



# Match them up: visually explainable few-shot image classification

Bowen Wang<sup>1</sup> · Liangzhi Li<sup>1</sup> · Manisha Verma<sup>1</sup> · Yuta Nakashima<sup>1</sup> · Ryo Kawasaki<sup>2</sup> · Hajime Nagahara<sup>1</sup>

Accepted: 8 August 2022 / Published online: 27 August 2022  
© The Author(s) 2022

## Abstract

Few-shot learning (FSL) approaches, mostly neural network-based, assume that pre-trained knowledge can be obtained from base (seen) classes and transferred to novel (unseen) classes. However, the black-box nature of neural networks makes it difficult to understand what is actually transferred, which may hamper FSL application in some risk-sensitive areas. In this paper, we reveal a new way to perform FSL for image classification, using a visual representation from the backbone model and patterns generated by a self-attention based explainable module. The representation weighted by patterns only includes a minimum number of distinguishable features and the visualized patterns can serve as an informative hint on the transferred knowledge. On three mainstream datasets, experimental results prove that the proposed method can enable satisfying explainability and achieve high classification results. Code is available at <https://github.com/wbw520/MTUNet>.

**Keywords** Deep learning · Explainable AI · Few shot learning · Learning representation · Attention

## 1 Introduction

Few-shot learning (FSL) is of great significance for at least the following two scenarios [1]: First, FSL can relieve the heavy needs for data gathering and labelling, which can boost the ubiquitous use of deep learning techniques, especially for users without enough resources. Second, FSL is an important solution for applications in which rare

cases matter or image acquisition is costly because of high operation difficulty or ethical issues. Typical examples of such applications include computer-assisted diagnosis with medical imaging, and classification of endangered species.

An FSL task is typically formulated as follows: Given *support* images with corresponding labels and a query image without any label, it requires to finding the label of the query image based on the labels of support images. With this formulation, most FSL methods train the model on base (seen) classes and evaluate the model on novel (unseen) classes. It is assumed that knowledge can be well extracted from base classes and transferred to novel classes. However, this is not always the case. The knowledge in a pre-trained backbone convolutional neural networks (CNNs), which computes the features of an input image, may sometimes be useless when novel classes have significant visual differences from base class images [2]. For example, having sheep always on grass and cats mostly in indoor environments, FSL models may classify an image showing a cat on grass as the class of “sheep” because “cat” has a very large visual difference with all base classes while owning a similar background with one base class. What makes matters worse is that we even have no way to see if the visual differences between the base and novel classes are significant for an FSL model. This raised one essential question: *Is there any way to see what is transferred from base classes to novel classes?* Most research on FSL tasks do not pay attention to what is extracted from the backbone CNNs.

---

✉ Bowen Wang  
bowen.wang@ids.osaka-u.ac.jp

Liangzhi Li  
li@ids.osaka-u.ac.jp

Manisha Verma  
mverma@ids.osaka-u.ac.jp

Yuta Nakashima  
n-yuta@ids.osaka-u.ac.jp

Ryo Kawasaki  
ryo.kawasaki@ophthal.med.osaka-u.ac.jp

Hajime Nagahara  
nagahara@ids.osaka-u.ac.jp

<sup>1</sup> Institute for Dataability Science, Osaka University, 2-1, Yamadaoka, Suita, 5650871, Osaka, Japan

<sup>2</sup> Graduate School of Medicine, Osaka University, 2-1, Yamadaoka, Suita, 5650871, Osaka, Japan

In this study, we redesign the mechanism of knowledge transfer for FSL tasks, offering an answer to the above question. Our approach is inspired by what humans seemingly do when trying to recognize a rarely seen object. That is, we usually try to find some patterns in the object and match them in a small number of previously seen examples in our memory. We mimic this process by designing a self-explainable attention module, and propose a new FSL method, named a match-them-up network (MTUNet), which consists of a *pattern extractor* (PE) and *pairwise matching* (PM).

The PE is designed to find discriminative patterns for image representation. The knowledge transferred from the base classes to the novel classes is thus the learned patterns. Owing to the explainability of the PE, the extracted patterns themselves can be easily visualized by exemplifying them in the images as shown in Fig. 1(a). This directly means that we have a way to see what is transferred in our FSL pipeline. The patterns extracted from each of the support and query images are aggregated to form discriminative image representation, which is shown as overall attention in Fig. 1(b) and is used for matching. In Fig. 1(b), the visualization of aggregated patterns collectively shows a consistent and meaningful clue for the images of the same

class. For example, the PE shows strong attention on the neck of the goose in the second column, which is consistent in both support and query images (even for sub-images in the latter). Image representation based on the patterns learned from base classes makes matching between a pair of images much easier by incorporating only a small number of regions to pay attention to.

