Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.026745

ARTICLE

Filter Bank Networks for Few-Shot Class-Incremental Learning

Yanzhao Zhou, Binghao Liu, Yiran Liu and Jianbin Jiao*

School of Electronic, Electric and Communication Engineering, University of Chinese Academy of Sciences, Beijing, 140018, China

*Corresponding Author: Jianbin Jiao. Email: jiaojb@ucas.ac.cn

Received: 28 September 2022 Accepted: 14 December 2022

ABSTRACT

Deep Convolution Neural Networks (DCNNs) can capture discriminative features from large datasets. However, how to incrementally learn new samples without forgetting old ones and recognize novel classes that arise in the dynamically changing world, e.g., classifying newly discovered fish species, remains an open problem. We address an even more challenging and realistic setting of this problem where new class samples are insufficient, i.e., Few-Shot Class-Incremental Learning (FSCIL). Current FSCIL methods augment the training data to alleviate the overfitting of novel classes. By contrast, we propose Filter Bank Networks (FBNs) that augment the learnable filters to capture fine-detailed features for adapting to future new classes. In the forward pass, FBNs augment each convolutional filter to a virtual filter bank containing the canonical one, i.e., itself, and multiple transformed versions. During back-propagation, FBNs explicitly stimulate fine-detailed features to emerge and collectively align all gradients of each filter bank to learn the canonical one. FBNs capture pattern variants that do not yet exist in the pretraining session, thus making it easy to incorporate new classes in the incremental learning phase. Moreover, FBNs introduce model-level prior knowledge to efficiently utilize the limited few-shot data. Extensive experiments on MNIST, CIFAR100, CUB200, and Mini-ImageNet datasets show that FBNs consistently outperform the baseline by a significant margin, reporting new state-of-the-art FSCIL results. In addition, we contribute a challenging FSCIL benchmark, Fishshot1K, which contains 8261 underwater images covering 1000 ocean fish species. The code is included in the supplementary materials.

KEYWORDS

Deep learning; incremental learning; few-shot learning; Filter Bank Networks

1 Introduction

The enormous success of Deep Convolution Neural Networks (DCNNs) [1–3] in computer vision tasks is built upon the collection of large-scale datasets [4,5]. However, with new samples and novel classes emerging, sequential data collection is required for many tasks, e.g., classifying newly discovered fish species in the ocean. Towards lifelong learning ability like humans, deep learning models need to acquire the ability to incorporate new class knowledge incrementally, i.e., Class-Incremental Learning (CIL) [6–12].

Current CIL methods mainly focus on the setting where novel classes arise with sufficient training samples. However, the cost of collecting and labeling new samples is considerably high, and it can be impossible to gather enough new class data in real-world applications. For example, collecting photos of rare fish species in the ocean could take years. Therefore, designing effective deep learning models to incorporate knowledge from limited data and recognize novel classes sequentially, i.e., Few-Shot Class-Incremental Learning (FSCIL), has recently drawn the attention of the deep learning community [13–15]. The FSCIL models are first pretrained with some base classes with sufficient data. Then the model is updated in multiple incremental learning sessions where only a few samples of novel classes, e.g., 1-shot or 5-shots, are provided in each session. The metric of classification accuracy on all seen classes is used to evaluate the model performance of each session.

Why the naive fine-tuning of the model with new samples will not work is twofold. First, the over-parameterized deep learning models tend to overfit the biased distribution of limited few-shot training samples, resulting in poor performance of the new classes. Second, the incremental tuning damages the learned filters of the model, causing the drastic decline of old classes' classification accuracy, namely the catastrophic forgetting problem. Current few-shot learning methods augment the data of new classes through distribution sampling, e.g., FreeLunch [16], to alleviate the overfitting problem. And to resist the forgetting issue, current approaches mainly concentrate on improving the backward compatibility, which restricts the model parameters from updating in the incremental learning sessions [10–12,17–21]. Despite substantial progress, the problem of addressing the overfitting and forgetting issues in a uniform framework remains open.

In this work, we argue that solving Few-Shot Class-Incremental Learning requires endowing models with the ability to foresee the upcoming new classes and capture the variants of discriminative semantic patterns that do not yet exist in the limited training data. Take an example from software engineering. If the early version is poorly designed, it usually takes significantly more work to improve the later versions without breaking backward compatibility. On the contrary, a proper design of the early version, which foresees the upcoming features and reserves interfaces, can make it considerably easier to maintain and upgrade the software. Consequently, we concentrate on models' forward compatibility by improving the most fundamental element of Deep Convolution Neural Networks, i.e., convolutional filters.

We propose Filter Bank Networks (FBNs), Fig. 1, a simple yet effective deep learning framework to tackle Few-Shot Class-Incremental Learning (FSCIL). FBNs are built upon the commonly used Deep Convolution Neural Networks (DCNNs) and support modern model architectures, e.g., VGG, Inception, and ResNet. In the forward pass, FBNs augment each convolutional filter to a virtual filter bank containing the canonical one, i.e., itself, and multiple transformed versions, e.g., rotation, flip, or scaling. In the back-propagation, FBNs collectively gather and align gradients of all transformed versions in the filter bank to learn the canonical filter. During incremental learning, FBNs explicitly stimulate fine-detailed features to emerge through mining the instance-aware discriminations.

While conventional models only capture existing patterns in the dataset, FBNs learn semantic patterns that do not yet exist in the pertaining session, thus reserving the feature space for future new classes. During the incremental sessions, FBNs do not need to squeeze former classes' space, thus alleviating the catastrophic forgetting issue. Note that FBNs augment filters to virtual filter banks in all convolution layers and can model the transformed variants of intermediate representation at different semantic levels, e.g., bird head rotated to its body. Therefore, FBNs are more expressive than the conventional DCNNs trained with image-level data augmentation. Whatsmore, the virtual filter bank in FBNs aggregates gradients from all transformed versions in the dataset to update one canonical filter, thus can utilize the limited few-shot data more efficiently. We conduct extensive experiments and ablation studies on commonly used FSCIL datasets and report new state-of-the-art results, validating the effectiveness of FBNs. The contribution of this paper are summarized as follows:

- We propose Filter Bank Networks (FBNs) to improve the most fundamental element of DCNNs, i.e., convolutional filters. FBNs endow DCNNs with the capability of capturing variants of semantic patterns that do not yet exist in the training session, addressing catastrophic forgetting and overfitting problems of Few-Shot Class-Incremental Learning (FSCIL).
- FBNs achieve new state-of-the-art performance on several commonly used FSCIL datasets, including CIFAR100, CUB200, and Mini-ImageNet.
- We contribute a challenging FSCIL benchmark, i.e., Fishshot1K, which contains 8261 underwater images covering 1000 ocean fish species.

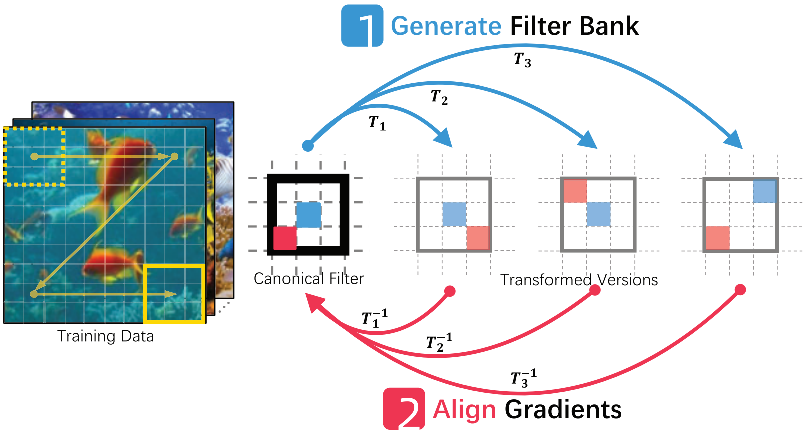

Figure 1: We propose Filter Bank Networks(FBNs) that augment each filter in the Deep Convolution Neural Networks to a virtual filter bank including itself and multiple transformed versions. The filter bank aggregate and align gradients to learn variants of semantic patterns that do not yet exist in the current training session, thus preserving feature space for adapting future novel classes in the Few-Shot Class-Incremental Learning tasks

2 Related Works

Image Classification. Image Classification [1–3,22], i.e., assigning an input image one label from a fixed set of semantic categories such as fish, cat, or airplane, is one of the fundamental problems in the Computer Vision field. Despite its simplicity, it has a lot of practical applications, and many other computer vision tasks, e.g., object detection [23,24] and scene segmentation [25,26], can be simplified to image classification. Recognizing a visual concept in the image is relatively trivial for a human to perform. However, it is considered challenging for Computer Vision algorithms due to several affecting factors. First, an instance of an object can be oriented in many ways concerning the camera, namely rotation variation. Second, visual concepts often vary in size, namely scale variation. Next, many objects are not rigid bodies and can be deformed in extreme ways, e.g., a bird's head can significantly rotate relative to its body. Moreover, occlusion, illumination conditions, and background clutter can substantially affect classification accuracy.

