIC)

IIC [14] is described above 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. *Common ideas: MI, OC*

FUZZY OVERCLUSTERING (FOC)

FOC [27] is described above as a multi-stage-semi-supervised method. Like IIC, FOC can also be seen as an one-

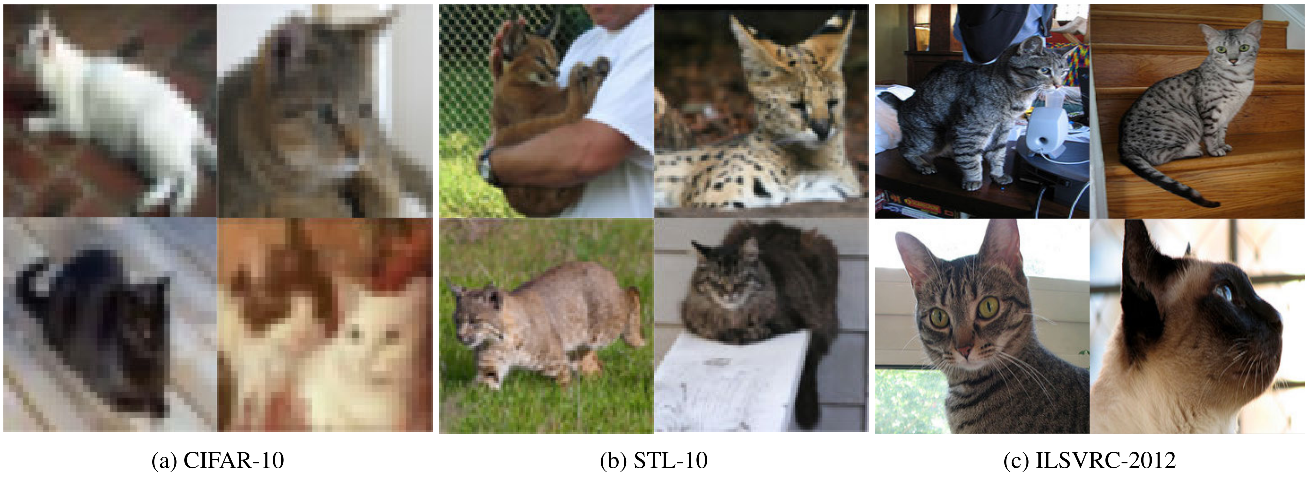


FIGURE 11. Examples of four random cats in the different datasets to illustrate the difference in quality.

stage-unsupervised method because the first stage yields cluster predictions. *Common ideas: MI, OC*

SEMANTIC CLUSTERING BY ADOPTING NEAREST NEIGHBORS (SCAN)

Gansbeke *et al.* calculate clustering assignments building on self-supervised pretext task by mining the nearest neighbors and using self-labeling. They propose to use SimCLR [25] as a pretext task but show that other pretext tasks [40], [81] could also be used for this step. For each sample, the k nearest neighbors are selected in the gained feature space. The novel semantic clustering loss encourages these samples to be in the same cluster. Gansbeke *et al.* noticed that the wrong nearest neighbors have a lower confidence and propose to create Pseudo-Labels on only confident examples for further fine-tuning. They also show that Overclustering can be successfully used if the number of clusters is not known before. *Common ideas: OC, PL, PT*

IV. ANALYSIS

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. 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 11.

CIFAR-10 AND CIFAR-100

are large datasets of tiny color images with size 32 × 32 [86]. 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 examples and 10,000 validation examples (image + label). 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 [52]. The dataset is inspired by CIFAR-10 [86] 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 96 × 96 color images and were acquired in combination with their labels from ImageNet [1].

ILSVRC-2012

is a subset of ImageNet [1]. The training set consists of 1.2 million images whereas the validation and the test set include 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 IV-B. 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.