On top of the PE, PM is adopted to determine whether image pairs belong to the same class. Each pair consists of one image from the *support* set and one image from the *query* set. The category of the support image that has the highest similarity score is regarded as the query image’s category. Together with the PE, MTUNet can provide a matching score to further relate the visualization and model decision.

The main contributions of our work include:

- We propose a new explainable FSL model that achieves high classification accuracy, qualitatively and quantitatively showing its explainability.
- We design the PE module to spatially filter the original image’s features provided by a backbone CNN, keeping only informative regions of specific patterns that contribute to better FSL classification performance. Visualization of these regions plays a central role in MTUNet’s explainability as it presents the model’s basis of prediction.
- A PM mechanism that can relate the visual explanations with the model decision using matching scores, which may help find potential prediction failures.
- Our method combines several techniques and concepts, e.g., FSL, attention, feature representation, and explainable AI, which can inspire future research.

This paper is an extension of a four-page CVPR2021 workshop paper [4]. In addition to more detailed description of our method (Section 3), extensive literature review (Section 2), discussion based on our experimental results (Section 4.5), and limitations and future works (Section 4.5). The extension includes technical contributions as follows: (1) We introduce the PE pre-training, which allows better FSL classification performance. We also redesigned and detailed our methodology (e.g. in Section 3), re-did all experiments with redesigned method (Section 4.3), and additional figures (e.g. Fig. 1) are added for easier understanding. (2) We add new experimental results over another dataset, CIFAR-FS, which show superior classification performance than existing methods and validate the generalizability of our method to different datasets. (3) To compare with previous XAI methods, we design an experiment using existing XAI metrics in Section 4.4.2. The results quantitatively demonstrate the explainability of the proposed method. (4) We add a discussion based on our experimental results (Sections 4.5.1 and 4.5.2).

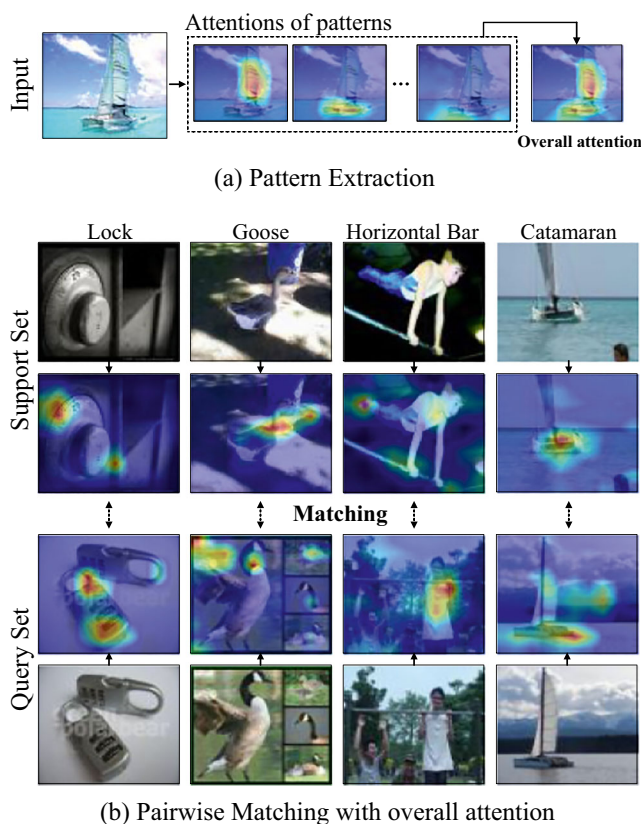


Fig. 1 Few-shot learning using pair-matching with the *pattern extractor* (PE). Images are from the mini-ImageNet dataset [3]

## 2 Related work

### 2.1 Few-shot learning

Recently, due to the availability of a sufficient number of labelled images, deep neural networks have achieved outstanding performance on various classification tasks. Such large datasets usually require a large amount of effort for their creation, and some tasks, such as medical tasks [5, 6], may not inherently have enough supervising signals. For these tasks, we require a new paradigm that allows training a model with a small number of labelled images. The popular FSL model [3, 7, 8] serve as a testbed for certain aspects of such small tasks. Recent efforts toward FSL are summarized as follows.

