Neural Computing and Applications (2022) 34:6067–6083
https://doi.org/10.1007/s00521-021-06753-6

ORIGINAL ARTICLE

Open set task augmentation facilitates generalization of deep neural networks trained on small data sets

Wadhah Zai El Amri[1] · Felix Reinhart[2] · Wolfram Schenck[1]

Received: 12 April 2021 / Accepted: 10 November 2021 / Published online: 9 December 2021

Abstract

Many application scenarios for image recognition require learning of deep networks from small sample sizes in the order of a few hundred samples per class. Then, avoiding overfitting is critical. Common techniques to address overfitting are transfer learning, reduction of model complexity and artificial enrichment of the available data by, e.g., data augmentation. A key idea proposed in this paper is to incorporate additional samples into the training that do not belong to the classes of the target task. This can be accomplished by formulating the original classification task as an open set classification task. While the original closed set classification task is not altered at inference time, the recast as open set classification task enables the inclusion of additional data during training. Hence, the original closed set classification task is augmented with an open set task during training. We therefore call the proposed approach open set task augmentation. In order to integrate additional task-unrelated samples into the training, we employ the entropic open set loss originally proposed for open set classification tasks and also show that similar results can be obtained with a modified sum of squared errors loss function. Learning with the proposed approach benefits from the integration of additional "unknown" samples, which are often available, e.g., from open data sets, and can then be easily integrated into the learning process. We show that this open set task augmentation can improve model performance even when these additional samples are rather few or far from the domain of the target task. The proposed approach is demonstrated on two exemplary scenarios based on subsets of the ImageNet and Food-101 data sets as well as with several network architectures and two loss functions. We further shed light on the impact of the entropic open set loss on the internal representations formed by the networks. Open set task augmentation is particularly valuable when no additional data from the target classes are available—a scenario often faced in practice.

Keywords Convolutional neural networks · Image recognition · Open set classification · Transfer learning

1 Introduction

Machine learning algorithms have been a huge success in the field of image classification, image recognition and image processing. Many of these achievements in computer vision are accomplished based on convolutional neural networks (CNN) [24]. State-of-the-art algorithms reach human-level or even superhuman performance [18]. This success is based on an enormous amount of training data. In typical application scenarios, however, much less data is available for training, which increases the risk of overfitting when transferring models with many free parameters to the target task.

✉ Wadhah Zai El Amri
wadhah.zai_el_amri@fh-bielefeld.de

Felix Reinhart
felix.reinhart@miele.com

Wolfram Schenck
wolfram.schenck@fh-bielefeld.de

[1] Faculty of Engineering and Mathematics, Center for Applied Data Science Gütersloh, FH Bielefeld – University of Applied Sciences, Bielefeld, Germany

[2] Miele & Cie. KG, Gütersloh, Germany

In this context, two additional circumstances often hamper the application of CNNs for image recognition further:

Firstly, many applications have to deal with inputs from an open set of classes while actions are supported only for a finite set of classes. Then, the identification of inputs from unknown classes becomes a requirement for the modeling, which is known under the notion of open set classification [2, 40]. For example, consider a smart camera oven that supports automatic recognition and cooking of a set of food items. In case a food item is placed inside the oven, for which a cooking program is available, the appliance shall suggest the respective program to the user. However, if the food item is not supported, i.e., "unknown" to the oven, no suggestion shall be displayed. Hence, the application needs to identify unknown food items in order to reduce the false positive rate.

Secondly, many applications require very domain-specific discrimination of inputs. This typically means that rather subtle changes in visual appearance shall be discriminated. For the oven example, this is the case for distinguishing fresh from frozen food items like pizza that look very similar but require different processing. Also in other domains, e.g., monitoring the biodiversity of a particular group of animals (birds), such fine-grained classification tasks are of high practical relevance.

Fine-grained classification in combination with rather small data sets can ultimately result in a high risk for overfitting: While the model is tuned to discriminate small changes in the inputs, training is based only on few samples.

This paper addresses these challenges by proposing a novel way how to utilize additional data for training that is not directly related to the target task. The basic idea is to rephrase the original classification task as an open set classification task. We show that this open set task augmentation can be useful even if the target application does not require rejection of inputs from unknown classes. The main idea of the paper is illustrated in Fig. 1.

In open set classification, the data can be separated into samples from a set of *known* classes, here the classes of the target task, and into samples from *unknown* classes, here the additional data. The overall goal during inference is twofold: Correct classification of the samples from known classes and correct identification of samples from unknown classes (rejection of these samples). For the purpose in this paper, we only consider the incorporation of unknown samples during training. That is, we augment the original closed set classification task with the open set task during training and investigate the impact on the generalization on the original task during inference.

While there is a range of techniques available for open set classification [16, 39–41], for our purpose it is essential that the approach actually impacts the internal representation of the model during training. This requirement rules out approaches to open set identification that solely are based on a threshold selection, e.g., to reject samples based on the distances to the nearest neighbors or neural activations of the classification layer after training. We therefore employ the entropic open set loss (EOS, [10]) originally proposed for open set classification tasks to supply the training with additional data. The EOS loss enforces the model to output small activations for all neurons within the final classification layer (such that the entropy is maximized) if the samples belong to an unknown class. Hence, the EOS loss actually shapes the internal representation of the model formed during training and is a valid candidate for the proposed approach.

Additionally, another candidate loss function for open set task augmentation is considered in this work which is an adaptation of the sum of squared errors (SSE) loss, which we refer to as open set sum squared error (OSSE). While not optimal for classification tasks, we show that this loss can also be employed for open set task augmentation by simply one-hot encoding targets for samples from known classes and providing the zero vector for the additional samples from unknown classes.

We show that incorporation of additional data from unknown classes using the EOS or OSSE can facilitate generalization of the model compared to only using the data from the known classes for training. We demonstrate that the selection of the unknown classes is not critical and argue that such data is often available, e.g., from open data sets. We analyze the impact of the EOS loss on the internal representation in more detail and show that it increases both the sparsity of the neural activations and the weight distribution of the upper network layers. We point out that the proposed open set task augmentation can theoretically be understood as regularization of neural activations in the last layer for the additional input samples from the unknown categories. We argue that open set task augmentation results overall in more selective neural activity in the higher network layers, i.e., more selective features, which is also reflected qualitatively in more expressive low-dimensional feature embeddings. Finally, our results indicate that the technique is superior to incorporating additional data from unknown classes via a so-called background class, and we show that it can easily be integrated in a transfer learning setup.

Hence, the proposed open set task augmentation is highly relevant for many image recognition applications for which only limited training data is available and that require rather complex models for fine-grained classification. Incorporation of "unknown" (task-unrelated) samples via a recast of the original classification task as an open set classification task can mitigate the risk of overfitting and