B. 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 (σ) from the network cluster predictions to the ground-truth classes. For N images $x_1, \ldots, x_N \in X_l$ with labels z_{x_i} and predictions $f(x_i) \in \mathbb{R}^C$ the accuracy is defined in Equation 7 whereas the cluster accuracy is defined in Equation 8.

$$ACC(x_1, \ldots, x_N) = \frac{\sum_{i=1}^{N} \mathbb{1}_{z_{x_i} = \mathrm{argmax}_{1 \le j \le C} f(x_i)_j}}{N} \tag{7}$$

$$ACC(x_1, \ldots, x_N) = \max_{\sigma} \frac{\sum_{i=1}^{N} \mathbb{1}_{z_{x_i} = \sigma(\mathrm{argmax}_{1 \le j \le C} f(x_i)_j)}}{N} \tag{8}$$

C. 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 CONCERNING USED COMMON IDEAS

In Table 1 we present all methods and their used common ideas. Following our definition of common ideas in subsection II-B, 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 III 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 major separation between the training strategies can be based on CE and pretext tasks. All one-stage-semi-supervised methods use a cross-entropy loss during training whereas only two use additional losses based on pretext tasks. All multi-stage-semi-supervised methods use a pretext task and use CE for fine-tuning. All one-stage-semi-supervised methods use no CE and often use a pretext task. 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 of the two semi-supervised strategies. These common ideas are EM, KL, MSE, and MU for one-stage-semi-supervised methods and CL, MI, and OC for multi-stage-semi-supervised methods. We hypothesize that this shared and different usage of ideas exists due to the different usage of unlabeled data. For example, one-stage-semi-supervised methods use the unlabeled and labeled data in the same stage and therefore might need to regularize the training with MSE.

If we compare multi-stage-semi-supervised and one-stage-unsupervised training we notice that MI, OC, and PT are often used in both. All three of them are not often used 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. CE*, PL, 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 PL to filter the unlabeled data for useful images. CE* seems to be more often used by multi-stage-semi-supervised methods. The parentheses in Table 1 indicate that they often also motivate another idea like CE^{-1} [27] or the CL loss [25], [55]. 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 but is also supported by their usage of common ideas.

COMPARISON CONCERNING 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. Whereas 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 [89]. They state that 'neither […] the ranking of architectures [is] consistent across different methods, nor is the ranking of methods consistent across architectures' [89]. Most methods try to achieve comparability with previous ones by a similar setup but over time small differences still aggregate and lead to a variety of used architectures. Some methods use only early convolutional networks such as AlexNet [1] but others use more modern architectures like Wide ResNet-Architecture [90] or Shake-Shake-Regularization [91].

Oliver *et al.* proposed guidelines to ensure more comparable evaluations in semi-supervised learning [92]. They showed that not following these guidelines may lead to changes in the performance [92]. Whereas 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 specialties 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 labels as normally defined in subsection IV-A.

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

TABLE 1. Overview of the methods and their used common ideas — On the left-hand side, the reviewed methods from section III are sorted by the training strategy. The top row lists the common ideas. Details about the ideas and their abbreviations are given in subsection II-B. The last column and some rows sum up the usage of ideas per method or training strategy. *Legend*: (X) The idea is only used indirectly. The individual explanations are given in section III.