**Image embedding and metric learning** Many works focus on transforming images into vectors in embedding space, in which the distance between a pair of vectors represents the conceptual dissimilarity. A Siamese network [9] uses a shared feature extractor to produce image embeddings for both support and query images. The weighted  $\ell_1$  distance is used for the classification criterion. Metric learning [3, 7] can offer a better way to train the mapping into the embedding space. Some works try to improve the discriminatory power of image embeddings. Simple Shot [10] applies an  $\ell_2$  normalization and a central method to make the distance calculation easier. Instead of physical distance calculation, some works use a multi-layer perceptron (MLP) to parameterize and train similarity metrics [11–13]. A recent work [14] uses a two-stream network for better feature representation, which improves the FSL performance.

**Meta-learning** Another major approach to FSL is to optimize models so they can rapidly adapt to novel classes. The method in [15] fine-tuned the feature extractor using support images of novel classes. However, due to very few support samples, overfitting limited the model's success. MAML [16] and its extensions [17, 18] train initial parameters, and through one or more gradient adjustment steps from the initial parameters, they can be easily adapted to a target task with only a small amount of data. Besides training good initial parameters, Meta-SGD [19] trains the update direction and step size. UDS [20] adopted an unsupervised meta-learning algorithm to localize and select semantically meaningful regions in feature maps, which enables better FSL performance. A recent work [21] extends FSL into a multi-label scenario, which is meaningful to real-world applications.

**Data augmentation** Solving an FSL problem by augmenting training data is straightforward and easy to understand. Data augmentation aims at introducing immutability to models to capture information at both image and feature levels [22, 23]. There are also some works that try to use samples that

are weakly labelled or unlabelled [24, 25]. ICI [26] introduces a judgment mechanism to enhance the training set by utilizing unlabelled data with confidently predicted labels.

**Transductive or Semi-supervised Paradigm** Transductive or semi-supervised approaches [27, 28] have made great progress in the past few years. They use the statistics of query examples or statistics across FSL tasks, assuming that all novel images for classification are accessible. We only employ the original inductive paradigm to explore explainable feature extraction, but our idea can be easily adapted to a transductive paradigm.

### 2.2 Zero-shot learning

Zero-shot learning (ZSL) is another challenging task as there is no sample available for the unseen classes. An early attempt [29] proposed an attribute-based classification using human-specified high-level labels. The unseen classes can be predicted based on the combination of detected attributes, without training with the classes. Some methods were developed to utilize inter-class relationships through graph neural networks [30, 31]. Wang et al. [30] adopt a graph to use both semantic embeddings and categorical relationships to generate classifiers. OCITN [32] is designed to deal with the situation where training data with only one class. The target is to determine if the input data is seen class or unseen class. Recently, a cluster-based ZSL method [33] was proposed, which expands the idea of ZSL tasks to multivariate binary classification problem.

Our method employs a similar idea to attribute-based classification. PE is designed to learn and extract a certain set of patterns that can describe all possible classes in episodes of the FSL classification task.

### 2.3 Explainable AI

Deep neural networks are considered black-box technology, and explainable artificial intelligence (XAI) is a series of attempts to unveil them. Most XAI methods for classification tasks are based on *back-propagation* [34–36] or *perturbation* [37]. These methods are *post-hoc*, which can only provide explanations outside model training. There are also *intrinsic* methods that aim to explain the model decision spontaneously. A new type of *intrinsic* XAI, coined SCOUTER [38], has been proposed, which applies a self-attention mechanism [39] to the classifier. This method can extract the attention for each class during training, which makes classification results explainable.

XAI methods have been widely applied to many deep learning tasks [40], however, a few works [4, 41–43] have tried XAI for FSL tasks. Geng et al. [42] uses a knowledge graph to make an explanation for zero-shot tasks.

Sun et al. [41] adopt layer-wise relevance propagation (LRP) [44] to explain the output of a classifier. StarNet [43] realizes visualization through heat maps derived from back-projection. These methods are based on the idea of XAI for general classification tasks, which are not suitable for the training rule of FSL (sampling support and query [3]). Most of them are not evaluated on FSL benchmark datasets, which make these methods not comparable. Thus, an FSL model which has both high classification accuracy and interpretability is important.