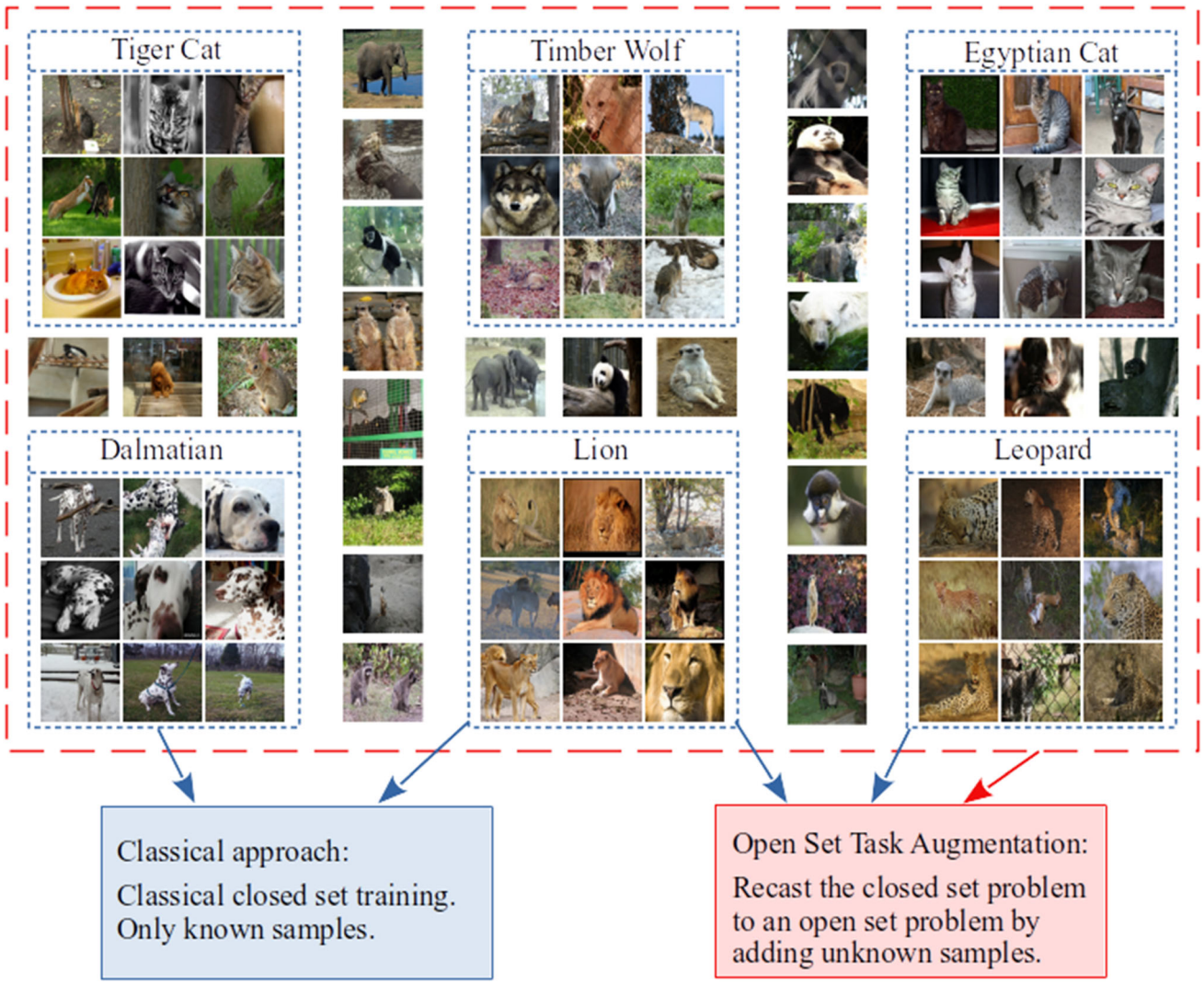

Fig. 1 Classical approach (blue box): The images within the blue rectangles represent inputs from the task-related ("known") classes that are used for model training in a classical closed set classification setup. Open set task augmentation (red box): Utilize additional samples from task-unrelated ("unknown") categories for model training by recasting the original closed set classification task to an open set classification task. In this approach, the samples from "unknown" categories (images not contained in blue rectangles) are incorporated in the model fitting in addition to the samples from the known classes (Color figure online)

ultimately improve recognition accuracy. This work therefore contributes to solving frequent challenges when applying deep neural networks to custom image recognition scenarios. Overall, the contribution of this work entails the introduction of the open set task augmentation framework including two candidate loss functions together with the empirical evaluation on multiple network architectures and data sets. This paper moreover sheds light on the impact of open set task augmentation on the learned representations.

The paper is structured as follows: We first discuss general strategies in related work to address the aforementioned challenges. Section 3 introduces the methodological basis of the paper. Section 4 introduces the experimental setup that is meant to be representative with respect to many application scenarios for image recognition. Section 5 presents the main results of the paper and investigates details concerning internal representations, choice and required amount of "unknown" data. Section 6 discusses further aspects and concludes the paper.

2 Related work

In this section, we discuss general strategies to deal with the typical challenges when training deep neural networks from small-sized data sets.

2.1 Overfitting

Overfitting is a central phenomenon in machine learning and describes the undesired state that the model is accurately fit to the training data but cannot generalize well to novel data. When fitting models with high model complexity, i.e., many free parameters, using too few samples, the trained model will likely suffer from overfitting.

This often occurs in modern image recognition application scenarios where data is rather parsimonious (in the order of few hundred samples per class) but high-dimensional (high-resolution multi-channel images) at the same time. Convolutional neural networks (CNNs) or otherwise deep neural networks are nowadays the model of choice for such applications when domain-specific feature engineering, e.g., based on computer vision techniques, is not sufficiently available in order to preprocess the raw inputs. While CNNs are the state of the art in many image recognition tasks [21, 22, 25], they come with many free parameters quickly in the range of millions. Hence, modeling in these scenarios does basically mean to trade off model complexity, which excellent deep feature extractors have, with the means to mitigate overfitting.

A multitude of approaches are available to avoid overfitting. We refer to further discussions on how to tackle overfitting, e.g., in [15, 47]. We here only focus on a few selected approaches that are often applied in the context of gradient-based training of deep neural networks.

Early stopping of the training when using gradient-based learning algorithms is a common way to prevent overfitting that has already been used in the 1970s [5, 37]. Another approach to avoid overfitting is to reduce the network complexity. This can be accomplished at the time of network design by choosing smaller layer sizes or fewer layers. Model complexity can also be reduced at or after training by so-called pruning techniques [26]. For instance, pruning can be accomplished by ranking the neural weights by their importance w.r.t. to the input–output mapping and then removing the weights that are ranked lowest [31]. Moreover, a frequently applied technique to avoid overfitting structurally is by introducing a bottleneck behind the feature representation of the pretrained network [34]. A bottleneck layer does effectively reduce the number of features used for the final classification stage. A bottleneck is often used in deep networks, e.g., in [10, 38].

Regularization is another core concept put forward to tackle overfitting. A canonical regularization technique is based on a L^2 penalty for model parameters in the loss function. In gradient-based learning, this penalty results in a weight decay term for the weight update which forces the model parameters to be smaller in norm. This effectively reduces the model complexity by restricting the model to learn smoother input–output mappings. By doing this, the network can better generalize to new data and avoids overfitting [23, 32].

Another form of regularization penalizes large neural activations [15]. In contrast to classical weight regularization, this activation regularization favors sparse over dense representations which is advantageous for classification [36]. It is therefore also referred to as sparse feature learning or representation regularization. In contrast to regularization of activations within the hidden layers of a network and all samples, open set task augmentation can be understood as activation regularization in the last layer for the additional samples from the unknown categories. In fact, open set task augmentation with the open set sum squared error loss does implement an L^2 penalty of the neural activations in the last layer only for the added samples.

A very common approach to address overfitting from small data sets is to generate additional samples through data augmentation techniques. Data augmentation denotes the enhancement of the size and quality of training data [42] in particular by adding slightly modified copies of the available samples while preserving the original label [44] or by adding otherwise synthesized data, etc. Hence, the essence of common data augmentation is to generate more inputs from the original data categories, e.g., with the help of a domain-specific noise model. Our approach, in contrast, does add completely new input samples from other, task-unrelated classes together with a dedicated schema to formulate targets for these inputs. Hence, it does not solely generate samples from the available source data from the original categories as in common data augmentation, but adds additional source samples from other classes and additionally devises a strategy how to provide output targets for these samples. Hence, it does augment the task, not solely the input data. Nevertheless, common data augmentation techniques can and should be applied in addition to the proposed open set task augmentation method at the same time.

It has also been shown that extending the original classification task to a multi-task learning setup using a single network can improve generalization from small data sets [1, 4, 27]. Here, the rationale is that the addition of related tasks provides extra supervised information that helps to form better features from which also the original target task can benefit. Hence, converting the original single classification task to a multi-task setup can be understood as another kind of task augmentation. That is, in contrast to data augmentation, where additional inputs from the same classes are generated with the help of a domain-specific noise model, task augmentation does provide additional tasks in order to improve performance

of the original target task. In fact, task augmentation is also of recent interest in meta-learning, e.g., [35, 46], where generation of additional tasks helps to later on generalize better to novel tasks. However, whereas meta-learning does target the generalization over tasks, open set task augmentation does target better generalization on the original task by incorporating additional input samples from other, task-unrelated categories into the training.

In conclusion, typical ways to tackle overfitting are based on using regularization, bottlenecks, early stopping or data augmentation techniques. In this work, we explore a way to overcome the overfitting problem by augmenting the closed set classification task with an open set classification task in order to incorporate additional data from classes that are not task-related.

2.2 Open set classification

Another way of addressing overfitting is by making more data available for the training. The main idea put forward in this paper is based on recasting the original image recognition task from small data sets to an open set classification task such that additional data not related to the original task can be incorporated into the training. We therefore briefly discuss here the main concepts of open set classification.

Image recognition is classically defined as a closed set classification (CSC) task: CSC means that the inputs, also at inference time, belong to one of a finite set of classes. In many real-world tasks, this assumption is not valid, i.e., inputs are drawn from a possibly infinite (open) set of classes. Basically, open set classification requires to identify whether an input belongs to one of the *known* classes or to an *unknown* class (open set identification). These samples can then be rejected during inference.

Dhamija et al. proposed a way to tackle the open set problem by introducing a special loss function, namely the entropic open set loss (EOS, [10]). The EOS loss is an adaptation of the cross-entropy loss which enforces the model to have only small softmax outputs in the last layer for input samples from unknown categories. Then, the maximal activation of the softmax layer is small for these inputs and can be used for open set identification by a simple thresholding mechanism. While we also employ the EOS loss, our focus lies on its utilization for the purpose of our proposed open set task augmentation scheme. Also, we introduce an additional variant of an open set loss, the open set sum squared error (OSSE).