	CE	CE*	EM	CL	KL	MSE	MU	MI	OC	PT	PL	VAT	*Overall Sum*
One-Stage-Semi-Supervised													
Pseudo-Labels [47]	X	X									X		3
π model [49]	X					X							2
Temporal Ensembling [49]	X					X							2
Mean Teacher [48]	X					X							2
VAT [65]	X											X	2
VAT + EntMin [65]	X		X									X	3
ICT [70]	X					X	X				X		4
fast-SWA [71]	X					X							2
MixMatch [46]	X		(X)			X	X				X		5
EnAET [73]	X		(X)		X	X	X			AET	X		7
UDA [16]	X		X		X						(X)		4
SPamCO [76]	X	X				X					X		4
ReMixMatch [45]	X	X	(X)				X	(X)		Rotation	X		7
FixMatch [26]	X	X	(X)								X		4
Sum	14	4	6	0	2	8	4	1	0	2	8	2	47
Multi-Stage-Semi-Supervised													
Exemplar [68]	X	X								Augmentation			3
Context [42]	X	X								Context			3
Jigsaw [43]	X	X								Jigsaw			3
DeepCluster [67]	X	X							X	Clustering	X		5
Rotation [40]	X	X								Rotation			3
CPC [55], [56]	X	(X)		X				(X)		CL			5
CMC [54]	X	(X)		X				(X)		CL			5
DIM [77]	X							X		MI			3
AMDIM [78]	X							X		MI			3
DMT [79]	X	X				X				Metric	X		5
IIC [14]	X							X	X	MI			4
S^{4}L [15]	X	X	X							Rotation	X	X	6
SimCLR [25]	X	(X)								CL			3
MoCo [82]	X			X						Metric			3
BYOL [28]	X					X				Bootstrap			3
FOC [27]	X	(X)						X	X	MI			5
SimCLRv2 [57]	X	(X)		X						CL	X		5
Sum	17	11	1	5	0	1	0	6	3	17	4	1	66
One-Stage-Unsupervised													
DAC [50]											X		1
IMSAT [85]								X				X	2
IIC [14]								X	X	MI			3
FOC [27]								X	X	MI			3
SCAN [41]									X	CL	X		3
Sum	0	0	0	0	0	0	0	3	3	3	2	1	12
Overall Sum	31	54	7	5	2	9	4	10	6	22	14	4	125

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 FixMatch reach an accuracy of more than 95% and are roughly three percent worse than the fully supervised baseline. For the CIFAR-100 dataset, fewer results are reported. FixMatch is with about 77% on this dataset the best method in comparison to the fully supervised baseline of about 80%. Newer methods also provide results for 1000 or even 250 labels instead of 4000 labels. Especially EnAET, ReMixMatch, and FixMatch 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 more than 95%. FixMatch reports an accuracy of nearly 95% with only 1000 labels. This is more than most methods achieve with 5000 labels.

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 is given in the second column. The last four columns report the Top-1 accuracy score in % for the respective dataset (See subsection IV-B 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. Be aware that different architectures and frameworks are used which might impact the results. Please see subsection IV-C for a detailed explanation. *Legend*: † 100% of the labels are used instead of the default value defined in subsection IV-A. ‡ Multilayer perceptron is 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. ^{7} The network uses selective kernels [87].