In this study, we adopted the *intrinsic* approach of XAI to explore a new explainable FSL paradigm. Compared to previous FSL methods, MTUNet has PE, which is based on the self-attention mechanism [45], that can extract informative regions to improve FSL classification performance. Another difference from previous FSL is MTUNet’s explainability. Through the combination of PE and PM, MTUNet can provide insight into why the model classified a query image into a certain unseen class (refer to Section 4.4). Our experiments showed that explanation by MTUNet can help find potential prediction failures, which is important for some risk-sensitive domains like medical applications.

### 3 Material and methods

#### 3.1 Problem definition

This study addresses an inductive FSL task (*c.f.*, and a transductive task [27, 28]), in which we are given two disjointed sets  $\mathcal{D}_{base}$  and  $\mathcal{D}_{novel}$  of samples. The former is a base set of many labelled base class images whereas the latter is a novel set of a few labelled novel class images, where the disjointed sets of base and novel classes are

denoted by  $\mathcal{C}_{base}$  and  $\mathcal{C}_{novel}$ , respectively. The FSL task is to find a mapping from a novel image  $x \in \mathcal{D}_{novel}$  to the corresponding class  $y \in \mathcal{C}_{novel}$ , with the images in  $\mathcal{D}_{base}$  and the corresponding labels available in training.

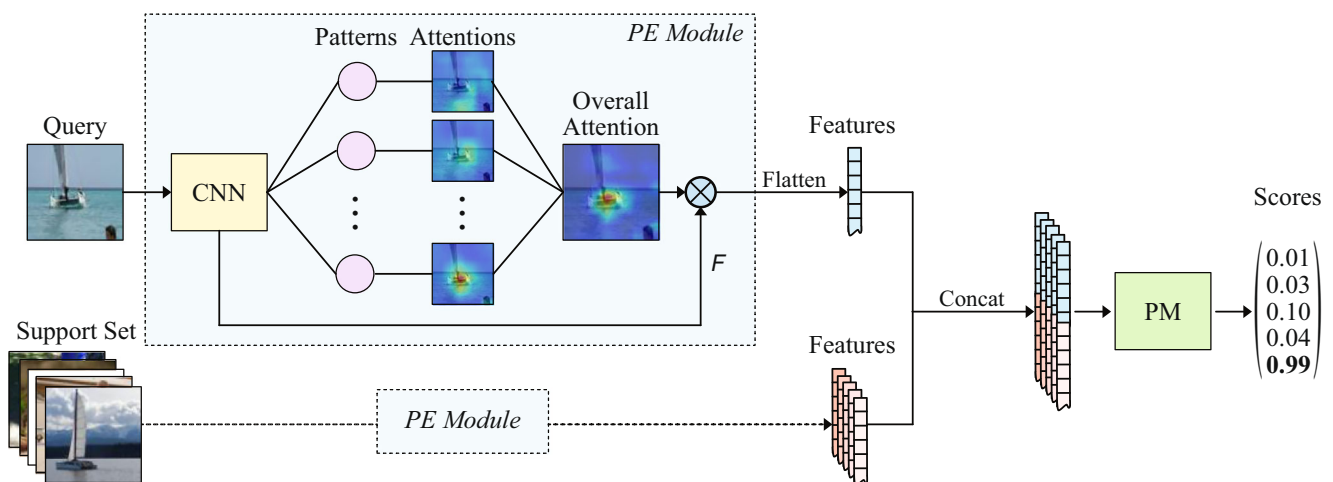
The literature typically uses the  $K$ -way  $N$ -shot episodic paradigm for training/evaluating FSL models. For each episode in training, we sample a *support set*  $\mathcal{S} = \{(x_{kn}, y_{kn}) \mid k = 1, \dots, K, n = 1, 2, \dots, N\}$  and a *query image*  $x^q$  from *query set*  $\mathcal{Q}$ . The support set contains  $N$  images for each of  $K$  classes in  $\mathcal{C}_{base}$  and serves as the basis for classification of a query image into the same  $K$  classes.

Our FSL model is trained to find a match between a query image and a support image in  $\mathcal{S}$ , i.e., the query image is classified with the class of the matched image in  $\mathcal{S}$ . Evaluation can be performed within the same paradigm by sampling query and support sets from  $\mathcal{D}_{novel}$ .