2.3 Fine-grained classification

Fine-grained visual classification is a common task in the image recognition field. Differentiating between animal species or similar looking food items is not a trivial task to accomplish. Different prior works deal with such tasks by proposing diverse solutions. Trilinear Attention Sampling Network (TASN) [48] is a method that uses a CNN to solve such tasks. This network includes three modules, namely a trilinear attention module to localize the details presented in the input, an attention-based sampler to extract these different details, and a feature distiller to optimize the details and use them for the classification task. Bilinear CNN [29] is another related work. Two parallel CNNs are trained and each one of them extracts different features from the inputs. The different features obtained from both networks are pooled as an outer product and then piped through a linear and softmax layer in order to get the prediction of the network. This idea is later on improved by a kernel pooling method [8]. Instead of using the 2^{nd} order of the features, higher-order feature maps are extracted and used with the help of this algorithm. Higher-order feature maps give a better classification accuracy in comparison to the bilinear CNN [8]. Compact Bilinear Pooling [11] is another approach that upgrades the bilinear CNN. Instead of having high dimensions of bilinear features, this paper proposes two representations of only a few thousands of features. These two representations support back-propagation for end-to-end visual tasks.

We take a different stance here and argue that a main driver for the difficulty of fine-grained classification from small data sets is primarily due to overfitting complex models to the data. Due to the inherent characteristics of fine-grained classification tasks that similar inputs require different classifications enforces a high sensitivity of the model to changes of the input also in cases where this is undesired. Ultimately, the model does not generalize. Our contribution in this paper consists of showing that with open set task augmentation, models can still learn fine-grained, class-specific features but also generalize better to novel data.

3 Open set task augmentation

We start off from describing the targeted closed set classification task τ_c: Let $\mathcal{D}_k = \{(\mathbf{x}, c)\}$ be the set of supervised input-target pairs drawn from a finite set of known classes $\mathcal{K} = \{c_i\}, i = 1, \ldots, C$. Goal of the modeling is to classify $\hat{c}(\mathbf{x})$ for novel input samples $\mathbf{x}$ such that $\hat{c} = c$ in order to solve the closed set classification task τ_c.

Let further be $\mathcal{D}_u = \{(\mathbf{x}, \tilde{c})\}$ a set of additional input samples from task-unrelated, unknown classes $\mathcal{U} = \{\tilde{c}\}$, where $\mathcal{U} \cap \mathcal{K} = \emptyset$. The key idea of this paper is to utilize the joint set $\mathcal{D} = \mathcal{D}_k \cup \mathcal{D}_u$ for training a model that solves the closed set classification task τ_c. We refer to this

approach as Open Set Task Augmentation (OSTA). The proposed method is illustrated in Fig. 1.

We remark here that it may be also of interest from an application point of view to reject (identify) inputs from unknown classes during inference time. This open set identification task τ_o poses a binary classification problem that discriminates input samples that belong to a class $\tilde{c} \in \mathcal{U}$. For the purpose of this paper, however, we are solely interested in the impact of adding samples from task-unrelated, unknown categories to the training set on the performance of the closed set classification task τ_c.

In order to incorporate the additional samples $\mathcal{D}_u$ in the model training, we propose to use an adapted version of the cross-entropy loss [17] (log loss) function, namely the entropic open set loss (EOS, [10]). The EOS loss was originally defined as

$$J_E(\mathbf{x}) = \begin{cases} -\log(S_c(\mathbf{x})) & \text{if } \mathbf{x} \in \mathcal{D}_k \text{ is from class } c \\ -\frac{1}{C}\sum_{c=1}^{C}\log(S_c(\mathbf{x})) & \text{if } \mathbf{x} \in \mathcal{D}_u \end{cases}, \tag{1}$$

in [10], where $S_c(\mathbf{x})$ is the softmax output of the neuron c. The EOS reduces to the standard cross-entropy loss if there are no samples in $\mathcal{D}_u$. When samples from unknown categories are present in $\mathcal{D}_u$, the EOS loss (1) is minimized for these samples if all activations in the softmax output layer have small values. This corresponds to a maximum entropy distribution of the outputs for these samples, hence the name entropic open set loss.

The EOS can be rephrased as

$$\mathcal{L}_{\text{EOS}} = C^{\text{k}} + R^{\text{u}}, \tag{2}$$

where $C^{\text{k}} = -\sum_{\mathbf{x}\in\mathcal{D}_k}\log(S_c(\mathbf{x}))$ is the cross-entropy loss for all samples from known categories in $\mathcal{D}_k$ and $R^{\text{u}} = -\frac{1}{C}\sum_{\mathbf{x}\in\mathcal{D}_u}\sum_{c=1}^{C}\log(S_c(\mathbf{x}))$ can be interpreted as activation regularization for samples from unknown categories.

Analogously, we define the open set sum of squared errors loss function (OSSE) by

$$\mathcal{L}_{\text{OSSE}} = E^{\text{k}} + E^{\text{u}}, \tag{3}$$

where $E^{*} = \frac{1}{2}\sum_{\mathbf{x}\in\mathcal{D}_*}\sum_{c=1}^{C}(y_c - \hat{y}_c(\mathbf{x}))^2$ is the sum of squared errors, $\mathbf{y} = (y_1, \ldots, y_C)^T$ is the vector with target activations, and $\hat{\mathbf{y}}(\mathbf{x})$ the respective network output in the last layer. E^{k} in (3) computes the sum of squared errors for samples from known categories in $\mathcal{D}_k$, where $\mathbf{y}$ are vectors with one-hot encoded targets. For the samples from unknown categories in $\mathcal{D}_u$, we set the target activations $\mathbf{y} = \mathbf{0}$ to zero. Then, the term E^{u} in (3) reduces to $\frac{1}{2}||\widehat{\mathbf{Y}}||^2$, where $\widehat{\mathbf{Y}}$ is the matrix composed of all output vectors $\hat{\mathbf{y}}$ for samples in $\mathcal{D}_u$ from the unknown categories. Hence, the term E^{u} in (3) can be understood as activation regularization of the output layer for these samples similar to R^{u} in (2).

While we do not propose to use the sum of squared errors E^{k} for classification in practice, we investigate in this paper whether the principle benefit of open set task augmentation can also be observed with an alternative loss function different to the EOS loss. For this purpose, we compare in the following experiments the results of training networks with or without open set task augmentation. For the sake of brevity, we use the following notion in the subsequent sections: We denote models that are trained only on a set of samples from the known classes $\mathcal{K}$ as "models trained only with known data" and, likewise, models that are trained on samples also from the unknown classes as "models trained with known and unknown data."

4 Experimental setup

Our experimental setup aims at showing that training with additional data from non-task-related classes (samples from "unknown" classes in the terminology of open set classification) can boost the performance of deep classifiers and reduce the overfitting of the network. We therefore devise an experimental setup that is in many respects representative for typical image recognition application scenarios, including the scenario for the recognition of food items in domestic ovens as outlined in the introduction. The main experimental design features the following properties:

- **High-Dimensional Data:** 3-channel RGB images of size 224×224.
- **Data Set Size:** A rather small sample size of $|\mathcal{D}_k| = 23.525$ images from known classes and $|\mathcal{D}_u| = 17.289$ images from unknown classes.
- **Fine-grained Classification:** $\mathcal{K}$ comprises $C = 29$ known classes that are similar in visual appearance (details below).
- **Backbone Network:** ResNet-50 [12], a common network architecture for transfer learning.

In addition, we conduct further experiments with two other backbone networks (MobileNet [13] and EfficientNet-B4 [43]), another classification task based on the Food-101 [3] data set, and also compare the two different loss functions EOS and OSSE in order to test the robustness of the open set task augmentation scheme.

4.1 Task and data set

We use a subset of the database ImageNet [9] for the majority of our experiments. The choice of the

subcategories for the known and unknown sets is done manually. We picked classes that are visually and biologically similar. We use different carnivore subcategories for the known set (i.e., canidae and felidae) posing a rather fine-grained classification scenario. For example, the picked classes entail different cat, dog, wolf breeds, etc. The set of known classes $\mathcal{K}$ comprises $C = 29$ subcategories[1] in total with overall 23.525 images.

For the set of samples $\mathcal{D}_u$ from unknown classes, images of other animal categories are used, e.g., monkeys, elephants, rabbits, etc. We picked 22 subcategories[2] as unknown categories $\mathcal{U}$ with 17.289 images in total. We denote the data with these unknown categories as "close to the domain."

In a later experiment, we substitute the unknown data $\mathcal{D}_u$ with other, far from domain data. These classes belong mostly to the "Misc" group according to the ImageNet taxonomy in contrast to the animal group used in the other experiments of this work. We manually picked 23 far from domain subcategories[3] that are for example objects of daily life such as acoustic guitar, electric guitar, hammer and car mirror. Hence, these far from domain, unknown classes are visually not connected to the known or the unknown classes. The total size of the picked images is equal to 17.193. We denote this data as "far from domain."

The sample size for each class used in this work varies between 304 and 1.165 samples with an average of 811 instances. These sizes are meant to correspond with the ranges that are typically apparent in machine learning application scenarios with small-sized data sets. The known and unknown data (close and far from domain) are split into training (80%), validation (10%) and test set (10%). This split is identical for all conducted experiments.

Inputs are preprocessed by resizing the RGB images to a fixed size of 224×224 pixels. Moreover, we performed a common, randomized data augmentation step. Images are vertically flipped (left to right) with a probability of 50%. Pictures are then shifted up to 20 pixel rows/columns left, right, up or down with a probability of 12.5% each. Next, images are rotated (by 90°, 180° or 270°) with a probability of 12.5% each. Finally, a Gaussian blurring is applied with a kernel size of 5×5 and 50% probability.