Handcrafted Features. Handcrafted image features have been extensively explored in the classical image classification field, e.g., SIFT [27], LBP [28,29], and Gabor features [30,31], to address those challenging variation issues in Image Classification, SIFT-like methods first detect stable feature key points that can be detected across scales. Image gray values of the local regions are then accumulated to summarize local patterns and generate the feature descriptors. The dominant orientations are found according to local gradient directions. With dominant orientation-based feature alignment, SIFT achieves invariance to rotation and robustness to moderate scale transforms. LBP emanates an invariant encoding operator against the monotonic transformation of the grayscale of local regions. LBP minimizes the encoded value via the bit cyclic shift operator based on the gray values of a circularly symmetric neighbor set of pixels in a local region. Other representative handcrafted descriptors include CF-HOG [32], which uses orientation alignment, and RI-HOG [33], which leverages radial gradient transform to be rotation invariant. Despite the progress made in handcrafted features, designing the invariant feature descriptors for different data domains and types of pattern variants is tedious and can not guarantee global optima.

Deep Convolutional Neural Networks (DCNNs). DCNNs can capture discriminative patterns from datasets in a data-driven manner to learn features that can tolerate moderate transformations in the image, such as scale changes and small rotations. DCNNs achieve those abilities through the fundamental design of convolutional operations, redundant convolutional filters, and hierarchical spatial pooling [34]. Modern architectures of DCNNs, e.g., VGG [1], Inception [2], and ResNet [3], achieve great success in image classification. However, it can be seen in the first row of Fig. 2 that the convolution filters in the DCNN can only learn patterns in the training data by rote. And they failed to generally model the variants of semantic patterns. Thus, conventional DCNNs require large-scale training datasets [4,5], which are often expensive to collect and label.

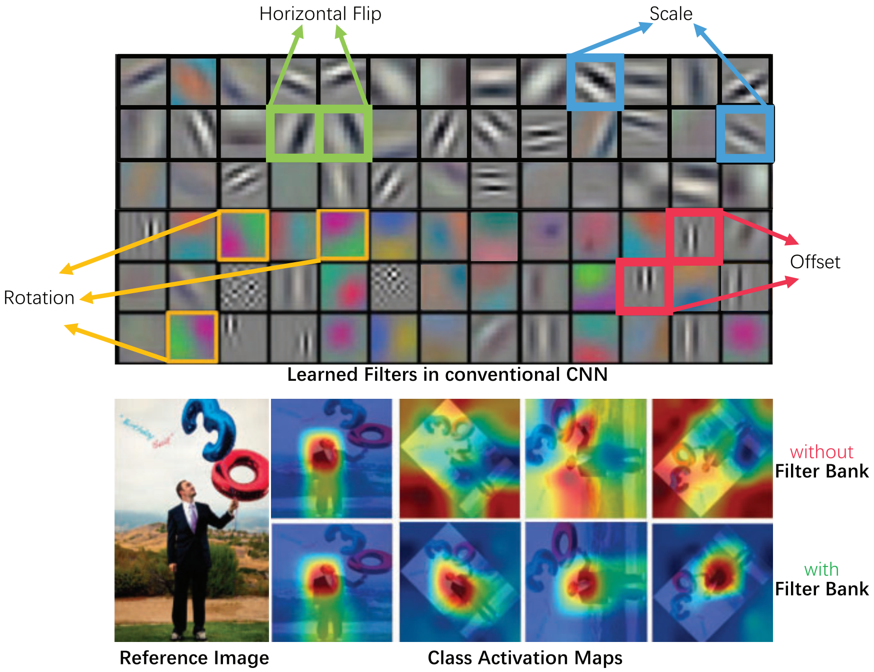

Figure 2: The first row shows the visualization of filters learned in a Deep Convolution Neural Network (DCNN). Transformed variants of patterns are redundantly learned, indicating conventional DCNNs lack the ability to generalize and require abundant data for training. The Class Activation Maps in the second row show that Filter Bank Networks align and learn from all variants of semantic patterns, thus capable of efficiently utilizing the limited few-shot data to update filters. Best viewed in color

Few-Shot Learning (FSL). Few-shot learning focuses on training models to classify novel classes with limited training samples. Conventional methods can be generally grouped into three types, i.e., Metric Learning, Meta-Learning, and Data Augmentation. Metric Learning methods [35–41] find a proper latent embedding space where the feature distances between intra-class samples are significantly smaller than inter-class samples. A typical two-branch network design is used to determine classes of the test images by comparing the feature of few-shot training samples. Meta-Learning methods [42–44] target optimizing the training process to achieve fast adaptation for new classes with limited data. Data augmentation methods [16,45–47] utilize the hidden information of abundant data of base classes to generate pseudo examples for novel classes via sampling techniques, aiming to rectify biased distributions and alleviate the overfitting issue.

Class-Incremental Learning (CIL). With the emergence of new classes, many real-world applications require the capability to incrementally adapt to novel classes, e.g., classifying newly discovered fish categories in the ocean. According to the availability of task IDs, incremental learning methods can be broadly categorized as Task-Incremental or Class-Incremental Learning. CIL methods design models to support learning from data sequence, i.e., incrementally recognizing novel semantics while not forgetting old ones. There are rehearsal, regularization, or architecture configuration methods for the Class-Incremental Learning tasks. Rehearsal methods [7–9,21,48–50] recall samples stored from a previous session to prevent the catastrophic forgetting issue. Regularization methods [10–12,17–21] introduce auxiliary training loss that utilizes prior knowledge or distillation to constrain network parameters from significant changes. Architecture configuration methods restrictedly update parts of network parameters by leveraging attention [51], pruning mechanisms [52], or dynamic expansion [6] to ease model drift.

Few-Shot Class Incremental Learning (FSCIL). The setting of conventional class incremental learning methods assumes sufficient training samples of the novel classes. However, the cost of collecting and labeling new samples is considerably high in many real-world applications. Few-shot Class-Incremental Learning amalgamates the challenges of catastrophic forgetting caused by incremental learning and overfitting caused by biased and insufficient training samples. Tao et al. [13] implemented a Neural Gas structure to address FSCIL by building and preserving feature topology. SKAD [14] designs semantic-aware knowledge distillation [17] to consolidate the features learned from base classes. Continually evolving prototypes [53,54] learn novel classes by optimizing classifier parameters to adjust the decision boundary progressively. The mixture sub-space method [55] synthesizes new samples of incremental classes in the latent embedding space via a variational auto-encoder.

The distribution calibration [16] method initiated the idea of solving the overfitting issue caused by biased distributions. However, it is nontrivial to migrate and apply those to FSCIL due to the enormous memory costs caused by sample storage. Despite substantial progress of FSCIL state-of-the-arts, the problem of solving the forgetting and over-fitting issues in a uniform framework remains open.

Forward Compatible Learning. Compatibility is a core concept in the field of software engineering. Backward Compatibility provides interoperability with an older legacy system, while Forward Compatibility allows a system to accept input intended for an updated later version of itself. The concept of Compatibility has been introduced to FSCIL in recent works [15]. Conventional FSCIL methods concentrate on improving models' Backward Compatibility by preventing the model from significant changes in the incremental sessions [11,12,17,18]. In contrast, we propose Filter Bank Networks, a simple yet effective model to improve Forward Compatibility by reserving filters for unseen patterns in future novel classes.

Augmented Convolution Filter. Previous works have studied several ways of improving convolution filters. Dilated convolution [56] insert holes between filters to enlarge the effective receptive field. Oriented Response Networks [57] introduce active rotating filters to explicitly model orientation information. Deformable convolution [58] designs learnable offsets to the sampling location of filters, improving data-specific deformation tolerance. Despite the success, those methods aim at single transformation, e.g., rotation, and require abundant data to learn auxiliary parameters, thus are unsuitable for Few-Shot Class-Incremental Learning (FSCIL). We propose Filter Bank Networks (FBNs), a uniform framework that supports arbitrary filter transformations, e.g., rotation, flip, or scaling. FBNs introduce model-level prior knowledge to effectively and sequentially learn from limited few-shot data.

Equivariant Feature Encoding. A number of previous works have addressed the problem of learning or constructing equivariant representations. For instance, transforming autoencoders [59], equivariant Boltzmann machines [60,61], and equivariant filtering [62]. The learned features of conventional Convolutional Neural Networks (CNNs) are naturally equivariant to the translation of image patterns. Nonetheless, CNNs are incapable of handling other transformations like rotations. Group equivariant Convolutional Neural Network (G-CNN) [63] is a representative work that implements equivariant feature encoding in the CNN. However, there are three distinct differences between G-CNNs and our proposed Filter Bank Networks (FBNs). First of all, G-CNNs mainly consider translation, flip, and 90 degrees rotation while FBNs use a set of transformation matrices to support more general feature equivariance such as arbitrary angle rotation. Second, the filter transformation in G-CNNs is implemented as an indices lookup procedure that can not support the change of spatial shape. FBNs introduce a sampling technique to support mixing sizes of filters, e.g., scaling. Last but not least, G-CNNs are evaluated on the regular classification task which assumes abundant training samples. FBNs are designed for the more challenging Few-Shot Class-Incremental Learning benchmark where overfitting and catastrophic forgetting issues can be caused by insufficient training samples.