#### 3.2 Overview

The overall process is illustrated in Fig. 2. In each episode, we extract feature map  $F = f_{\theta}(x) \in \mathbb{R}^{c \times h \times w}$  from each image  $x$  in  $\mathcal{S}$  and Query image using the CNN backbone  $f_{\theta}$ , where  $\theta$  is the set of learnable parameters.  $F$  is then fed into the *pattern extractor* (PE) module,  $f_{\phi}$ , with learnable parameter set  $\phi$ . This module provides attention  $A = f_{\phi}(F) \in \mathbb{R}^{z \times l}$  over  $F$ . Our *pairwise matching* (PM) module uses an MLP to compute a score that indicates how likely query image  $x^q$  is to belong to one of the  $K$  classes in  $\mathcal{S}$ .

The PE plays a major role in the learning of FSL tasks. It is designed to learn a transferable attention mechanism, which finds common patterns that are shared among different episodes sampled from  $\mathcal{D}_{base}$ . Consequently the patterns are more likely to be shared among  $\mathcal{D}_{novel}$  given that  $\mathcal{D}_{base}$  and  $\mathcal{D}_{novel}$  are from similar domains.



**Fig. 2** Overall structure of MTUNet. One query is processed by the CNN backbone and *pattern extractor* (PE) to provide exclusive patterns and then turned into overall attention. The query is concatenated

to each support to make a pair for final discrimination through pairwise matching (PM). The dotted line represents each support image undergoing the same calculation as the query

### 3.3 Pattern extractor

Figure 3 shows the structure of our PE module. The input feature map  $F$  is first fed into a  $1 \times 1$  convolution layer followed by a ReLU nonlinearity to squeeze the dimensionality of  $F$  from  $c$  to  $d$ . The spatial dimensions of the squeezed features are flattened to form  $F' \in \mathbb{R}^{d \times l}$ , where  $l = hw$ . To maintain the spatial information, position embedding  $P$  [38, 46, 47] is added to the features, i.e.,  $\tilde{F} = F' + P$ .

The self-attention [45] mechanism provides the attention over  $F$  for the spatial dimension using the dot-product similarity between a set of  $z$  patterns and  $\tilde{F}$  after nonlinear transformations. The PE repeats this process  $T$  times by updating the patterns with a gated recurrent unit (GRU) to refine the attention. That is, let  $W^{(t)} \in \mathbb{R}^{z \times d}$  denote the patterns in the  $t$ -th repetition, where  $t = 1, 2, \dots, T$  and  $W^{(1)} = W$  is the learnable parameters. The nonlinear transformations for  $W^{(t)}$  and  $\tilde{F}$  are given by

$$g_Q(W^{(t)}) \in \mathbb{R}^{z \times d}, \quad g_K(\tilde{F}) \in \mathbb{R}^{d \times l}. \quad (1)$$

The attention is given using a normalization function  $\xi$  as

$$\bar{A}^{(t)} = g_Q(W^{(t)})g_K(\tilde{F}) \quad (2)$$

$$A^{(t)} = \xi(\bar{A}^{(t)}) \in (0, 1)^{z \times l}, \quad (3)$$

where the patterns  $W^{(t)}$  is updated by

$$U^{(t)} = A^{(t)}F'^T \quad (4)$$

$$W^{(t+1)} = \text{GRU}(U^{(t)}, W^{(t)}). \quad (5)$$

Let  $\text{Softmax}_R(X)$  and  $\sigma(X)$  be a softmax function over respective row vectors of matrix  $X$  and sigmoid respectively. MTUNet modulates this map by

$$A^{(t)} = \xi(\bar{A}^{(t)}) = \sigma(\bar{A}^{(t)}) \odot \text{Softmax}_R(\bar{A}^{(t)}), \quad (6)$$

which suppresses weak attention over different patterns at the same spatial location, where  $\odot$  is the Hadamard product. The function enforces the network to find more specific yet discriminative patterns with less redundancy among them, thus giving more pinpoint attention. This ensures the learned patterns are exclusive. As shown in Fig. 1(a), the attention map responds to a single pattern that rarely includes its peripheral region.

The input feature  $F$  is finally described by the overall attention  $A'$  corresponding to the extracted patterns, i.e.,

$$A' = \frac{1}{z} A^{(T)} \mathbf{1}_z \quad (7)$$

where  $\mathbf{1}_z$  is a row vector with all  $z$  elements aggregated being 1.  $A'$  is reshaped from  $l$  into the same spatial structure as  $F$ . Then the features corresponding to the overall attention are extracted and average pooled over the spatial dimensions as

$$V = \frac{1}{hw} \sum_{ij} A'_{ij} F_{ij}, \quad (8)$$

where  $A'_{ij} \in \mathbb{R}$  and  $F_{ij} \in \mathbb{R}^c$  are the elements of  $A'$  and  $F$  at the  $(i, j)$ -th spatial location ( $i = 1, 2, \dots, h$  and  $j = 1, 2, \dots, w$ ).