4.2 Network training

In the majority of the conducted experiments, a residual network (ResNet) [12] is used as a backbone network. Mainly, networks with 50 residual layers are used (ResNet-50). However, for the evaluation of specific aspects, networks with 18 residual layers (ResNet-18) are also trained. Network weights are randomly initialized if not otherwise stated.

In this work, gradient descent is conducted using the Adam optimizer [19] with a learning rate of 0.001. We use a mini-batch size of 64 and train the networks for 150 epochs. In each iteration, we validate the model performance on samples from the known categories $\mathcal{K}$ only by means of the accuracy (percentage of correctly classified known samples). This corresponds to an evaluation of the model solely on the closed set classification task τ_c. After training, we select the model from the epoch with the maximum accuracy on the validation set.

Hyper-parameters like learning rate and number of epochs have been manually tuned such that the gradient descent displays a clear convergence to a local minimum in a typically shaped learning curve. All experiments are repeated three times each in order to account for random factors in model training.

5 Results

In the following sections, we present the experimental results that target at answering the following questions:

- Section 5.1: How does the utilization of unknown samples compare to the baseline at which only known samples are used for training?
- Section 5.2: How does the amount of unknown data impact the training?
- Section 5.3: How critical is the choice of the unknown data?
- Section 5.4: How does the approach relate to the choice of the model complexity / representational capacity?
- Section 5.5: Does the observed benefit of open set task augmentation transfer to other tasks, network architectures and loss functions?
- Section 5.6: How does the proposed approach compare to using a background class?

[1] Synset subcategories of ImageNet [9] picked as known categories: n02123159, n02509815, n02124075, n02123394, n02123045, n02123597, n02497673, n02111277, n02110341, n02109961, n02106662, n02116738, n03218198, n02129165, n02125311, n02129604, n02128385, n02128757, n02128925, n02114548, n02114712, n02114855, n02114367, n02119022, n02120079, n02120505, n02119789, n02127052, n02117135.

[2] Synset subcategories of ImageNet [9] picked as unknown categories close to the target domain: n02134418, n02510455, n02134084, n02132136, n02133161, n02441942, n02447366, n02444819, n02445715, n02508021, n02137549, n02138441, n02325366, n02328150, n02494079, n02493509, n02486261, n02488702, n02489166, n02484975, n02504458, n02504013.

[3] Synset subcategories of ImageNet [9] picked as far from domain: n02676566, n03272010, n03481172, n02965783, n03895866, n04037443, n02814533, n03393912, n02918964, n04005630, n03376595, n02727426, n04081281, n03032252, n03379051, n01484850, n03180011, n03793489, n03642806, n03085013, n01770393, n11955896, n04555897.

– Section 5.7: Does the proposed approach contribute to common transfer learning setups?

5.1 Training with known and unknown data

5.1.1 Training results

First of all, we empirically investigate the impact of using unknown data along with the known data. Can the additional data improve the accuracy of the network and reduce overfitting in comparison to simply training on samples from the known categories?

For this purpose, two networks with the same architecture were trained. The first network uses only the known training set and a cross-entropy loss function. The second network uses in addition the unknown training set and the entropic open set (EOS) loss.

Table 1 shows an improvement of the accuracy on the test set when incorporating the additional samples from the unknown categories in the training. The average improvement is **3.5%**. Based on these results, we deduce that auxiliary unknown data can improve generalization performance of networks using the entropic open set loss.

5.1.2 Impact of training on internal representation

In a next step, we analyze the internal representation of the networks. For this purpose, we calculated the Hoyer sparseness measure [14] for the last average pooling layer for each sample of the known test set. Hoyer's sparseness measure is defined by

$$\text{sparseness}\ (\mathbf{x}) = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\Sigma x_i^2}}{\sqrt{n} - 1}, \tag{4}$$

where x_i is the activation of the ith neuron in the layer and n the size of the layer. In our case, $\mathbf{x}$ is the last average pooling layer, which has 2.048 neurons.

Figure 2 shows that the neural networks trained on known and unknown data display more sparse neural activations in the last average pooling layer. Networks trained with the additional samples from unknown classes with the help of the entropic open set loss display significantly more sparse neural activations in this layer (two-sided Kolmogorov–Smirnov test comparing the combined sparsity value distribution from all runs trained with known data with the distribution from all runs trained with known and unknown data; $p < 0.001$).

We further investigate the distribution of the neural activations for each training run for the last two layers in more detail. Figure 3 displays the distribution of the neural activations for the last average pooling layer (second-last layer) on all known samples from the test set with a logarithmically scaled ordinate axis. The results in Fig. 3 show that the activations in the average pooling layer are more sparsely distributed for the networks trained with the known and unknown data. This observation is consistent over all runs and corresponds to the Hoyer sparsity values.

Figure 4 displays the distribution of the neural activations for the last dense layer (before softmax). The impact of the additional data together with the entropic open set loss is also for this layer evident for all conducted runs of the experiments: The peak of the neural activations is more narrow and located at smaller absolute activation values for these networks.

Overall, these results indicate that the additional data together with the entropic open set loss systematically impacts the internal representation of the classifier networks to form more sparse encodings. Sparse representations are known to be beneficial in neural processing [28, 33] and can facilitate class separability (e.g., [45]). The increased sparseness of the neural representation can be explained by the term R^u in (2) which acts as activation regularization for the samples $\mathcal{D}_u$ from unknown categories. Interestingly, R^u affects the overall sparseness of the neural representation also for the samples from known categories as evident from Figs. 2, 3 and 4.

We assess the impact of the additional data on the formed representation qualitatively by visualizing the feature vectors. The original feature vectors of the second-last layer (average pooling layer) have 2048 components. We use t-Distributed Stochastic Neighbor Embedding (t-SNE, [30]) in order to visualize the high-dimensional feature vectors in two dimensions. Figure 5 delivers qualitative information about the internal representation formed by the different networks. It shows the embedded feature vectors for all test samples from the known classes. From the second row of Fig. 5, it can be clearly seen that feature vectors from different classes can be better clustered when networks are trained with additional unknown data compared to networks which are trained solely with samples from the target classes (cf. upper row of Fig. 5). Hence,

Table 1 Test accuracies of networks on samples from the known classes

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	64.7%	64.6%	65.8%	**65.0%**
Known and unknown data	68.7%	68.9%	67.8%	**68.5%**

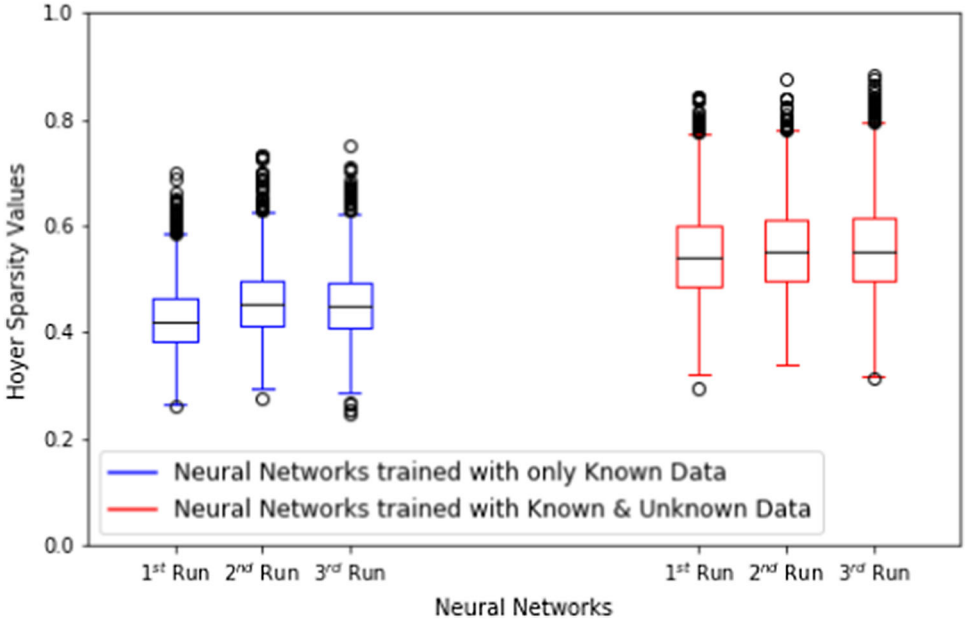

Fig. 2 Hoyer sparsity values of the last average pooling layer for all known samples in test set