	Architecture	Publication	CIFAR-10	CIFAR-100	STL-10	ILSVRC-2012	ILSVRC-2012 (Top-5)
Supervised (100% labels)	Best reported	-	98.01 [73]	79.82 [78]	68.7 [77]	85.7 [88]	97.6 [88]
One-Stage-Semi-Supervised							
Pseudo-Label [47]	ResNet50v2 [2]	2013					82.41 [15]
π model [49]	CONV-13	2017	87.64				
Temporal Ensembling [49]	CONV-13	2017	87.84				
Mean Teacher [48]	CONV-13	2017	87.69				
Mean Teacher [48]	Wide ResNet-28	2017	89.64				90.9 [57]
VAT [65]	CONV-13	2018	88.64				
VAT [65]	ResNet50v2	2018					82.78 [15]
VAT + EntMin [65]	CONV-13	2018	89.45				
VAT + EntMin [65]	ResNet50v2	2018	86.41 [15]				83.3 [15]
ICT [70]	Wide ResNet-28	2019	92.34				
ICT [70]	CONV-13	2019	92.71				
fast-SWA [71]	CONV-13	2019	90.95	66.38			
fast-SWA [71]	ResNet-26^{1}	2019	93.72				
MixMatch [46]	Wide ResNet-28	2019	95.05	74.12	94.41		
EnAET [73]	Wide ResNet-28	2019	94.65	73.07	95.48		
UDA [16]	Wide ResNet-28	2019	94.7			68.66	88.52
SPamCo [76]	Wide ResNet-28	2020	92.95				
ReMixMatch [45]	Wide ResNet-28	2020	94.86	76.97 [26]			
FixMatch [26]	Wide ResNet-28	2020	95.74	77.40			
FixMatch [26]	ResNet-50	2020				71.46	89.13
Multi-Stage-Semi-Supervised							
Exemplar [68]	ResNet50	2015				46.0† [89]	81.01 [15]
Context [42]	ResNet50	2015				51.4† [89]	
Jigsaw [43]	AlexNet	2016				44.6† [89]	
DeepCluster [67]	AlexNet	2018			73.4 [14]	41†	
Rotation [40]	AlexNet	2018				55.4† [89]	
Rotation [40]	ResNet50v2	2018					78.53 [15]
CPC [56]	ResNet-170	2020	77.45† [77]		77.81† [77]	61.0	84.88
CMC [54]	AlexNet	2019			86.88‡		
CMC [54]	ResNet-50^{6}	2019				70.6	89.7*
DIM [77]	AlexNet	2019			72.57‡		
DIM [77]	GAN Discriminator	2019	75.21†‡	49.74†‡			
AMDIM [78]	ResNet18	2019	91.3† / 93.6†‡	70.2† / 73.8†‡	93.6 / 93.8‡	60.2† / 60.9†‡	
DMT [79]	Wide ResNet-28	2019	88.70				
IIC [14]	ResNet34	2019			85.76 [27] / 88.8‡		
S^{4}L [15]	ResNet50v2^{2}	2019				73.21	91.23*
SimCLR [25]	ResNet50v2^{2}	2020				74.4 [57] / 76.5†	92.6 / 93.2†
MOCO [82]	ResNet50^{2}	2020				68.6	
MOCO [82]	ResNet50	2020				60.6† / 71.1†‡ [83]	
BYOL [28]	ResNet200^{2}	2020				77.7	93.7
FOC [27]	ResNet34	2020			86.49		
SimCLRv2 [57]	ResNet-152^{2,7}	2020				80.9‡	95.5‡
One-Stage-Unsupervised							
DAC [50]	All-ConvNet	2017	52.18	23.75	46.99	52.72^{3}	
IMSAT [85]	Autoencoder^{5}	2017	45.6	27.5	94.1		
IIC [14]	ResNet34	2019	61.7	25.7^{4}	59.6		
FOC [27]	ResNet34	2020			60.45		
SCAN [41]	ResNet18	2020	88.3	50.7^{4}	80.9		

The ILSVRC-2012 dataset is the most difficult dataset based on the reported Top-1 accuracies. Most methods only achieve a Top-1 accuracy which is roughly 20% worse than the reported supervised baseline with around 86%. Only the methods SimCLR, BYOL, and SimCLRv2 achieve an accuracy that is less than 10% worse than the baseline. SimCLRv2 achieves the best accuracy with a Top-1 accuracy of 80.9% and a Top-5 accuracy of around 96%. For fewer labels also SimCLR, BYOL and SimCLRv2 achieve the best results.

The unsupervised methods are separated from the supervised baseline by a clear margin of up to 10%. SCAN achieves the best results in comparison to the other methods as it builds on the strong pretext task of SimCLR. This also illustrates the reason for including the unsupervised method in a comparison with semi-supervised methods. Unsupervised methods do not use labeled examples and therefore are expected to be worse. However, the data show that the gap of 10% is not large and that unsupervised methods can benefit from ideas of self-supervised learning. Some paper report results

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 is either given as an absolute number or in percent. A blank entry represents the fact that no result was reported.