3 Method

Filter Bank Networks (FBNs) are built upon Deep Convolution Neural Networks (DCNNs) via augmenting convolution filters to virtual filter banks. Each filter bank contains the canonical filter itself and transformed versions. Thus, a filter bank produces feature maps with additional channels that capture the response of variants of the semantic pattern. Note that all transformed versions are generated on-the-fly, and only one filter in the bank is materialized and learned from the aggregated gradients of the entire filter bank. Model-level prior knowledge is introduced via the design of transformations of the filter bank. Thus, FBNs learn unseen transformed intermediate representations from limited pretraining data, preserving feature space for adopting future novel classes in incremental learning sessions.

In what follows, we address the two problems in implementing FBNs for Few-Shot Class-Incremental Learning (FSCIL). First, we describe the technique of convolution with filter banks. We construct a sampling-based method to efficiently generate virtual filter banks in the forward pass. And we reverse-sampling the gradients to align and aggregate error signals to update one canonical filter in the backward pass. Next, we elaborate on the proposed FBN framework for FSCIL, Fig. 3. We jointly train FBNs with both class- and instance-level losses to stimulate the fine-detailed features to emerge for adapting future novel classes. Last but not least, we discuss the insight into the design of FBNs from the perspective of inductive bias and equivariant representation.

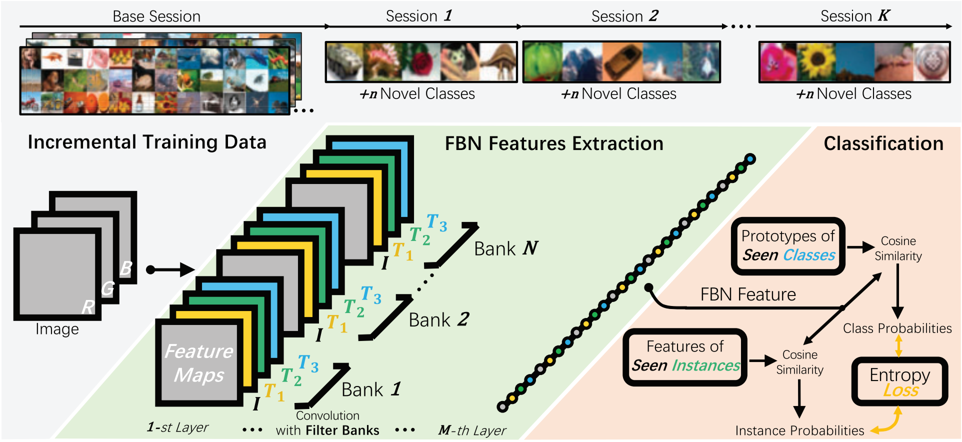

Figure 3: FBN framework

3.1 Convolution with Filter Banks

Filter Bank Generation. Filter Bank Networks are upgraded from conventional Deep Convolution Neural Networks, e.g., ResNet [3]. Without loss of generality, each augmented filter $\mathcal{F}$ in FBNs has the size of $M \times H \times W$, where M is the number of transformations $\mathcal{P} = \{P^m\}, 0 < m \leq M$, in the previous layer's filter banks. $\mathcal{F}$ transforms $N-1$ times during the convolution to produce feature maps of N channels. It can be virtually seen as a filter bank of the size $N \times M \times H \times W$, where $H \times W$ is the spatial resolution of filters, e.g., 3×3 in VGG [1] and ResNet [3]. Only the $\mathcal{F}$ part is materialized and learned, while the remaining $N - 1$ transformed versions are generated on the fly. The $n-$th version in the filter bank, $\mathcal{F}^n$, $n \in [1, N]$, is obtained by $\mathcal{F}$ and the $n-$ th generic homogenous transformation matrix T^n in a predefined set $\mathcal{T} = \{T^i\}, 0 < i \leq N$. Note that the first transformation matrix T^1 is the identity matrix, i.e., $\mathcal{F}^1 = \mathcal{F}$. Note each convolutional layer has $C_{output} \times C_{input}$ FBN filters, where C_{output} and C_{input} are the number of output and input convolution channels.

Each element of $\mathcal{F}$ and $\mathcal{F}^n$ can be accessed with $\mathcal{F}_{k,i,j}$ and $\mathcal{F}^n_{k,i,j}$, where $0 < k \leq N, 0 \leq |i| \leq \frac{H}{2}, 0 \leq |j| \leq \frac{W}{2}, k \in \mathbb{N}, i, j \in \mathbb{R}$. And the transformed version $\mathcal{F}^n$ can be derived from $\mathcal{F}$ through a two-step technique. First, the base channel index o is found where $T^n = T^k \times P^o$. Next, the location of (p, q) in $\mathcal{F}^n$ is obtained from the location of (p', q') in $\mathcal{F}$ through bilinear sampling as

$$\begin{pmatrix} p' \\ q' \\ 1 \end{pmatrix} = T^n \begin{pmatrix} p \\ q \\ 1 \end{pmatrix} T^n = \begin{pmatrix} t^n_{11} & t^n_{12} & t^n_{13} \\ t^n_{21} & t^n_{22} & t^n_{23} \\ 0 & 0 & 1 \end{pmatrix}, p, q, p', q', t^n \in \mathbb{R}, \tag{1}$$

$$\mathcal{F}^n_{o,p,q} = (1-\mu)(1-\omega)\mathcal{F}_{k,u,v} + (1-\mu)\omega\mathcal{F}_{k,u,v+1} + \mu(1-\omega)\mathcal{F}_{k,u+1,v} + \mu\omega\mathcal{F}_{k,u+1,v+1} \tag{2}$$

where $u = \lfloor p' \rfloor, v = \lfloor q' \rfloor, \mu = p' - u, \omega = q' - v$. Zero padding is used when no base channel index o can be found. Note that filters in different convolution layers are composited to form semantic patterns at different scales, e.g., edges, object parts, and objects [64]. And the above procedure can be interpreted as transforming the composited semantic patterns via changing coordinate systems of corresponding filters in all layers based on $\mathcal{T}$. The transformation matrix T^i can represent rotation

$\begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \theta \in \mathbb{R}$, scaling $\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}, s_x, s_y \in \mathbb{R}$, flip $\begin{pmatrix} h & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{pmatrix}, h, v \in \{-1, 1\}$, translation $\begin{pmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{pmatrix}, tx, ty \in \mathbb{R}$, or their arbitrary composition. Note that the same predefined $\mathcal{T}$ is used for all convolution layers and $\mathcal{P} = \mathcal{T}$ except the first layer where $\mathcal{P}_{first} = \left\{ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}$.

Transformations like scaling can alter the size of the filter. And the convolution of the mixing size of filters is not computation-friendly for modern GPUs. Therefore, we implement the generic form of convolution of filter banks based on a fixed-size sampling of input feature maps $\mathcal{M}$. For each convolution receptive field centering (r_i, r_j) of $\mathcal{M}$, we rewrite Eq. (1) by adding (r_i, r_j) as the offset to (p', q'), and we rewrite Eq. (2) by replacing the source of bilinear sampling from the canonical filter $\mathcal{F}$ to the feature maps $\mathcal{M}$. The result is then multiplied with $\mathcal{F}$ and reshaped to the output feature maps. The proposed approach can be efficiently implemented and accelerated with CUDA and the computational complexity is invariant to the change of filter shape. Note when the transformed filter sizes are consistent, the implementation can be simplified to first generate the filter bank with Eqs. (1) and (2) then convolute it with $\mathcal{M}$. In this way, the computational overhead is negligible.

Filter Bank Learning. The filter bank represents the semantic patterns of the canonical filter $\mathcal{F}$ and its transformed variants $\mathcal{F}^n$. Only $\mathcal{F}$ is materialized and learned during training. We reverse-sample the gradients of $\mathcal{F}^n$ by $(T^n)^{-1}$ to align and accumulate gradients of all transformed filter variants to update $\mathcal{F}$ through SGD optimization. The learning of filter banks is end-to-end with the rest parts of the model.