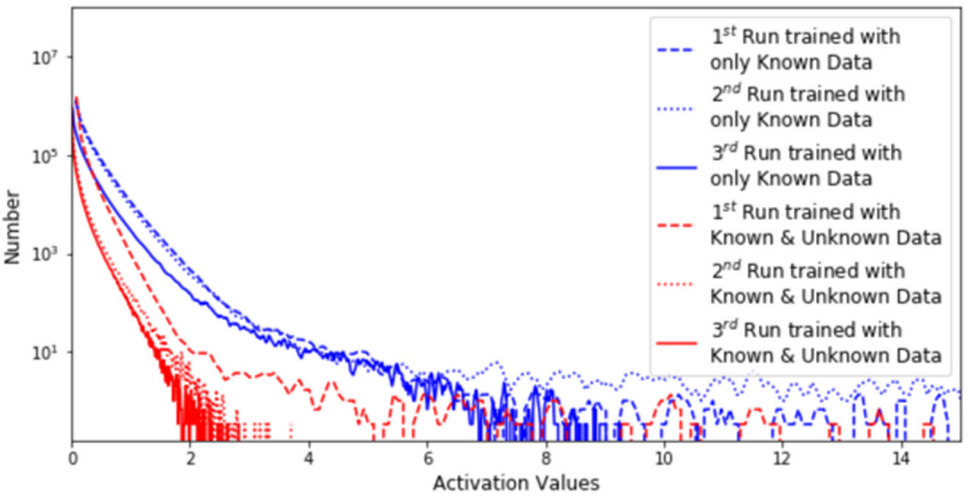

Fig. 3 Distribution of the activation values of the last average pooling layer for all known samples in test set. The combined activation value distribution of all runs trained with known data differs significantly from the combined distribution of all runs trained with known and unknown data (two-sided Kolmogorov–Smirnov test; $p < 0.001$)

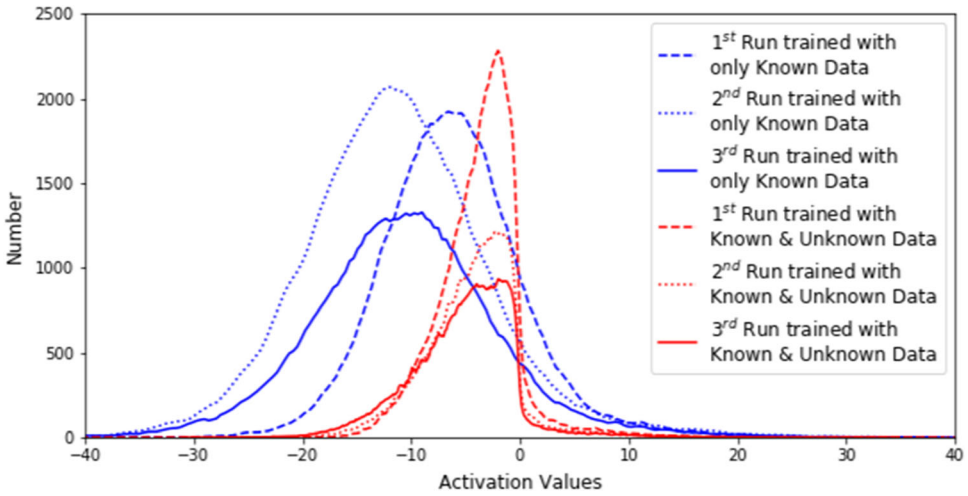

Fig. 4 Distribution of the activation values of the last dense layer (before softmax) for all known samples in test set. The combined activation value distribution of all runs trained with known data differs significantly from the combined distribution of all runs trained with known and unknown data (two-sided Kolmogorov–Smirnov test; $p < 0.001$)

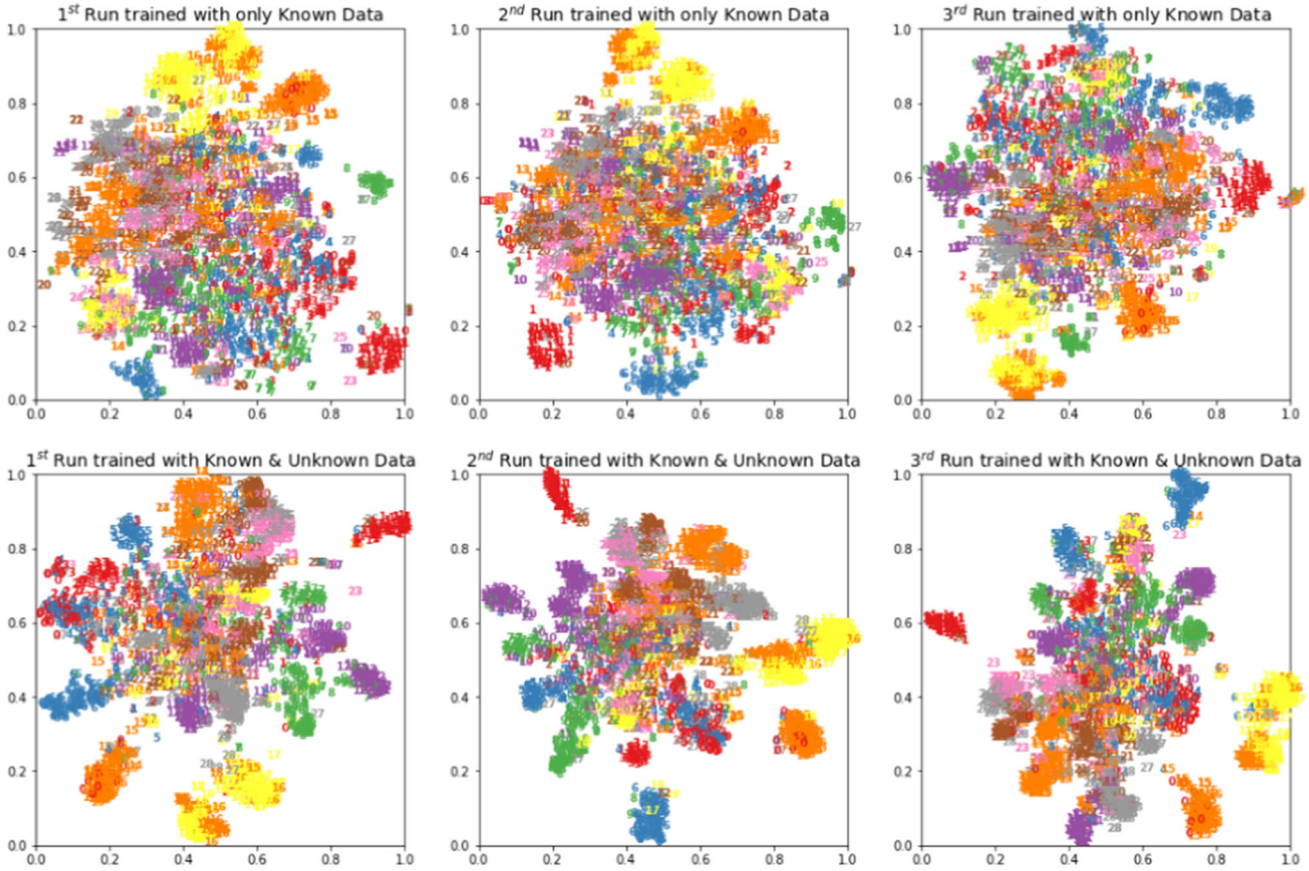

Fig. 5 T-SNE embeddings of the neural activations in the last average pooling layer for all known samples in test set. Ground truth class labels are color-coded. Top row: Embeddings for three networks trained only with samples from the known classes. Bottom row: Embeddings for three networks trained with additional samples from unknown categories.

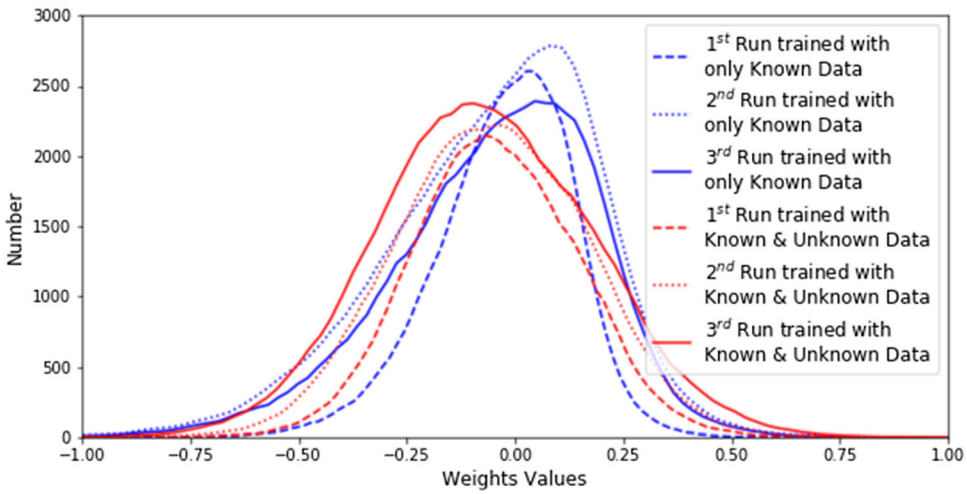

Fig. 6 Distribution of the weights of the last dense layer. The combined weight distribution of all runs trained with known data differs significantly from the combined distribution of all runs trained with known and unknown data (two-sided Kolmogorov–Smirnov test; $p < 0.001$)

Fig. 5 indicates that the additional data together with the entropic open set loss does impact the learned representation of inputs such that separability of classes in the top feature layer is fostered.