	CIFAR-10			STL-10		ILSVRC-2012		ILSVRC-2012 (Top-5)	
	4000	1000	250	5000	1000	10%	1%	10%	1%
One-Stage-Semi-Supervised									
Mean Teacher [48]	89.64	82.68	52.68						
ICT [70]	92.71	84.52	61.4 [46]						
MixMatch [46]	93.76	92.25	88.92	94.41	89.82				
EnAET [73]	94.65	93.05	92.4	95.48	91.96				
UDA [16]	95.12 [26]		91.18 [26]		92.34 [26]	68.66		88.52	
ReMixMatch [45]	94.86	94.27	93.73		93.82				
FixMatch [26]	95.74		94.93		94.83	71.46		89.13	
Multi-Stage-Semi-Supervised									
DMT [79]	88.70		80.3				58.6		
SimCLR [25]						74.4 [57]	63.0 [57]	92.6	85.8
BYOL [28]						77.7	71.2	93.7	89.5
SimCLRv2 [57]						80.9	76.6	95.5	93.4

for even fewer labels as shown in Table 3 which closes the gap to unsupervised learning further. 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.

D. DISCUSSION

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

1) TREND: REAL WORLD APPLICATIONS?

Previous methods were not scalable to real-world images and applications and used workarounds e.g. extracted features [85] to process real-world images. Many methods can report a result of over 90% on CIFAR-10, a simple low-resolution dataset. Only five methods can achieve a Top-5 accuracy of over 90% on ILSVRC-2012, a high-resolution dataset. We conclude that most methods are not scalable to high-resolution and complex image classification problems. However, the best-reported methods like FixMatch and SimCLRv2 seem to have surpassed the point of only scientific usage and could be applied to real-world classification tasks.

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 [93], [94] or noisy labels [27], [95] are not treated by the presented methods. Datasets with also few unlabeled data points are not considered. We see that good performance on well-structured datasets does not always transfer completely to real-world datasets [27]. We assume that these issues arise due to assumptions that do not hold on real-world datasets like a clear distinction between datapoints [27] and non-robust hyperparameters like augmentations and batch size [28]. Future research has to address these issues so that reduced supervised learning methods can be applied to any real-world datasets.

2) TREND: HOW MUCH SUPERVISION IS NEEDED?

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 20% 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.

In general, we considered a reduction from 100% to 10% of all labels. However, we see that methods like FixMatch and SimCLRv2 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. FixMatch even achieves a median accuracy of around 65% for one label per class for the CIFAR-10 dataset [26].

The trend that results improve overtime is expected. But the results indicate that we are near the point where semi-supervised learning needs very few to almost no labels per class (e.g. 10 labels for CIFAR10). In practice, the labeling cost for unsupervised and semi-supervised will almost be the same for common classification datasets. Unsupervised methods would need to bridge the performance gap on these classification datasets to be useful anymore. It is questionable if an unsupervised method can achieve this because it would need to guess what a human wants to have classified even when competing features are available.

We already see that on datasets like ImageNet additional data such as JFT-300M is used to further improve the supervised training [96]–[98]. These large amounts of data can only be collected without any or weak labels as the collection process has to be automated. It will be interesting to investigate if the discussed methods in this survey can also scale to such datasets while using only few labels per class.

We conclude that on datasets with few and a fixed number of classes semi-supervised methods will be more important than unsupervised methods. However, if we have a lot of classes or new classes should be detected like in few- or zero-shot learning [38], [94], [99], [100] unsupervised methods will still have a lower labeling cost and be of high importance. This means future research has to investigate how the semi-supervised ideas can be transferred to unsupervised methods as in [14], [41] and to settings with many, an unknown or rising amount of classes like in [39], [96].

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 a 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.

If we look at methods like SimCLRv2, EnAET, ReMixMatch, or S^{4}L we see that it can be beneficial to combine different ideas and mindsets. These methods used a broad range of ideas and also ideas uncommon for their respective training strategy. S^4L calls their combined approach even ''Mix of all models'' [15] and SimCLRv2 states that ''Self-Supervised Methods are Strong Semi-Supervised Learners'' [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 [15].