3.2 *Few-Shot Class-Incremental Learning (FSCIL)*

FSCIL contains a pretraining session and multiple incremental sessions. During the pretraining session, FSCIL methods learn a representation model to recognize some base classes $\mathcal{C}^{(0)}$ with abundant samples. During the incremental sessions, the FSCIL methods generalize the developed model to novel classes $\mathcal{C}^{(t)}$, where $t \in \mathbb{N}+$ is the session index, with only a few samples, e.g., 1-shot or 5-shots. Incremental datasets are denoted as $\{\mathcal{D}^{(t)}, t \in \mathbb{N}+\}$, where $\mathcal{D}^{(t)}$ is the samples of classes $\mathcal{C}^{(t)}$ for the t-th session. For $t_1 \neq t_2$, we have $\mathcal{C}^{(t_1)} \cap \mathcal{C}^{(t_2)} = \emptyset$. In the t-th session, only the training data from $\mathcal{D}^{(t)}$ is available, while all test data of seen classes $\{\mathcal{C}^{(0)}, \ldots ; , \mathcal{C}^{(t)}\}$ is used for model evaluation. In other words, novel classes shall be learned through limited few-shot data, while the old seen classes shall not be forgotten.

It can be seen in Fig. 3 that the proposed Filter Bank Networks (FBN) framework for FSCIL consists of two parts, i.e., a convolutional feature extractor based on FBNs and a classification head that can adapt to the incremental classes. The whole model is jointly trained with multiple losses in an end-to-end manner.

FBN Feature Extraction. To obtain a discriminative feature representation, a feature extractor is developed by upgrading a conventional Deep Convolution Neural Network (DCNN), e.g., VGG [1] or ResNet [3], to the Filter Bank Network via equipping all convolution layers with the Filter Banks. Each filter bank contains the canonical filter itself and multiple transformed versions defined by a predefined transformation matrix set $\mathcal{T} = \{T_1, T_2, \ldots ; , T_N\}$, where N is the number of variants and T_1 is identity matrix. The convolution output of each filter bank is a group of feature maps containing

N channels which correspond to the spatial response of transformed variants of an intermediate semantic pattern. Along with the convolution and pooling operation, the feature maps gradually shrink and the output of the top convolution layer is spatially averaged to form the M-dimensional feature representation $\mathcal{R}$ of the input image.

Through the multiple layers' convolution, each dimension of $\mathcal{R}$ detects a discriminative pattern in the image, such as the dog's head or the wheel of the car [64]. Conventional DCNNs can only detect patterns that exist in the training data. In contrast, the proposed FBN feature extractor captures unseen pattern variants via the filter banks, alleviating the issue of overfitting and forgetting in the classification phase.

Scalable Classification. The commonly used multi-layer perception (MLP) classifier can not be used in the FSCIL tasks as the number of classes that needs to be recognized is incremental. Therefore, we make classification scalable by allocating class prototypes $\mathbf{P} = \{P_1, P_2, \ldots;, P_K\}$, where K is the number of seen classes and it is dynamically changed in the incremental sessions. Each class prototype is a learnable M-dimentional tensor representing the class's cluster center in the latent feature space. We use random tensors to initial the class prototypes in the pretraining session and use the averaged features of few-shot training samples to initial class prototypes in the incremental sessions. Given an input image, we first extract its feature $\mathcal{R}_I$ and compute the classification logits x_k for the k-th seen classes as

$$x_k = E \cdot CosineSimilarity(\mathcal{R}_I, P_k) = E \cdot \frac{\mathcal{R}_I \cdot P_k}{max(\parallel \mathcal{R}_I \parallel \cdot \parallel P_k \parallel, \epsilon)}, \tag{3}$$

where E is a temperature hyperparameter[1] and $\epsilon = 1e^{-8}$ is to prevent division by zero. The class prototypes are jointly learned with the filter banks during the model training.

Model Learning. To train the model parameters, we compute the loss between the predicted classification logits $X_c = \{x_1, \ldots;, x_K\}$ and the ground truth class index $g_c \in [1, K]$ as

$$L_c(X_c, g_c) = -log(\frac{exp(x_{g_c})}{\sum_{k=1}^{K} exp(x_k)}), \tag{4}$$

where c indicates *classes*. Minimizing the loss $L_c(X_c, g_c)$ drives the model to capture patterns for recognizing classes, such as the wheel for cars or the wings for birds. However, previous research has shown that DCNNs tend to learn the most discriminative patterns of each class while ignoring detailed visual cues [65]. We design an auxiliary loss to stimulate fine-detailed patterns to emerge to improve the learned filter banks further. We assign a different instance index g_i to each training image and follow Eq. (3) to compute the classification logits for each instance X_i where i indicates *instances*. Then we follow Eq. (4) to minimize the instance-aware loss $L_i(X_i, g_i)$ to guide the model to learn rich filter banks for fine-detailed representations, thus improving the model's generalization ability for the Few-Shot Class-Incremental Learning. The overall model training is formulated as

$$\arg\min_{\theta_f, \theta_c} L_c(X_c, g_c) + \gamma L_i(X_i, g_i), \tag{5}$$

where θ_f is the parameters of the FBN feature extractor, θ_c is the learnable class prototypes, and γ is a balance factor which we use 0.1 in all experiments.

[1] We use $E = 16$ in all FSCIL experiments.

3.3 Discussion

Design choices in machine learning signify inductive biases. For example, the composition of layers in Deep Convolution Neural Networks (DCNNs) provides a type of relational inductive bias, i.e., hierarchical processing. And the use of convolution in DCNNs introduces the inductive bias of spatial translation. More generally, anything that imposes constraints on the learning trajectory introduces inductive biases, e.g., dropout, data augmentation, batch normalization, and weight decay. Previous studies have shown that implementing inductive biases to deep learning architectures can facilitate model learning about entities, relations, and rules for composing them and thus improve performance and generalization [66].

The conventional DCNNs leverage inductive biases introduced in convolution, hierarchical structure, and local pooling to handle moderate transforms, i.e., spatial transitions, mild scale changes, and small rotations. However, DCNNs lack the ability to handle significant and generic transforms; thus, the most straightforward way to decrease loss is using the abundant filters to memorize the seen patterns by rote. Transformed variants are often redundantly learned in low-level, middle-level, and relatively high-level filters, Fig. 2. Consequently, the models tend to overfit the existing data and cannot generalize to unseen variants, especially when the data is insufficient.

The proposed Filter Bank Networks approach provides a unified framework for implementing inductive biases within the Deep Convolution Networks (DCNNs) while retaining end-to-end learning capability. Fig. 4 shows the top layer convolutional features of a LeNet-like FBN model trained on the MNIST dataset. It can be seen that the FBN features are intra-class equivariant while maintaining inter-class discrimination. Specifically, the feature difference between the upright 9 and the upside-down 6 is significant despite being visually similar. And the features of the upright 9 and the upside-down 9 are equivariant. FBNs endow DCNNs with equivariant feature representations for generic transformations, improving the essential capability for FSCIL, namely generalization.

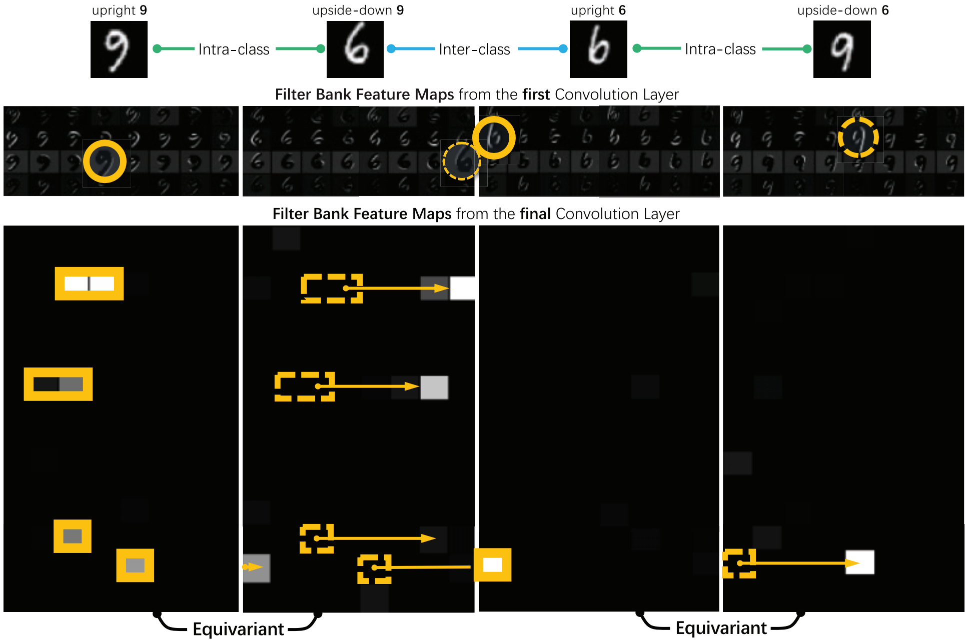
upright 9
upside-down 9
upright 6
upside-down 6
Intra-class
Inter-class
Intra-class
Filter Bank Feature Maps from the first Convolution Layer
Filter Bank Feature Maps from the final Convolution Layer
Equivariant
Equivariant

Figure 4: Feature maps produced by filter banks learned from MNIST dataset. FBNs endow DCNNs with equivariant features for transformations. Best viewed by zooming on the screen

4 Experiments

The proposed Filter Bank Networks (FBNs) are extensively evaluated. In Section 4.1, experiments on the transformed MNIST dataset [67] are conducted, showing that filter banks can learn transformed variants of intermediate representation and significantly improve the generalization ability of Deep Convolution Neural Networks (DCNNs). In Section 4.2, FBNs are tested on three commonly used benchmarks of Few-Shot Class-Incremental Learning (FSCIL), i.e., CIFAR100, Mini-ImageNet, and CUB200, demonstrating the state-of-the-art performance of FBNs. In Section 4.3, ablation studies are performed to validate the effectiveness of designs in FBNs.