Finally, we investigate the data distribution of the weights between the second-last and last layer (2.048×29 matrix). Figure 6 shows the weight distribution for the different runs. Also, here we observe a distinct shape of the weight distributions for the two different training

conditions. The networks trained with known and unknown data have more inhibitory connections to the last layer, which may be due to the effective activation regularization of the entropic open set loss: It enforces the network to have smaller neural activations in the output layer for samples from unknown classes.

5.2 Increasing the amount of unknown data

In practice, it is often not easily possible to increase the amount of labeled data for the known classes. Increasing the amount of unknown data in open task augmentation is in contrast more easily possible. However, it remains open how many samples from unknown categories are required to improve the accuracy on the primary target task τ_c.

In order to investigate the impact of the sample sizes of the unknown data, we trained networks with different amounts of unknown training data. We consider four additional experimental conditions where we use 10%, 25%, 50% or 75% of the unknown data used in the previous experiments. Table 2 shows the results for each training run in detail. The results are also displayed with error bars over the different training runs in Fig. 7. The results in Table 2 and Fig. 7 show that the network training already benefits from a rather small amount of additional data.

5.3 Impact of choice of the unknown samples

Besides the amount of samples from unknown categories, it is of interest how the characteristics of these additional samples impact the training results. To shed light on this question, we replace the close of domain unknown data, used in the training in the preceding sections, with far of domain unknown data (introduced in Sect. 4.1). The other parameters of the experimental settings remain unchanged.

The results are presented in Table 3 and confirm the previous results: Also with unknown data more far from the domain of the target task, training benefits from this additional data and shows significantly increased accuracy on the test set (improvement of **+4.0%**). It seems to be actually more beneficial to use this far from domain data compared to the close to domain data from the previous experiments even though the difference is small (cf. results from Tables 1 and 3). However, it remains open which characteristics of the unknown data are beneficial for the training.

5.4 Reducing the representational capacity

In this section, we investigate how the proposed approach relates to other means to improve generalization from small data sets. Firstly, one could argue that the ResNet-50 simply overfits the data due to the large model complexity and the additional data just acts as a kind of regularization. Thus, the question is if the additional data is also helpful in case of a smaller network which does not require as much regularization as the large network for proper generalization. Secondly, the representational complexity of the large ResNet-50 could also be reduced by introducing a bottleneck layer before the classification stage. The paper [10] that originally introduced the entropic open set loss did actually report results only for networks with a bottleneck layer. From this previous work, it remained open whether the bottleneck is an important ingredient for the entropic open set method to work properly. One can already conclude from the results in the preceding sections that this is not the case. However, the open question is if the combination of bottleneck and additional data yields even better results than additional data alone.

We therefore first trained smaller ResNet-18 networks [12] with 18 residual layers with both conditions (only known and known plus unknown data). The ResNet-18 has obviously a significantly reduced model complexity with a smaller number of free parameters compared to the ResNet-50. The ResNet-18 has approximately 11 million free parameters, whereas the ResNet-50 has approximately 23 million free parameters. The results in Table 4 show that the reduced model complexity indeed increases the generalization accuracy when trained from known data only (cf. results for ResNet-18 and ResNet-50 trained on known data only). This confirms the typical issue of overfitting when training large networks from small, fine-

Table 2 Test accuracies of networks on samples from the known classes

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	64.7%	64.6%	65.8%	**65.0%**
Known and 10% of the unknown data	66.6%	66.3%	68.8%	**67.2%**
Known and 25% of the unknown data	67.2%	69.1%	69.9%	**68.7%**
Known and 50% of the unknown data	69.2%	68.4%	68.2%	**68.6%**
Known and 75% of the unknown data	68.3%	70.8%	67.7%	**68.9%**
Known and all unknown data	68.7%	68.9%	67.8%	**68.5%**

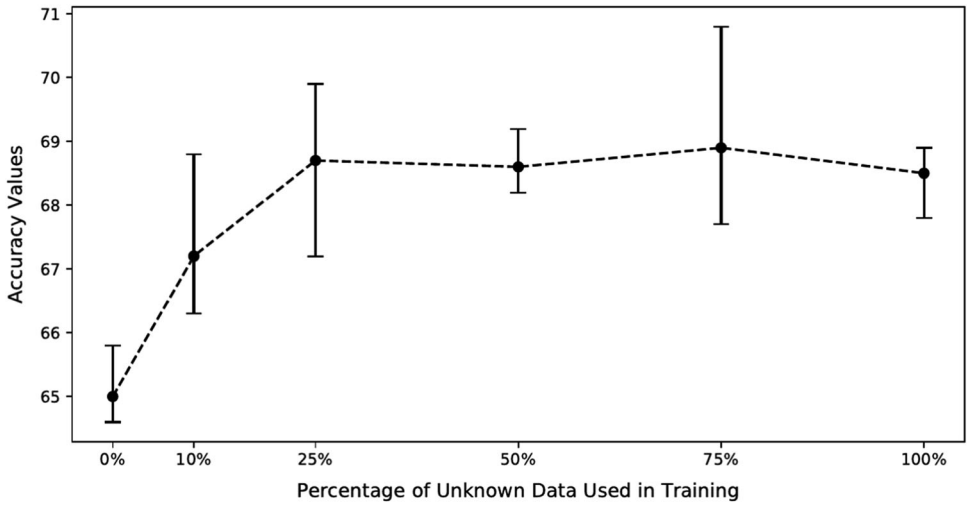

Fig. 7 Test accuracies for networks trained with different amounts of unknown data. The minimum, mean and maximum accuracy are depicted for each percentage of unknown data

Table 3 Test accuracies of networks on samples from the known classes

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	64.7%	64.6%	65.8%	**65.0%**
Known and unknown data (far from domain)	69.8%	68.2%	69.1%	**69.0%**

Table 4 Test accuracies for ResNet-18 networks

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	68.6%	68.0%	70.0%	**68.9%**
Known and unknown data	69.6%	69.0%	69.7%	**69.4%**

grained data sets. However, already with this smaller network, we observe a slight improvement of the generalization ability when the additional unknown data is incorporated into the training.

Another way of restricting the representational capacity of a model is to introduce a bottleneck layer before the classification stage, while the previous network layers may still be of larger size. To illustrate this, we introduce a bottleneck layer with 5 neurons to the ResNet-50 between the average pooling layer (2.048 neurons) and the last dense layer. An illustration of this network can be found in Fig. 8. We manually selected the size of the bottleneck layer by conducting experiments with 2, 3, 5 and 10 neurons in the bottleneck beforehand. A bottleneck of 5 neurons was optimal under the tested bottleneck sizes with regard to the validation accuracy.

The results of the trained ResNet-50 networks with bottleneck (5 neurons) are presented in Table 5. Indeed the bottleneck does effectively help to prevent overfitting by restricting the feature representation for classification to, e.g., 5 dimensions when training only on the known data (e.g., compare results for ResNet-50 trained only on known data in Table 1 without bottleneck and with bottleneck in Table 5). But also for the larger networks with bottleneck the additional data significantly improves the generalization performance (Table 5).

Overall, these results show that the proposed open set task augmentation does consistently improve generalization also under conditions with less complex models or

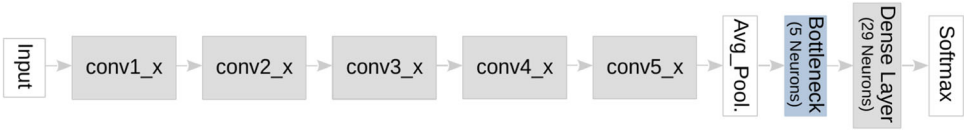

Fig. 8 ResNet architecture derived from the original ResNet paper [12] and edited by adding a bottleneck of 5 neurons before the last layer

Table 5 Test accuracies for ResNet-50 networks with bottleneck (5 neurons)

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	69.3%	70.5%	69.6%	**69.8%**
Known and unknown data	72.8%	70.6%	72.2%	**71.9%**

Table 6 Mean test accuracies averaged over three independent training runs for different network architectures (ResNet-50, MobileNet, EfficientNet-B4) and loss functions (EOS, OSSE) on the ImageNet subset introduced in Sect. 4.1

Model	Loss function	Training data	Accuracy	
			Mean value	Difference
ResNet-50		Known data only	**65.0%**	
	EOS	Known and unknown data	**68.5%**	**+3.5%**
		Known data only	**61.4%**	
	OSSE	Known and unknown data	**67.3%**	**+5.9%**
MobileNet		Known data only	**70.1%**	
	EOS	Known and unknown data	**73.9%**	**+3.8%**
		Known data only	**71.9%**	
	OSSE	Known and unknown data	**74.7%**	**+2.8%**
EfficientNet-B4		Known data only	**73.2%**	
	EOS	Known and unknown data	**75.2%**	**+2.0%**
		Known data only	**73.5%**	
	OSSE	Known and unknown data	**75.0%**	**+1.5%**

reduced representational complexity (bottleneck). We emphasize the practical relevance of being able to utilize the additional model complexity when training models from small data sets by simply adding task-unrelated data to the training.

5.5 Robustness of observed effect

This section shows that the benefit of open set task augmentation (OSTA) observed in the previous experiments does also transfer to other tasks, network architectures and loss functions.