Methods like Fixmatch illustrate that it does not need a lot of common ideas to achieve state-of-the-art performance but rather that the selection of the correct ideas and combining them in a meaningful is important. We identified that some common ideas are not often combined and that the combination of a broad range and unusual ideas can be beneficial. We believe that the combination of the different common idea is a promising future research field because many reasonable combinations are yet not explored.

V. CONCLUSION

In this paper, we provided an overview of semi-, self-, and unsupervised methods. We analyzed their difference, similarities, and combinations based on 34 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 indicate that semi-supervised methods could be applied to real-world problems. However, issues like class imbalance and noisy or fuzzy labels are not considered. More robust methods need to be researched before semi-supervised learning can be applied to real-world issues.

The performance gap between supervised and semi- or self-supervised methods is closing and the number of labels to get comparable results to fully supervised learning is decreasing. In the future, the unsupervised methods will have almost no labeling cost benefit in comparison to the semi-supervised methods due to these developments. We conclude that in combination with the fact that semi-supervised methods have the benefit of using labels as guidance unsupervised methods will lose importance. However, for a large number of classes or an increasing number of classes the ideas of unsupervised are still of high importance and ideas from semi-supervised and self-supervised learning need to be transferred to this setting.

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. 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.

ACKNOWLEDGMENT

This work was supported by Land Schleswig-Holstein through the Open Access Publikationsfonds Funding Program.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 60, no. 6. New York, NY, USA: Association for Computing Machinery, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] J. Brünger, S. Dippel, R. Koch, and C. Veit, '''Tailception': Using neural networks for assessing tail lesions on pictures of pig carcasses,'' *Animal*, vol. 13, no. 5, pp. 1030–1036, 2019.

[4] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[5] S. Clausen, C. Zelenka, T. Schwede, and R. Koch, ''Parcel tracking by detection in large camera networks,'' in *Pattern Recognition*, T. Brox, A. Bruhn, and M. Fritz, Eds. Cham, Switzerland: Springer, 2019, pp. 89–104.

[6] J. Long, E. Shelhamer, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[9] 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,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 181–196.

[10] 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 *Proc. DAGM German Conf. Pattern Recognit.*, in Lecture Notes in Computer Science, vol. 11824, 2019, pp. 374–386.

[11] G. E. Hinton and T. J. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation*. Cambridge, MA, USA: MIT Press, 1999.

[12] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 1, 2016, pp. 740–749.

[13] M. Kaya and H. S. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, Aug. 2019.

[14] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[15] L. Beyer, X. Zhai, A. Oliver, and A. Kolesnikov, "S4L: Self-supervised semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1476–1485.

[16] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–20.

[17] O. Chapelle, B. Schölkopf, and A. Zien, "Semi-supervised learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, p. 542, Mar. 2009.

[18] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[19] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 4, 2020, doi: 10.1109/TPAMI.2020.2992393.

[20] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.

[21] G. Ciocca, C. Cusano, S. Santini, and R. Schettini, "On the use of supervised features for unsupervised image categorization: An evaluation," *Comput. Vis. Image Understand.*, vol. 122, pp. 155–171, May 2014.

[22] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[23] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep., 2008, vol. 2. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60444

[24] 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*, vol. 6, pp. 39501–39514, 2018.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[26] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–21.

[27] L. Schmarje, J. Brünger, M. Santarossa, S.-M. Schröder, R. Kiko, and R. Koch, "Beyond cats and dogs: Semi-supervised classification of fuzzy labels with overclustering," 2020, *arXiv:2012.01768*. [Online]. Available: http://arxiv.org/abs/2012.01768

[28] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–35.