4.1 Cross Generalization

Dataset Description. We transform test samples in the MNIST dataset [67] via bilinear sampling to develop the *scaling*, *flip*, and *rotation* versions. Samples of *scaling* are resized by a random factor between $[0.2, 1.0]$. Samples of *flip* are randomly flipped horizontally or vertically with a probability of 0.5. Samples of *rotation* are randomly rotated by a angle between $[0, 2\pi]$. We train the models with the original upright MNIST training samples and test with transformed versions of test samples to measure the cross-generalization ability (upright → transformed) of the conventional DCNN baseline and the upgraded FBN.

Implementation Details. We set up a baseline DCNN with four convolution layers with a kernel size of 3×3. Each convolution layer is followed by a ReLU activation layer. A 2×2 Max Pooling operator is used after the first three convolution layers and a global average pooling operator is used after the final convolution layer to obtain the feature representation. The feature is classified by a multi-layer perceptron including two linear layers. For different versions of transformed MNIST, we upgrade the baseline DCNN to FBN with the filter banks defined by a transformation matrix set $\mathcal{T}$. For instance, we use $\mathcal{T}_s = \left\{ \begin{pmatrix} 2^i & 0 & 0 \\ 0 & 2^i & 0 \\ 0 & 0 & 1 \end{pmatrix}, i \in [0, 3] \right\}$, $\mathcal{T}_f = \left\{ \begin{pmatrix} i & 0 & 0 \\ 0 & j & 0 \\ 0 & 0 & 1 \end{pmatrix}, i, j \in \{-1, 1\} \right\}$, and $\mathcal{T}_r = \left\{ \begin{pmatrix} cos(i * \frac{\pi}{4}) & -sin(i * \frac{\pi}{4}) & 0 \\ sin(i * \frac{\pi}{4}) & cos(i * \frac{\pi}{4}) & 0 \\ 0 & 0 & 1 \end{pmatrix}, i \in [0, 7] \right\}$ for the *scaling*, *flip*, and *rotation* versions of MNIST, repectively. Note we also divide the number of channels in each convolution layer by the number of transforms in each $\mathcal{T}$ to match FBN's feature dimension with the baseline. The Max Pooling across response channels of each filter bank is used in the final convolution layer to encode the invariant FBN representation. Both the DCNN baseline and the FBN models are trained with standard Cross-Entropy loss, batch size of 128, and Adadelta optimization for 100 epochs.

Performance. It can be seen in Table 1 that FBN improves the accuracy in *scaling* and *flip* setting by 3.17% (70.17% *vs*. 67.00%) and 20.7% (91.00% *vs*. 70.30%) respectively with 25.27% learnable parameters. FBN significantly outperform the baseline in the *rotation* setting by 34.37% (78.97% *vs*. 44.60%) while using only 12.82% parameters. The third column of Table 1 shows that conventional DCNNs can be generalized to samples of different scales to a certain extent, thanks to the hierarchical processing and pooling layers. The huge performance gaps (34.37% and 20.70%) in the fifth and seventh columns of Table 1 indicate that the conventional DCNN can not handle flipped and rotated samples due to the lack of corresponding model-level prior knowledge. Based on this observation and the fact that samples are often flipped in the data augmentation, we use $\mathcal{T} = \mathcal{T}_r$ in all Few-Shot Class-Incremental Learning experiments.

Table 1: Results on the MNIST-generalization

Method	original → *scaling*		original → *rotation*		original → *flip*	
	Params (%)	Accuracy (%)	Params (%)	Accuracy (%)	Params (%)	Accuracy (%)
Baseline DCNN	100.00	67.00	100.00	44.60	100.00	70.30
FBN (ours)	**25.27**	**70.17**	**12.82**	**78.97**	**25.27**	**91.00**

Fig. 5 shows the t-SNE 2D mapping of the features captured by DCNN and FBN where each dot corresponds to a test sample and different colors for different ground truth classes. It can be seen that the FBNs' features constitute clear clusters and produce a much more explicit feature distribution in manifold than the baseline DCNN.

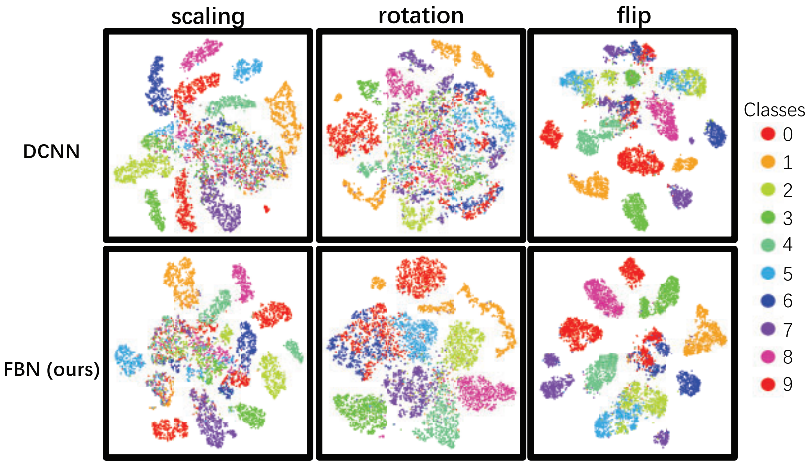

Figure 5: t-SNE visualization of the features produced by DCNN and FBN on the MNIST-generalization dataset. Best viewed in color

The Filter Bank Networks consistently outperform the baselines while using significantly fewer learnable parameters, demonstrating the strong generalization ability of filter banks to the unseen variants of semantic patterns.

4.2 Few-Shot Class-Incremental Learning

We evaluate the proposed FBN on the commonly used FSCIL benchmarks including CIFAR100, CUB200, and Mini-ImageNet. The categories in the datasets are divided into base classes with adequate annotations and new ones with K-shot annotated images. For FSCIL, the network is trained upon base classes for the first pretraining session. New classes are gradually added to train FBN in T incremental sessions. In each incremental session, N-way new classes are added.

Implementation Details. The proposed FBN is built upon the ResNet18/ResNet20 network and optimized with the standard SGD algorithm. We follow the state-of-the-art methods to use four data augmentation strategies, i.e., normalization, horizontal flipping, random cropping, and random resizing. During the first session, FBN is trained using $D^{(0)}$ upon the base classes. When $t > 0$, the model is trained upon $D^{(t)}$ with new classes. After the T-th session training, FBN is evaluated on all of the seen classes to obtain the average classification accuracy. We use the official implementation of [15] for the training loop to keep the hyperparameters consistent with baselines. All the training images ($N \times K$) are fed to the network through a batch. Since the performance could be sensitive to the orders of class identities and images, we conducted experiments 10 times with different random seeds and reported the average results.

4.2.1 CIFAR100

Dataset Description. CIFAR100 consists of 100 classes, where 60 classes are used as base classes in the pretraining session and 40 as new classes. Each new class has 5-shot annotated images ($K = 5$). The new classes are divided into 8 sessions ($T = 8$), each of which has 5 classes ($N = 5$). In this dataset, the image size is 32 × 32, and the detailed information on categories is in the left part of Table 2. We use the same training hyperparameters as FACT [15], i.e., learning rate 0.1 for the base session training, and a batch size of 256.