For this purpose, we train two other frequently used networks, namely MobileNet [13] and EfficientNet-B4 [43], on the previous task setup. In addition, we not only employ the EOS loss in training, but also the OSSE loss in order to show that the concept of open set task augmentation does not entirely depend on the EOS loss function. All results are reported in Table 6. Mean values in Table 6 are calculated over three trials for each experiment. Table 6 shows that the generalization ability of the trained networks can also benefit from open set task augmentation when using the OSSE loss (cf. condition "known data only" versus "known and unknown data" when using the OSSE loss). Also, this positive effect of open set task augmentation does show independent of the chosen network architecture in Table 6.

Moreover, we investigated the robustness of the OSTA method on another classification task based on a subset of the Food-101 [3] data set. For the known classes we picked 20 food categories[4]. The training set comprises 16.000 images. Test and validation set have 2.000 samples each. Another 15 food categories[5] are then picked to build the unknown categories. The task-augmented training set comprises 12.000 samples from these unknown categories.

We trained each network architecture (ResNet-50 [12], MobileNet [13] and EfficientNet-B4 [43]) three times each using the EOS loss and the OSSE loss and calculated the mean value of the accuracy. All results for the Food-101 task are presented in Table 7 and confirm the results from the previous experiments based on the ImageNet data set: OSTA can improve the generalization ability when training deep networks from small data sets for a range of network architecture.

Both result Tables 6 and 7 show that the benefit of open set task augmentation can be robustly observed under a

[4] Categories of Food-101 [3] picked as known categories: apple_pie, carrot_cake, strawberry_shortcake, cheesecake, chocolate_cake, lasagna, pizza, caesar_salad, caprese_salad, seaweed_salad, chicken_wings, fish_and_chips, french_fries, fried_calamari, macaroni_and_cheese, spring_rolls, spaghetti_carbonara, spaghetti_bolognese, steak, baby_back_ribs.

[5] Categories of Food-101 [3] picked as unknown categories: baklava, churros, donuts, falafel, ice_cream, omelette, macarons, hot_dog, paella, escargots, french_onion_soup, ramen, mussels, tacos, sushi.

Table 7 Mean test accuracies averaged over three independent training runs for different network architectures (ResNet-50, MobileNet, EfficientNet-B4) and loss functions (EOS, OSSE) on the subset of the Food-101 data set

Model	Loss function	Training data	Accuracy	
			Mean value	Difference
ResNet-50		Known data only	**73.6%**	
	EOS	Known and unknown data	**77.2%**	**+3.6%**
		Known data Only	**65.2%**	
	OSSE	Known and unknown data	**69.0%**	**+3.8%**
MobileNet		Known data only	**76.0%**	
	EOS	Known and unknown data	**79.1%**	**+3.1%**
		Known data only	**77.1%**	
	OSSE	Known and unknown data	**79.0%**	**+1.9%**
EfficientNet-B4		Known data only	**77.6%**	
	EOS	Known and unknown data	**80.3%**	**+2.7%**
		Known data only	**79.3%**	
	OSSE	Known and unknown data	**80.6%**	**+1.3%**

broader range of conditions: When open set task augmentation is applied, a consistent improvement of the generalization ability can be observed for various network architectures and under change of the loss function or the data set.

As statistical test, a four-way ANOVA was carried out on all data from Tables 6 and 7. The ANOVA takes the experimental factors (1) data set, (2) model, (3) loss function and (4) training data ("only known" vs. "known and unknown") into account to predict the accuracy as dependent variable. The sample size in each group amounted to three. We are mainly interested in the main effect of the factor training data. The full ANOVA including all interactions gave $F = 93.7, p \ll 0.001$ for this main effect, while nearly all interactions that include this factor were not significant ($p > 0.05$) with one exception: The interaction between model and training data yielded $F = 4.5, p < 0.05$. After removing all insignificant interactions from the ANOVA, these results persist at the same levels of significance.

Regarding the preconditions for ANOVA: The Levene test (using the median as center) resulted in $p = 0.78$. Therefore homogeneity of variances can be assumed. The Shapiro test (reg. the normal distribution of the residual values) was not significant at the 1% level. Therefore the assumption of a normal distribution still holds in our view. In summary, the most important preconditions for ANOVA are given.

The statistical results corroborate our claim that open set task augmentation has a consistently positive effect on classification accuracy. We finally show in the next section that the benefit of OSTA also persists when it is applied in a transfer learning setup.

5.6 Comparison of OSTA to using a background class

Another approach to incorporate samples from unknown classes into the training is by using an additional "background" class to which all the samples from the unknown classes are assigned during training [6, 7, 10]. Then, training can simply be accomplished as closed set classification task. At inference time, one then can simply discard the outputs for the background class, e.g., by removing the corresponding neuron from the output layer, if one is only interested in the closed set classification task τ_c as it is the case for our considerations in this paper.

In this section, we empirically show that the proposed open set task augmentation differs from this method. For this purpose, we trained ResNet-50 [12] networks on the ImageNet and Food-101 subsets as before, but incorporate the samples from the unknown classes as additional category with dedicated output neuron. Training then proceeds as before but now using the standard cross-entropy loss. In the inference phase, the neuron of the background class is removed.

The results presented in Table 8 show that there is a significant difference between the two conditions OSTA with EOS loss and using a background class only. The OSTA method achieves an improvement of **+1.7%** for the ImageNet data set and **+5.5%** for the Food-101 data set compared to using the background class approach (see Table 8).

This difference can be explained by the different training objectives formulated by the cross-entropy with background class and OSTA with EOS loss: In the background class condition, only the output of the neuron corresponding to the target class (one of the known classes or the unknown class) is maximized by the loss function for all

Table 8 Test accuracies of networks on samples from the known classes

Data set	Method	Accuracy				
		1st Run	2nd Run	3rd Run	Mean value	Difference
ImageNet	Background class	68.5%	67.0%	65.0%	**66.8%**	
	OSTA	68.7%	68.9%	67.8%	**68.5%**	**+1.7%**
Food-101	Background class	69.6%	74.0%	71.7%	**71.7%**	
	OSTA	77.8%	78.2%	75.6%	**77.2%**	**+5.5%**

Training is conducted with known and unknown data either by open set task augmentation with the entropic open set loss (OSTA) or by using a background class and the cross-entropy loss (Background Class). An ANOVA with the two experimental factors "data set" and "method" ("OSTA" vs. "background class") yielded significant main effects (for "method": $F = 14.6, p < 0.01$) and a nonsignificant interaction effect

samples. Outputs of the other neurons are not penalized. In the OSTA condition, any deviation from a low, constant activation level is penalized for all output neurons for samples from the unknown class while the output of the neuron corresponding to the target class is maximized only for samples from the known classes. Hence, the difference between the output activations for samples from unknown classes (constantly low) in comparison to the output activations for samples from the unknown class (as high as possible) is maximized. This favors more selective network responses to known samples and makes the neural representation more sparse as discussed earlier in Sect. 4.1. Figure 9 shows that the sparsity of the representation in the last average pooling layer is indeed significantly lower compared to the OSTA condition.

5.7 Transfer learning

In modern image recognition applications based on deep neural networks, typically transfer learning is applied. The main idea is to benefit from the well-tuned feature extractors previously trained on millions of images. These previously acquired feature extractors, i.e., deep networks, are then transferred to the target task at hand in order to reduce training time and improve the final task performance. In this section, we show that open set task augmentation fits perfectly together with this common transfer learning scheme.

We conducted the network training as before on the ImageNet subset introduced in Sect. 4.1, but now initializing the ResNets with weights pretrained on the ImageNet data set [20]. The results in Table 9 show that initializing training with a pretrained network on ImageNet is an effective way to achieve higher accuracies on the target task. More importantly, also when using transfer learning, training does benefit from the additional unknown data. For the considered task, we achieve an improvement of **2.4%** when incorporating the additional out of domain data together with transfer learning starting from the pretrained networks. We therefore conclude that open set task augmentation with entropic open set loss can improve the internal representation of the model and ultimately the final task performance also in transfer learning setups.