[29] 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, *arXiv:1903.11260*. [Online]. Available: http://arxiv.org/abs/1903.11260

[30] V. Cheplygina, M. de Bruijne, and J. P. W. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Med. Image Anal.*, vol. 54, pp. 280–296, May 2019.

[31] A. Mey and M. Loog, "Improvability through semi-supervised learning: A survey of theoretical results," 2019, pp. 1–28, *arXiv:1908.09574*. [Online]. Available: http://arxiv.org/abs/1908.09574

[32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 3, Mar. 2017, pp. 1856–1868.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Jun. 2014, pp. 2672–2680.

[34] L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, and C. Zhang, "Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3015–3022.

[35] N. Ukita and Y. Uematsu, "Semi- and weakly-supervised human pose estimation," *Comput. Vis. Image Understand.*, vol. 170, pp. 67–78, May 2018.

[36] D. Mahapatra, "Combining multiple expert annotations using semi-supervised learning and graph cuts for medical image segmentation," *Comput. Vis. Image Understand.*, vol. 151, pp. 114–123, Oct. 2016.

[37] P. Xu, Z. Song, Q. Yin, Y.-Z. Song, and L. Wang, "Deep self-supervised representation learning for free-hand sketch," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1503–1513, Apr. 2021.

[38] L. Liu, T. Zhou, G. Long, J. Jiang, X. Dong, and C. Zhang, "Isometric propagation network for generalized zero-shot learning," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2021, pp. 1–13.

[39] Z. Yu, L. Chen, Z. Cheng, and J. Luo, "TransMatch: A transfer-learning scheme for semi-supervised few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12856–12864.

[40] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[41] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. VanGool, "SCAN: Learning to classify images without labels," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 268–285.

[42] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[43] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[44] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9359–9367.

[45] 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," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–13.

[46] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.

[47] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, vol. 3, 2013, p. 2.

[48] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[49] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.

[50] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5880–5888.

[51] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[52] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[53] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.

[54] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2019, pp. 776–794.

[55] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*. [Online]. Available: http://arxiv.org/abs/1807.03748

[56] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 4182–4192.

[57] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 1–18.

[58] T. Chen and L. Li, "Intriguing properties of contrastive losses," 2020, *arXiv:2011.02803*. [Online]. Available: http://arxiv.org/abs/2011.02803

[59] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5171–5180.

[60] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–16.

[61] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 529–536.

[62] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[63] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

[64] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and D. R. Hjelm, "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 531–540.

[65] S.-S. Learning, T. Miyato, S.-I. Maeda, M. Koyama, S. Ishii, and M. Koyama, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[66] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[67] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 132–149.

[68] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[69] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.

[70] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–23.

[71] B. Athiwaratkun, M. Finzi, P. Izmailov, A. G. Wilson, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "There are many consistent explanations of unlabeled data: Why you should average," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–22.

[72] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 1–12.

[73] X. Wang, D. Kihara, J. Luo, and G.-J. Qi, "EnAET: A self-trained framework for semi-supervised and supervised learning with ensemble transformations," 2019, *arXiv:1911.09265*. [Online]. Available: http://arxiv.org/abs/1911.09265

[74] L. Zhang, G.-J. Qi, L. Wang, and J. Luo, "AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2542–2550.

[75] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*. [Online]. Available: http://arxiv.org/abs/1708.04552

[76] F. Ma, D. Meng, X. Dong, and Y. Yang, "Self-paced multi-view co-training," *J. Mach. Learn. Res.*, vol. 21, no. 57, pp. 1–38, 2020.

[77] 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 *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–24.

[78] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15509–15519.

[79] B. Liu, Z. Wu, H. Hu, and S. Lin, "Deep metric transfer for label propagation with limited annotated data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–10.

[80] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.

[81] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[82] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[83] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*. [Online]. Available: http://arxiv.org/abs/2003.04297

[84] P. H. Richemond, J.-B. Grill, F. Altché, C. Tallec, F. Strub, A. Brock, S. Smith, S. De, R. Pascanu, B. Piot, and M. Valko, "BYOL works even without batch statistics," 2020, *arXiv:2010.10241*. [Online]. Available: http://arxiv.org/abs/2010.10241

[85] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1558–1567.