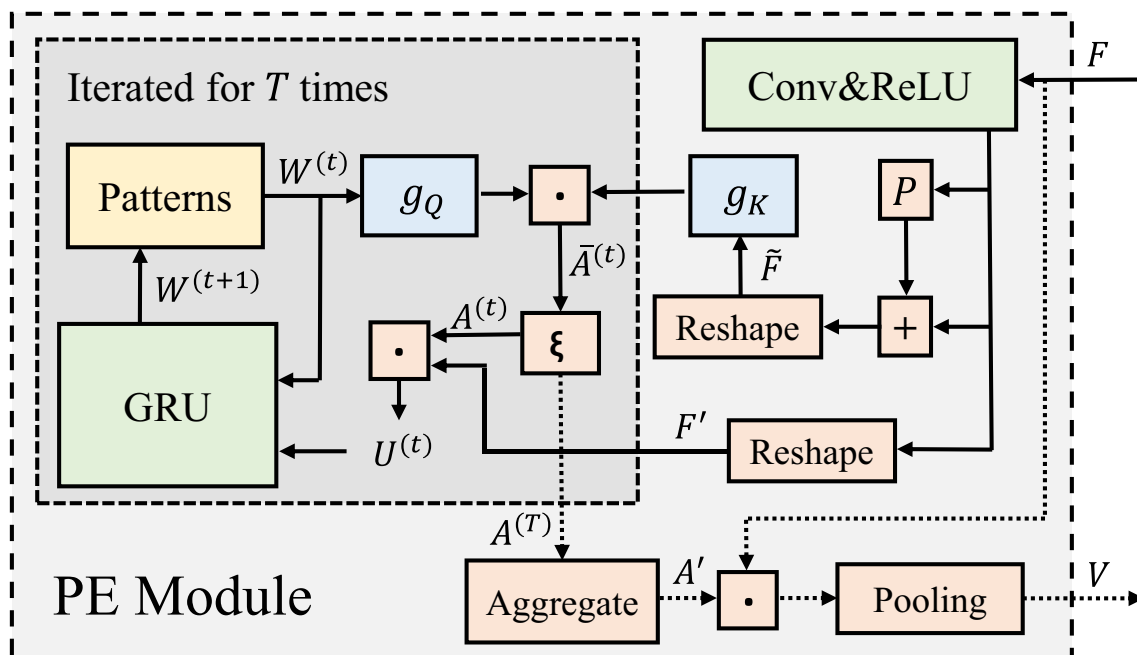


Fig. 3 The structure of our pattern extractor module

### 3.4 Pairwise matching

An FSL classification can be solved by finding the membership of a query in one of the given support images. Some FSL methods use metric learning [3, 7] to find matches between a query and the supports, and the cosine similarity or the  $\ell_2$  distance are typical choices [10, 48]. Learnable distances are another popular choice for metric learning-based FSL methods [11–13]. We use a learnable distance with an MLP (refer to Section 4.5.2).

Let  $V^q$  and  $\{V_{kn}\}$  be features obtained by applying the PE to query image  $x^q \in \mathcal{Q}$  and support images  $\{x_{kn}\}$  in  $\mathcal{S}$ , respectively, where the subscripts  $k = 1, 2, \dots, K$  and  $n = 1, 2, \dots, N$  stand for the  $n$ -th image of class  $k$  in the  $K$ -way  $N$ -shot episodic paradigm. For  $N > 1$ , the average over the  $N$  images are taken to generate representative feature  $\bar{V}_k$ ; otherwise (i.e.,  $N = 1$ ),  $\bar{V}_k = V_{k1}$ . For computing similarity score  $s$  between  $V^q$  and  $\bar{V}_k$ , we use MLP  $f_\gamma$  with learnable parameters  $\gamma$ :

$$s(V^q, \bar{V}_k) = \sigma(f_\gamma([V^q, \bar{V}_k])), \quad (9)$$

where  $[\cdot, \cdot]$  is concatenation.  $x^q$  is classified into class  $k^*$  with maximum  $s$  over  $k$ , i.e.,

$$k^* = \arg \max_k s(V^q, \bar{V}_k). \quad (10)$$

For a  $K$ -way task, our pairwise matching runs the similarity computation  $K$  times per query image, which is typical computational complexity for similarity-based methods, such as [7].