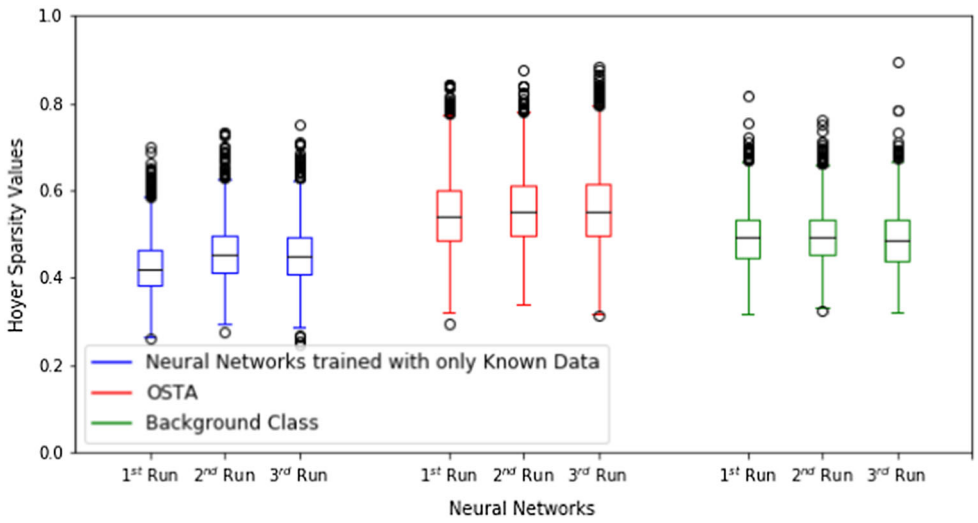

Fig. 9 Hoyer sparsity values of the last average pooling layer for all known samples in test set when training the network with known data only (blue), with known and unknown data using OSTA (red) or a background class (green). The combined sparsity value distribution of all runs using OSTA differs significantly from the combined distribution of all runs using a background class (two-sided Kolmogorov–Smirnov test; $p < 0.001$) (Color figure online)

Table 9 Test accuracies for ResNet-50 networks pretrained on ImageNet

Training data	Accuracy			
	1st Run	2nd Run	3rd Run	Mean value
Known data only	76.9%	76.9%	77.1%	**77.0%**
Known and unknown data	80.4%	79.0%	78.8%	**79.4%**

6 Discussion and conclusion

The open set task augmentation scheme proposed in this paper requires additional, auxiliary data. From the reported results, the selection of the additional data is not critical unless it stems from categories disjoint from the classes of the target task. We could so far not identify the characteristics of the unknown data which are required to support the learning. We expect that a certain similarity to the target task is important. This question points out an interesting direction for future research.

In the prior works discussed in Sect. 2.2, the core use of the unknown data is to tackle open set classification tasks. Also, the EOS loss has been originally proposed in [10] to handle open set classification. Our contribution is the introduction of the open set task augmentation scheme by utilizing, e.g., the EOS loss in order to incorporate unknown data into the task-augmented training. We empirically show that this open set task augmentation can reduce overfitting and improve the performance on closed set classification tasks.

With this approach, it is still possible to address the open set identification problem and to reject unknown samples by simply defining a suitable threshold value (e.g., see [10] for details). We therefore rate this approach as very attractive for practical application scenarios because it combines two advantages: improved generalization performance and the option to identify task-unrelated inputs during inference. The latter is relevant for the robust operation of many machine learning applications. Yet, we point out here that in this context of open set identification, the choice of the unknown classes is more intricate. For instance, picking classes, which are visually close to the domain of the known categories, can be expected to yield more strict rejection of inputs from other categories (small margin). Nevertheless, rather randomly picked unknown categories can be still expected to result in decent open set identification yet with a larger margin of accepted inputs.

In summary, this paper sheds light on an approach that can improve generalization on fine-grained classification tasks when training from small data sets. We showed that training the network with auxiliary unknown data and the EOS loss tends to make the internal network representation more sparse. The improvements achieved with the proposed approach are robust over different parameters like the network structure and the choice of unknown data. The presented method does well integrate with the state-of-the-art transfer learning strategy.

Acknowledgements WS acknowledges funding by the EFRE-NRW Funding Programme "Forschungsinfrastrukturen" (Grant No. 34.EFRE-0300119).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest There is no conflict of interest for any of the authors.

References

1. Baxter J (1997) A bayesian/information theoretic model of learning to learn via multiple task sampling. Mach Learn 28(1):7–39
2. Bendale A, Boult T (2016) Towards open set deep networks. IEEE Conference on Computer Vision and Pattern Recognition 1563–1572
3. Bossard L, Guillaumin M, Van Gool L (2014) Food-101–mining discriminative components with random forests. In: European Conference on Computer Vision
4. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
5. Caruana R, Lawrence S, Giles L (2001) Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. Adv Neural Inf Process Syst 13:402–408
6. Chang EI, Lippmann RP (1993) Figure of merit training for detection and spotting. In: Proceedings of the 6th International Conference on Neural Information Processing Systems, NIPS'93, p. 1019-1026. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
7. Chow CK (1957) An optimum character recognition system using decision functions. IRE Trans Electron Comput EC–6(4):247–254

8. Cui Y, Zhou F, Wang J, Liu X, Lin Y, Belongie S (2017) Kernel pooling for convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition pp 3049–3058
9. Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L(2009) ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition pp 248–255
10. Dhamija AR, M G, Boult TE (2018) Reducing network agnostophobia. Neural Inf Process Syst 31:9157–9168
11. Gao Y, Beijbom O, Zhang N, Darrell T (2016) Compact bilinear pooling. In: IEEE Conference on Computer Vision and Pattern Recognition pp 317–326
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition pp 770–778
13. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
14. Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. J Mach Learn Res 5:1457–1469
15. Goodfellow I, Bengio Y, Chu A (2016) Deep learning. In: Deep Learning, chap. 7, 224–270. MIT Press
16. Jain LP, Scheirer WJ, Boult TE (2014) Multi-class open set recognition using probability of inclusion. In: Computer Vision pp 393–409
17. Janocha K, Czarnecki W (2017) On loss functions for deep neural networks in classification. Schedae Informaticae 25
18. Kaiming H, Xiangyu Z, Shaoqing R, Jian S (2015) Delving deep into rectifiers: surpassing human-level performance on imageNet classification. IEEE Int Conf Comput Vision 1502:1026–1034
19. Kingma D, Ba J (2014) Adam: a method for stochastic optimization. International Conference on Learning Representations
20. Kornblith S, Shlens J, Le QV (2019) Do better imageNet models transfer better? In: IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 2656–2666
21. Krizhevsky A (2012) Learning multiple layers of features from tiny images. University of Toronto
22. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems p 25
23. Krogh A, Hertz J (1992) A simple weight decay can improve generalization. Adv Neural Inf Process Syst 4:950–957
24. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
25. LeCun Y, Cortes C, Burges CJC (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/
26. LeCun Y, Denker J, Solla S (1990) Optimal brain damage. Adv Neural Inf Process Syst 2:598–605
27. Lee T, Ndirango A (2019) Generalization in multitask deep neural classifiers: a statistical physics approach. In: Advances in Neural Information Processing Systems p 32
28. Lemme A, Reinhart R, Steil J (2012) Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. Neural Netw 33:194–203
29. Lin T, RoyChowdhury A, Maji S (2015) Bilinear CNN models for fine-grained visual recognition. In: IEEE International Conference on Computer Vision pp 1449–1457
30. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(86):2579–2605
31. Molchanov P, Tyree S, Karras T, Aila T, Kautz J (2017) Pruning convolutional neural networks for resource efficient inference. International Conference on Learning Representations
32. Ng AY (2004) Feature selection, L1 vs. L2 regularization, and rotational invariance. In: International Conference on Machine Learning p 78
33. Olshausen BA, Field DJ (2004) Sparse coding of sensory inputs. Curr Opin Neurobiol 14(4):481–487
34. Parviainen E (2010) Deep bottleneck classifiers in supervised dimension reduction. In: Artificial Neural Networks. Berlin, Heidelberg pp 1–10
35. Rajendran J, Irpan A, Jang E (2020) Meta-learning requires meta-augmentation. In: Advances in Neural Information Processing Systems pp 5705–5715
36. Ranzato Ma, Boureau Yl, Cun Y (2008) Sparse feature learning for deep belief networks. In: Advances in Neural Information Processing Systems p 20
37. Raskutti G, J WM, Yu B (2011) Early stopping for non-parametric regression: An optimal data-dependent stopping rule. In: Annual Allerton Conference on Communication, Control, and Computing pp 1318–1325
38. Roth K, Milbich T, Sinha S, Gupta P, Ommer B, Cohen JP (2020) Revisiting training strategies and generalization performance in deep metric learning. Int Conf Mach Learn 119:8242–8252
39. Rudd EM, Jain LP, Scheirer WJ, Boult TE (2018) The extreme value machine. IEEE Trans Pattern Anal Mach Intell 40(3):762–768
40. Scheirer WJ, de Rezende Rocha A, Sapkota A, Boult TE (2013) Toward open set recognition. IEEE Trans Pattern Anal Mach Intell 35(7):1757–1772
41. Scheirer WJ, Jain LP, Boult TE (2014) Probability models for open set recognition. IEEE Trans Pattern Anal Mach Intell 36(11):2317–2324
42. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):60
43. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. Int Conf Mach Learn 97:6105–6114
44. Taylor L, Nitschke G (2018) Improving deep learning with generic data augmentation. In: IEEE Symposium Series on Computational Intelligence (SSCI) pp 1542–1547
45. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
46. Yao H, Huang LK, Zhang L, Wei Y, Tian L, Zou JY, Huang J, Li Z (2021) Improving generalization in meta-learning via task augmentation. In: International Conference on Machine Learning pp 11887–11897
47. Ying X (2019) An overview of overfitting and its solutions. J Phys: Conf Ser 1168:022022
48. Zheng H, Fu J, Zha Z, Luo J (2019) Looking for the devil in the details: learning trilinear attention sampling network for fine-grained image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 5007–5016