Table 2: Categories in CIFAR100 and Mini-ImageNet dataset. The first column denotes the session index where 0 indicates the base classes for the pretraining session. Best viewed by zooming on the screen

	CIFAR100						Mini-ImageNet			
	mammals beaver	dolphin	otter	seal	whale	tench	goldfish	great-white-shark	tiger- shark	hammerhead
	aquarum fish	flatfish	ray	shark	trout	electric- ray	stingray	coc	k-hen	ostrich
	orchids	poppies	roses	sunflowers	tulips	brambling	goldfinch	house- finch	junco	indigo-bunting
	containers bottles	bowls	cans	cups	plates	robin	bulbul	jay	magpie	chickadee
	apples	mushrooms	oranges	pears	sweet peppers	water- ouzel	kite	bald- eagle	vulture	great-grey-owl
	clock	computer keyboard	lamp	telephone	television	European -fire- salamander	common- newt	eft	spotted-salamander	axolotl bullfrog
0	furniture bed	chair	couch	table	wardrobe	tree- frog	tailed- frog	loggerhead	leatherback-turtle	mud-turtl
	bee	beetle	butterfly	caterpillar	cockroach	terrapin	box-turtle	banded- gecko	common-iguana	American-chameleon
	bear	leopard	lion	tiger	wolf	whiptail	agama	frilled- lizard	alligator-lizard	Gila-monster
	bridge	castle	house	road	skyscraper	green- lizard	African-chameleon	Komodo-dragon	African-crocodile	American-alligator
	cloud	forest	mountain	plain	sea	triceratops	thunder- snake	ringneck- snake	hognose-snake	green- snake
	camel	cattle	chimpanzee	elephant	kangaroo	king-snake	garter- snake	water- snake	vine- snake	night- snake
1	fox	porcupine	possum	raccoon	skunk	boa-constrictor	rock- python	Indian- cobra	green-mamba	sea- snake
2	crab	lobster	snail	spider	worm	horned-viper	diamondback	sidewinder	trilobite	harvestman
3	baby	boy	girl	man	woman	scorpion	black-and- gold-garden-spider	barn-spider	garden-spider	black-widow

(Continued)

Table 2 (continued)

	CIFAR100						Mini-ImageNet			
4	crocodile	dinosaur	lizard	snake	turtle	tarantula	wolf-spider	tick	centipede	black-grouse
5	hamster	mouse	rabbit	shrew	squirrel	ptarmigan	ruffed- grouse	prairie- chicken	peacock	quail
6	maple	oak	palm	pine	willow	partridge	African-grey	macaw	sulphur-crested-cockatoo	lorikeet
7	bicycle	bus	motor cycle	pickup truck	train	coucal	bee-eater	hornbill	humming bird	jacamar
8	lawn-mower	rocket	streetcar	tank	tractor	toucan	drake	red-breasted-merganser	goose	black-swan

Performance. Table 3 shows the comparison of the proposed FBN and the state-of-the-art methods. We calculate the average classification accuracy of each session's seen classes. It is shown in Table 3 that FBNs achieve the new state-of-the-art performance. Specifically, FBNs outperform TOPIC by 29.55%, which is a large margin. Compared with recent works, it outperforms CEC and FACT by 9.78% and 7.31%, respectively. The significant performance improvement demonstrates the effectiveness of the proposed FBN. Fig. 6 shows that FBN achieves the highest accuracy across all the sessions, validating that the proposed FBN maintains high performance across the whole incremental learning and alleviates models' forgetting issue.

Table 3: Classification accuracy comparison on CIFAR100 using the ResNet20 backbone

Method	Session									Average
	0	1	2	3	4	5	6	7	8	
Ft-CNN	64.10	36.91	15.37	9.80	6.67	3.80	3.70	3.14	2.65	16.24
iCaRL [48]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	32.88
EEIL [68]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	33.79
NCM [69]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	34.22
TOPIC [13]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62
CEC [54]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53
FACT [15]	79.25	72.57	68.37	63.95	60.75	57.91	55.53	53.58	51.61	62.61
FBN (ours)	**84.03**	**78.79**	**75.16**	**71.20**	**68.14**	**65.11**	**63.31**	**61.57**	**58.92**	**69.58**

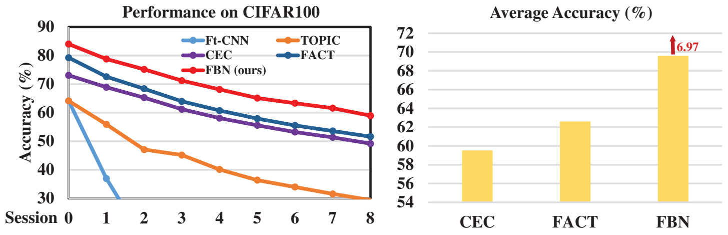

Figure 6: Performance of all the sessions and comparison of average accuracy on CIFAR100

Table 4: Classification accuracy comparison on Mini-ImageNet using the ResNet18 backbone

Method	Session									Average
	0	1	2	3	4	5	6	7	8	
TOPIC [13]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64
CEC [54]	72.00	66.86	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75
FACT [15]	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.69
FBN (ours)	**76.02**	**71.92**	**67.67**	**64.47**	**61.10**	**58.38**	**55.54**	**53.98**	**52.61**	**62.41**

4.2.2 Mini-ImageNet

Dataset Description. Mini-ImageNet is a subset of the ImageNet dataset. It is composed of 100 categories sampled from ImageNet, where 60 classes are set as base classes and 40 as new classes. Each new class has 5-shot annotated images ($K = 5$). The new classes are divided into 8 sessions ($T = 8$), each of which has 5 classes ($N = 5$). In this dataset, the image size is 64 × 64, and the detailed information on categories is in the right part of Table 2.

Performance. To validate the superiority of our proposed FBN, we compare the performance of all the sessions with state-of-the-art methods. From Table 4 we can see that FBN outperforms CEC [54] and FACT [15] by 4.98% and 2.12% in the last session, respectively. To clearly show the performance drop across all the sessions and comparison of the averaged accuracies, we drew the line chart and the plot chart, Fig. 7. As shown in Fig. 7, the proposed FBN outperforms all the existing methods across all the sessions and achieves the highest averaged accuracy, validating the superiority of our method in the dataset with strong class diversity.

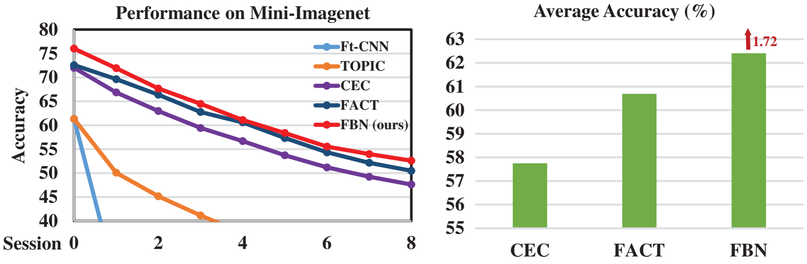

Figure 7: Performance of all the sessions and comparison of average accuracy on Mini-ImageNet

4.2.3 CUB200

Dataset Description. CUB200 consists of 200 classes where 100 classes are set as base classes and the other 100 classes as new classes under the settings of $K = 5$, $T = 10$, $N = 10$. All the categories in this dataset are birds. Thus it requires the model to learn fine-grained features between classes. The image size of CUB200 is 224 × 224.

Performance. To validate the models' ability to mine fine-detailed features, we do not use ImageNet-pretraining in the experiments. For a fair comparison, we re-evaluate CEC [54] and FACT [15] with a consistent setting based on their official implementations, and the result is reported in Table 5. One can see that without pretraining, FBN outperforms CEC [54] and FACT [15] by 13.94%

and 1.45% in the last session, respectively. We analyze the classes with the highest performance improvements for each session and visualize the samples of those classes, Fig. 8. It can be seen that the 'Fish Crow', 'Pigeon Guillemot', and 'Ovenbird' achieve the highest performance improvements on the base session. Typically, those samples have large intra-class differences and require more fine-detailed information for recognition. The proposed FBN is endowed with the ability to learn more diversified fine-detailed patterns with filter variants in filter banks and instance-aware loss, which benefits the learning and prediction of those classes.

Table 5: Classification accuracy comparison on CUB200 using the ResNet18 backbone

Method	Session											Average
	0	1	2	3	4	5	6	7	8	9	10	
CEC [54]	42.43	39.70	36.26	33.74	32.37	30.37	28.44	27.38	26.42	25.51	24.42	31.55
FACT [15]	64.56	59.31	54.62	50.13	49.07	45.22	42.81	40.46	39.80	38.25	36.91	47.38
FBN (ours)	**65.94**	**60.53**	**55.00**	51.18	**49.54**	**45.90**	**43.72**	**42.28**	**41.09**	**39.79**	**38.36**	**48.49**

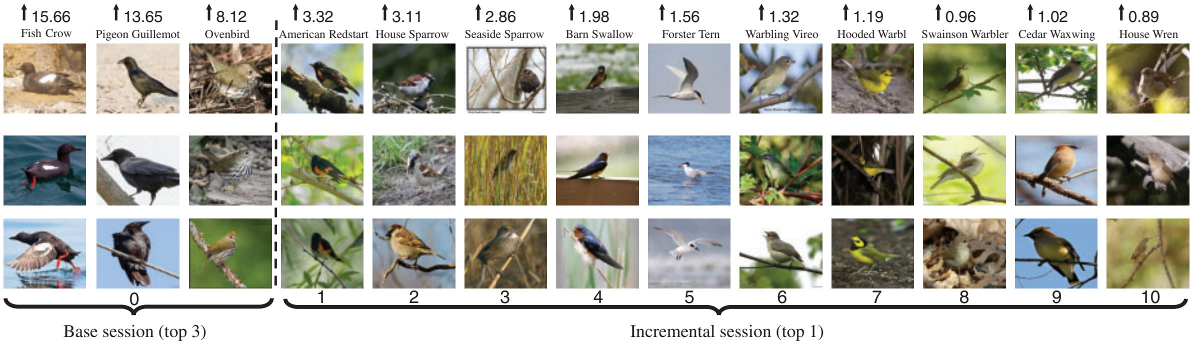

Figure 8: Samples of classes with the highest performance improvements for each session