### 3.5 Training

For training, we sample a set  $\mathcal{Q} = \{(x_{km}^q, y_{km}^q) \mid i = 1, \dots, K \times M\}$  of  $M$  query images for  $K$  classes as well as set  $\mathcal{S}$  of support images from  $\mathcal{D}_{\text{base}}$  for each episode, following the  $K$ -way  $N$ -shot episodic paradigm. We train the model with the cross-entropy loss:

$$L = - \sum_{(x^q, y^q) \in \mathcal{Q}} \sum_{k=1}^K y_k^q \log(\bar{s}(V^q, \bar{V}_k)), \quad (11)$$

where  $y_k^q$  is the  $k$ -th element of one-hot vector  $y^q$  for representing the corresponding label of image  $x^q$ .

## 4 Experiments

### 4.1 Datasets

We evaluate our approach on three commonly-used datasets, mini-ImageNet [3], tiered-ImageNet [22], and CIFAR-FS [49]. **Mini-ImageNet** consists of 100 classes sampled from ImageNet with 600 images per class. These images are

divided into the base  $\mathcal{D}_{\text{base}}$ , novel validation  $\mathcal{D}_{\text{val}}$ , and novel test  $\mathcal{D}_{\text{test}}$  sets with 64, 16, and 20 classes, respectively, where both  $\mathcal{D}_{\text{val}}$  and  $\mathcal{D}_{\text{test}}$  corresponded to  $\mathcal{D}_{\text{novel}}$  in Section 3.1. The images in miniImageNet are of size  $84 \times 84$ . As all recent work, we adopt the same splits of [3] **Tiered-ImageNet** consists of ImageNet 608 classes divided into 351 base classes, 97 novel validation classes, and 160 novel test classes. There are 779,165 images with size  $84 \times 84$ . **CIFAR-FS** is a dataset with images sampled from CIFAR-100 [50]. This dataset contains 100 classes with 600 images each. We follow the split given in [49], which are 64, 16, and 20 classes for the base, novel validation, and novel test sets.

### 4.2 Experimental setup

Following most of the literature, we evaluate MTUNet on 10,000 episodes of 5-way classification created by first randomly sampling 5 classes from  $\mathcal{D}_{\text{base}}$  and then sampling support and query images of these classes with  $N = 1$  or 5 and  $M = 15$  per class. We report the average accuracy over  $K \times M = 75$  queries in the 10,000 episodes and the 95% confidence interval. We employ three CNN architectures as our backbone  $f_\theta$ , which are often used for FSL tasks, namely Conv-4 [7], WRN-28-10 [51] and ResNet-18 [52]. For ResNet-18, we remove the first two down-sampling layers and change the kernel of the first  $7 \times 7$  convolutional layer to  $3 \times 3$ . We use the hidden vector of the last convolutional layer after ReLU as feature maps  $F$ , where the numbers of feature maps are 512 and 640 for ResNet-18 and WRN-28-10 respectively. There are three steps for training MTUNet.

**Pre-training of backbone** The pre-training of the backbone CNNs is important for our PE module. We adopted a distance-based strategy, which is similar to SimpleShot [10]. We train the backbone CNNs with all images in  $\mathcal{D}_{\text{base}}$ . The performance of a simple nearest-neighbour-based method is then evaluated over  $\mathcal{D}_{\text{val}}$  with 2,000 episodes of 5-way FSL tasks, and the best model is adopted. The learning rate for training starts at  $10^{-3}$  and is divided by 10 every 20 epochs. We train the models for 50 epochs.

**Pre-training of PE** As for the PE module pre-training, we set  $d$  to 64, and the number  $T$  of the update is set to 3. The number  $z$  of the patterns is empirically set to 1/10 of the number of classes in the base set, which are 7, 36, and 7 for the mini-ImageNet, tiered-ImageNet, and CIFAR-FS dataset, respectively. Corresponding number of classes' (subset of  $\mathcal{C}_{\text{base}}$ ) images are selected to pre-train the module as a normal classification task similar to [38]. The importance of this choice is discussed in Section 4.5.1. Both  $g_Q$  and  $g_K$  have three FC layers with ReLU nonlinearities between them. All the parameters in the backbone  $f_\theta$  are fixed. The

learning rate for training starts with  $10^{-4}$  and is divided by 10 at the 40th epoch, and the total number of epochs is 60.