[86] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/

[87] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[88] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy: FixEfficientNet," 2020, *arXiv:2003.08237*. [Online]. Available: http://arxiv.org/abs/2003.08237

[89] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1920–1929.

[90] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 87.1–87.12.

[91] X. Gastaldi, "Shake-shake regularization," 2017, *arXiv:1705.07485*. [Online]. Available: http://arxiv.org/abs/1705.07485

[92] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 3235–3246.

[93] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[94] S.-M. Schröder, R. Kiko, and R. Koch, "MorphoCluster: Efficient annotation of plankton images by clustering," *Sensors*, vol. 20, no. 11, p. 3060, May 2020.

[95] Q. Li, X. Peng, L. Cao, W. Du, H. Xing, Y. Qiao, and Q. Peng, "Product image recognition with guidance learning and noisy supervision," *Comput. Vis. Image Understand.*, vol. 196, Jul. 2020, Art. no. 102963.

[96] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le, "Meta pseudo labels," 2020, *arXiv:2003.10580*. [Online]. Available: http://arxiv.org/abs/2003.10580

[97] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (BiT): General visual representation learning," in *Proc. Eur. Conf. Comput. Vis.*, Lecture Notes in Computer Science, 2020, pp. 491–507.

[98] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10684–10695.

[99] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, Jul. 2020.

[100] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–37, 2019.

LARS SCHMARJE received the B.S. and M.S. degrees in computer science from Kiel University, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science.

In 2016, he worked as a Project Planner for a hybrid-e-mail-infrastructure with the direkt gruppe GmbH, Hamburg. From 2017 to 2018, he worked with Vater Solution GmbH, Kiel, as an IT Security Engineer. Since 2019, he has been an Research Assistant with the Multimedia Information Processing Group, Kiel University. He is also a part of a project that builds an autonomous racing car. His current research interest includes semi-supervised learning with a focus on fuzzy labels.

MONTY SANTAROSSA received the B.S. and M.S. degrees in computer science from Kiel University, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science.

In 2019, in connection with his master thesis, he spent six months at the Daimler Research and Development, researching learned environment perception for autonomous cars. His current research interests at Kiel University include image-based diagnosis and prognosis of eye diseases, and multi-modal image understanding tasks, including deep-learning-based image registration, classification, and segmentation.

Mr. Santarossa won the Prof. Dr. Werner Petersen Preis der Technik 2019 in the Category Best Master Thesis.

SIMON-MARTIN SCHRÖDER received the B.Sc. and M.Sc. degrees in computer science from Kiel University, Kiel, Germany, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in computer science.

His primary area of expertise is deep learning and image recognition and he is working on the recognition of plankton images. His research interests include supervised and unsupervised deep learning and the application of machine learning methods in natural sciences.

REINHARD KOCH (Member, IEEE) received the Diploma and Ph.D. degrees in electrical engineering from the University of Hannover, Germany, in 1985 and 1996, respectively.

After postdoctoral research at KU Leuven, Belgium, he joined the Department of Computer Science, Kiel University, Germany, as a Professor, in 1999, where he is currently the Director of the Department of Computer Science and the Vice Dean of the Faculty of Engineering. He is the author or coauthor of over 250 peer-reviewed publications. His research interests include 3D computer vision and object tracking to multi-view analysis, light-field processing, computer graphics, augmented reality applications, and deep learning approaches to scene understanding and applications. He received numerous awards, including the Olympus Award for Pattern Recognition, in 1997, and the David Marr Price at ICCV 1998. He also serves as the President for the German Association for Pattern Recognition (DAGM) and a German Delegate for the Governing Board of the International Association for Pattern Recognition (IAPR).