4.2.4 Fishshot1K

Dataset Description. To further evaluate Filter Bank Networks, we propose a new challenging Few-Shot Class-Incremental Learning (FSCIL) benchmark, i.e., Fishshot1K, which contains 8261 images covering 1000 ocean fish species. The dataset is built upon an open underwater photography database[2]. We preprocess and clean the raw data in three steps. First, we sort the fish categories by the number of samples and select the top 1000 classes. Then, we choose 600 classes as the base classes used in the pretraining session and divide the rest 400 classes into 8 incremental sessions, each containing 50 novel classes per session. For each novel class, we use only 1 sample to train the model whereas the rest samples are used for evaluation, i.e., 1-shot learning. Lastly, we resize each raw image to 92 × 92 and center crop the image of the size 84 × 84. It can be seen in Fig. 9 that Fishshot1K is challenging. The fish can have arbitrary poses relative to the camera, the background is varying, and the light for underwater photography is typically weak.

[2] http://www.fishdb.co.uk/

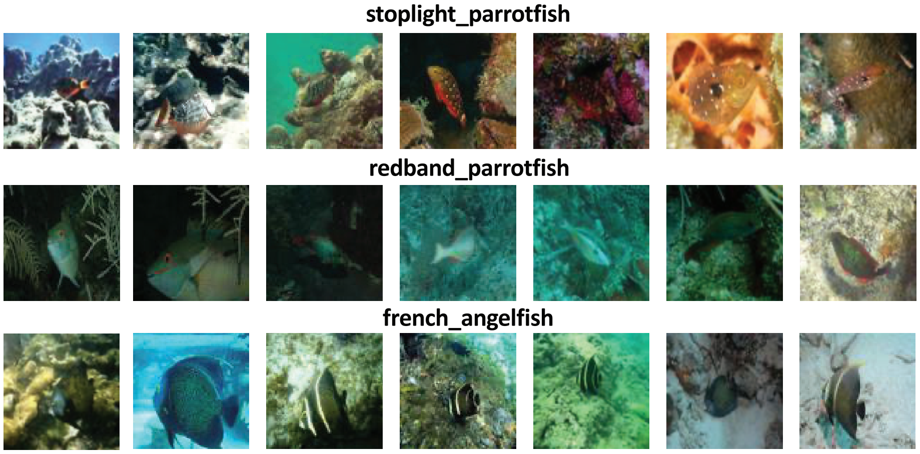

Figure 9: Samples of the proposed Fishshot1K dataset which contains 1000 ocean fish species. Images in each row belong to the same category showing significant intra-class differences

Performance. We evaluate our proposed Filter Bank Network approach on the challenging Fishshot1K benchmark and compare it with the state-of-the-art method. As shown in Table 6, FBN achieves better classification accuracy for the pretraining and all incremental sessions. The average accuracy significantly outperforms FACT [15] by 2.81% (22.01% *vs.* 19.20%). This further shows the effectiveness of FBN in learning from limited and incremental data and thus FBN has great potential in handling real-world longtail problems, e.g., research of ocean fish species recognition.

Table 6: Classification accuracy comparison on Fishshot1K

Method	Session									Average
	0	1	2	3	4	5	6	7	8	
FACT [15]	21.87	20.34	19.06	19.00	19.09	19.02	18.39	17.93	18.13	19.20
FBN (ours)	**24.96**	**23.48**	**21.89**	**21.83**	**21.83**	**21.91**	**20.63**	**20.72**	**20.83**	**22.01**

4.3 Ablation Studies

We perform ablation studies on the CIFAR100 FSCIL benchmark to evaluate the effectiveness of the two core designs in FBNs, i.e., the filter banks that learn unseen transformed variants of semantic patterns, and the instance-aware loss which stimulates fine-detailed features to emerge. We sample a subset of $\mathcal{T}_r$ ($|\mathcal{T}_r| = 8$), to form three settings. As shown in the left part of Table 7, increasing the number of transforms, i.e., $|\mathcal{T}|$, consistently improves the average accuracy over all FSCIL sessions, showing the effectiveness of filter banks. The ablation studies on instance-aware loss are evaluated in the right part of Table 7. We change the balance factor γ to different values where $\gamma = 0.0$ indicates the instance-aware loss is discarded. The performance drop ("PD") is defined as $PD = ACC^{(0)} - ACC^{(T)}$, where $ACC^{(0)}$ denotes the accuracy of pertaining session and $ACC^{(T)}$ denotes the accuracy of the T-*th* session. It can be seen that the proper use of instance-aware loss can guide the FBN to mine more fine-detailed features and alleviate the catastrophic forgetting issue. However, when the balance factor is too large, the model focuses on distinguishing instances and neglects to capture the discriminative features between categories, which affects the final performance. We use $\gamma = 0.1$ for all FSCIL experiments.

Table 7: Ablation studies of the designs choices of FBNs on CIFAR100

Filter banks			Average
$\|\mathcal{T}\| = 1$	$\|\mathcal{T}\| = 4$	$\|\mathcal{T}\| = 8$	
✓			65.02
	✓		66.27
		✓	**69.58**

Instance-aware loss					PD
$\gamma = 0.0$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.5$	$\gamma = 1.0$	
✓					26.59
	✓				**25.11**
		✓			25.48
			✓		25.61
				✓	26.24

5 Conclusion

In this paper, we proposed a simple yet effective Few-Shot Class-Incremental Learning (FSCIL) framework, i.e., Filter Bank Networks (FBNs), which introduce learnable inductive biases to address the issue of overfitting and catastrophic forgetting. FBNs improve modern Deep Convolution Neural Networks (DCNNs) to achieve the capability of learning variants of visual patterns that do not exist in the dataset. FBNs augment each learnable filter to a virtual filter bank, containing its canonical form and multiple transformed versions. During back-propagation, gradients of the entire filter bank are collectively aligned and aggregated to update the canonical filter. Moreover, FBNs stimulate instance-aware discriminative patterns to emerge and learn diverse features, reserving latent embedding space for incorporating future novel classes in incremental learning sessions. The primary contributions are three-fold. First, we design the learning paradigm of Filter Bank Networks, including learning augmented filters and mining instance-aware features. Second, we upgrade the modern architecture of DCNNs to FBN and achieve new state-of-the-art results in commonly used Few-Shot Class-Incremental Learning (FSCIL) benchmarks, including CIFAR100, Mini-ImageNet, and CUB200. Last but not least, we contribute a challenging FSCIL benchmark, namely Fishshot1K, which contains 8261 underwater images covering 1000 ocean fish species. With the same training hyperparameters, FBNs consistently outperform their baselines and report the best results, which indicates that the usage of learnable inductive biases is a crucial factor in training growable models which can incrementally learn from limited data.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: The authors are very grateful for support from the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA27000000.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations,* San Diego, CA, USA.
2. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence,* San Francisco, California, USA, AAAI Press.
3. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, NV, USA, IEEE Computer Society.
4. Deng, J., Dong, W., Socher, R., Li, L., Li, K. et al. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009),* Miami, Florida, USA, IEEE Computer Society.
5. Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P. et al. (2014). Microsoft COCO: Common objects in context. *Computer Vision-ECCV 2014*, pp. 740–755. Zurich, Switzerland, Springer.
6. Yoon, J., Yang, E., Lee, J., Hwang, S. J. (2018). Lifelong learning with dynamically expandable networks. *6th International Conference on Learning Representations*, Vancouver, BC, Canada.
7. Xiang, Y., Fu, Y., Ji, P., Huang, H. (2019). Incremental learning using conditional adversarial networks. *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), IEEE.
8. Kim, C. D., Jeong, J., Moon, S., Kim, G. (2021). Continual learning on noisy data streams via self-purified replay. *2021 IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, IEEE.
9. Smith, J., Hsu, Y., Balloch, J., Shen, Y., Jin, H. et al. (2021). Always be dreaming: A new approach for data-free class-incremental learning. *2021 IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, IEEE.
10. Hou, S., Pan, X., Loy, C. C., Wang, Z., Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.
11. Hu, X., Tang, K., Miao, C., Hua, X., Zhang, H. (2021). Distilling causal effect of data in class-incremental learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Computer Vision Foundation/IEEE.
12. Cha, H., Lee, J., Shin, J. (2021). Co^{2}L: Contrastive continual learning. *2021 IEEE/CVF International Conference on Computer Vision*, Montreal, QC, Canada, IEEE.
13. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X. et al. (2020). Few-shot class-incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Computer Vision Foundation/IEEE.
14. Cheraghian, A., Rahman, S., Fang, P., Roy, S. K., Petersson, L. et al. (2021). Semantic-aware knowledge distillation for few-shot class-incremental learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Virtual, Computer Vision Foundation/IEEE.
15. Zhou, D., Wang, F., Ye, H., Ma, L., Pu, S. et al. (2022). Forward compatible few-shot class-incremental learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, IEEE.
16. Yang, S., Liu, L., Xu, M. (2021). Free lunch for few-shot learning: Distribution calibration. *9th International Conference on Learning Representations*, Austria.
17. Li, Z., Hoiem, D. (2018). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12),* 2935–2947. https://doi.org/10.1109/TPAMI.2017.2773081

18. Dhar, P., Singh, R. V., Peng, K., Wu, Z., Chellappa, R. (2019). Learning without memorizing. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.
19. Zenke, F., Poole, B., Ganguli, S. (2017). Continual learning through synaptic intelligence. *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. Sydney, NSW, Australia.
20. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *IEEE International Conference on Computer Vision*, Venice, Italy, IEEE Computer Society.
21. Chaudhry, A., Dokania, P. K., Ajanthan, T., Torr, P. H. S. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547. Munich, Germany, Springer.
22. Krizhevsky, A., Sutskever, I., Hinton, G. E., Bartlett, P. L., Pereira, F. C. N. et al. (2012). Imagenet classification with deep convolutional neural networks. *26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, USA.
23. Girshick, R. B. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision*, Santiago, Chile, IEEE Computer Society.
24. Ren, S., He, K., Girshick, R., Sun, J., Cortes, C. et al. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc.
25. Ronneberger, O., Fischer, P., Brox, T., Navab, N., Hornegger, J. et al. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015*, pp. 234–241. Germany, Springer.
26. He, K., Gkioxari, G., Dollár, P., Girshick, R. B. (2017). Mask R-CNN. *IEEE International Conference on Computer Vision*, Venice, Italy, IEEE Computer Society.
27. Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, Kerkyra, Corfu, Greece, IEEE Computer Society.
28. Ojala, T., Pietikäinen, M., Mäenpää, T. (2000). Gray scale and rotation invariant texture classification with local binary patterns. *Computer Vision-ECCV 2000, 6th European Conference on Computer Vision*, vol. 1842, Dublin, Ireland, Springer.
29. Ahonen, T., Hadid, A., Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12),* 2037–2041. https://doi.org/10.1109/TPAMI.2006.244
30. Haley, G. M., Manjunath, B. S. (1995). Rotation-invariant texture classification using modified gabor filters. *Proceedings 1995 International Conference on Image Processing*, Washington DC, USA, IEEE Computer Society.
31. Han, J., Ma, K. (2007). Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image and Vision Computing, 25(9),* 1474–1481. https://doi.org/10.1016/j.imavis.2006.12.015
32. Skibbe, H., Reisert, M. (2012). Circular fourier-hog features for rotation invariant object detection in biomedical images. *9th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Spain, IEEE.
33. Liu, K., Skibbe, H., Schmidt, T., Blein, T., Palme, K. et al. (2014). Rotation-invariant HOG descriptors using fourier analysis in polar and spherical coordinates. *International Journal of Computer Vision, 106(3),* 342–364. https://doi.org/10.1007/s11263-013-0634-z
34. Scherer, D., Müller, A. C., Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *Artificial Neural Networks-ICANN 2010*, pp. 92–101. Thessaloniki, Greece, Springer.
35. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D. et al. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems 29*, Barcelona, Spain.

36. Snell, J., Swersky, K., Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems 30*.
37. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S. et al. (2018). Learning to compare: Relation network for few-shot learning. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, Computer Vision Foundation/IEEE Computer Society.
38. Zhang, C., Cai, Y., Lin, G., Shen, C. (2020). Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Computer Vision Foundation/IEEE.
39. Liu, B., Ding, Y., Jiao, J., Ji, X., Ye, Q. (2021). Anti-aliasing semantic reconstruction for few-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9747–9756.
40. Yang, B., Liu, C., Li, B., Jiao, J., Ye, Q. (2020). Prototype mixture models for few-shot semantic segmentation. *Computer Vision-ECCV 2020*, pp. 763–778. Glasgow, UK, Springer.
41. Liu, B., Jiao, J., Ye, Q. (2021). Harmonic feature activation for few-shot semantic segmentation. *IEEE Transactions on Image Processing, 30,* 3142–3153. https://doi.org/10.1109/TIP.2021.3058512
42. Finn, C., Abbeel, P., Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. Sydney, NSW, Australia, PMLR.
43. Elsken, T., Staffler, B., Metzen, J. H., Hutter, F. (2020). Meta-learning of neural architectures for few-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Computer Vision Foundation/IEEE.
44. Sun, Q., Liu, Y., Chua, T., Schiele, B. (2019). Meta-transfer learning for few-shot learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.
45. Zhang, H., Zhang, J., Koniusz, P. (2019). Few-shot learning via saliency-guided hallucination of samples. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.
46. Li, K., Zhang, Y., Li, K., Fu, Y. (2020). Adversarial feature hallucination networks for few-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Computer Vision Foundation/IEEE.
47. Kim, J., Kim, H., Kim, G. (2020). Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. *Computer Vision-ECCV 2020-16th European Conference*, vol. 12346. Glasgow, UK, Springer.
48. Rebuffi, S., Kolesnikov, A., Sperl, G., Lampert, C. H. (2017). ICARL: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, IEEE Computer Society.
49. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z. et al. (2019). Large scale incremental learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.
50. Shin, H., Lee, J. K., Kim, J., Kim, J. (2017). Continual learning with deep generative replay. *Advances in Neural Information Processing Systems 30*.
51. Serrà, J., Suris, D., Miron, M., Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. *Proceedings of the 35th International Conference on Machine Learning*, vol. 80. Stockholmsmässan, Stockholm, Sweden.
52. Mallya, A., Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, Computer Vision Foundation/IEEE Computer Society.
53. Zhu, K., Cao, Y., Zhai, W., Cheng, J., Zha, Z. (2021). Self-promoted prototype refinement for few-shot class-incremental learning. *IEEE Conference on Computer Vision and Pattern Recognition*, Virtual, Computer Vision Foundation/IEEE.

54. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P. et al. (2021). Few-shot incremental learning with continually evolved classifiers. *IEEE Conference on Computer Vision and Pattern Recognition*, Virtual, Computer Vision Foundation/IEEE.
55. Cheraghian, A., Rahman, S., Ramasinghe, S., Fang, P., Simon, C. et al. (2021). Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. *2021 IEEE/CVF International Conference on Computer Vision*, QC, Canada, IEEE.
56. Yu, F., Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations*.
57. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J. (2017). Oriented response networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, HI, USA, IEEE Computer Society.
58. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G. et al. (2017). Deformable convolutional networks. *IEEE International Conference on Computer Vision*, Venice, Italy, IEEE Computer Society.
59. Hinton, G. E., Krizhevsky, A., Wang, S. D., Honkela, T., Duch, W. et al. (2011). Transforming auto-encoders. *Artificial Neural Networks and Machine Learning-ICANN 2011*, pp. 44–45. Espoo, Finland, Springer.
60. Kivinen, J. J., Williams, C. K. I., Honkela, T., Duch, W., Girolami, M. A. et al. (2011). Transformation equivariant boltzmann machines. *Artificial Neural Networks and Machine Learning-ICANN 2011*, pp. 1–9. Espoo, Finland, Springer.
61. Sohn, K., Lee, H. (2012). Learning invariant representations with local transformations. *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, icml.cc/Omnipress.
62. Skibbe, H. (2013). *Spherical tensor algebra for biomedical image analysis = Sphärische Tensor Algebra für die Biomedizinische Bildanalyse (Ph.D. Thesis)*. University of Freiburg, Germany.
63. Cohen, T., Welling, M. (2016). Group equivariant convolutional networks. *Proceedings of the 33nd International Conference on Machine Learning*, vol. 48. New York, NY, USA.
64. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E. et al. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, IEEE Computer Society.
65. Ding, Y., Zhou, Y., Zhu, Y., Ye, Q., Jiao, J. (2019). Selective sparse sampling for fine-grained image recognition. *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), IEEE.
66. Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F. et al. (2018). Relational inductive biases, deep learning, and graph networks.
67. Liu, C., Nakashima, K., Sako, H., Fujisawa, H. (2003). Handwritten digit recognition: Benchmarking of state-of-the-art techniques. *Pattern Recognition, 36(10),* 2271–2285. https://doi.org/10.1016/S0031-3203(03)00085-2
68. Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., Alahari, K. (2018). End-to-end incremental learning. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 233–248. Munich, Germany, Springer.
69. Hou, S., Pan, X., Loy, C. C., Wang, Z., Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Computer Vision Foundation/IEEE.