**Training the whole network** For training the whole MTUNet, the learnable parameters in the backbone CNNs and PE are optimized with a small learning rate of  $10^{-5}$ . We completely implement 20 training epochs. In a single training epoch, we sample 1,000 episodes of 5-way tasks. Other learnable parts of the model are trained to start with an initial learning rate of  $10^{-4}$ , which is divided by 10 at the 10th epoch. We save the model with the best performance on 2,000 episodes evaluation sampled from  $\mathcal{D}_{\text{val}}$ .

Our model is implemented with PyTorch, and AdaBelief [53] is adapted as an optimizer. Input images are resized to  $80 \times 80$ , and we applied data augmentation including

random flip and an affine transformations, following [10]. A GPU workstation with two NVIDIA Quadro GV100 (32GB memory) GPUs is used for all experiments. Training 20 epochs on the mini-ImageNet dataset took approximately 19 minutes with a single NVIDIA V100 GPU. This computational cost is not high. We attested that a consumer-grade GPU can easily reproduce our results.

### 4.3 Few-shot classification results

MTUNet is compared with some popular FSL methods. We exclude methods in semi-supervised and transductive paradigms, which use the statistics of novel set across different FSL episodes. Besides the classification accuracy, we also consider the explainability of the raw image features

**Table 1** Average accuracy of 10000 episodes of 5-way tasks on the mini-ImageNet dataset test set

Approach	Backbone	One shot	Five shots
SimpleShot (UN) [10]	Conv-4	33.17±0.17	63.25±0.17
MetaLSTM [17]	Conv-4	43.44±0.77	60.60±0.71
MatchingNet [3]	Conv-4	43.56±0.84	55.31±0.73
MAML [16]	Conv-4	48.70±1.84	63.11±0.92
LLAMA [54]	Conv-4	49.40±1.83	–
ProtoNet [7]	Conv-4	49.42±0.78	68.20±0.66
PLATIPUS [55]	Conv-4	50.13±1.86	–
GNN [12]	Conv-4	50.33±0.36	66.41±0.63
RelationNet [13]	Conv-4	50.44±0.82	65.32±0.70
Meta SGD [19]	Conv-4	50.47±1.87	64.03±0.94
AD2AML+IR [56]	Conv-4	54.57±1.77	–
RCNet [57]	Conv-4	<b>54.85±0.84</b>	68.92±0.77
MTUNet (w/o PE)	Conv-4	51.20±0.32	65.88±0.39
MTUNet	Conv-4	54.01±0.37	<b>69.43±0.46</b>
<hr/>			
MAML [16] <sup>†</sup>	ResNet-18	49.61±0.92	65.72±0.77
R2-D2 [49] <sup>†</sup>	ResNet-18	51.20±0.60	68.20±0.60
RelationNet [13] <sup>†</sup>	ResNet-18	52.48±0.86	69.83±0.68
ProtoNet [7] <sup>†</sup>	ResNet-18	54.16±0.82	73.68±0.65
MTUNet (workshop) [4]	ResNet-18	55.03±0.49	70.22±0.55
Gidaris [48]	ResNet-15	55.45±0.89	70.13±0.68
SNAIL [8]	ResNet-15	55.71±0.99	68.88±0.92
AdaCNN [58]	ResNet-18	56.88±0.62	71.94±0.57
SimpleShot (UN) [10]	ResNet-18	57.81±0.21	<b>80.43±0.15</b>
MTUNet (w/o PE)	ResNet-18	55.27±0.33	67.51±0.39
MTUNet	ResNet-18	<b>58.13±0.44</b>	75.02±0.43
<hr/>			
MTUNet (workshop) [4]	WRN	56.52±0.43	71.93±0.40
SimpleShot (UN) [10]	WRN	57.26±0.21	78.99±0.14
Qiao [59]	WRN	59.60±0.41	73.74±0.19
MTUNet (w/o PE)	WRN	56.41±0.33	69.55±0.39
MTUNet	WRN	<b>60.12±0.45</b>	<b>79.23±0.42</b>

The results of our method are in orange

<sup>†</sup> Results are reported in [10]

The bold line is used to separate different settings for comparison

**Table 2** Average accuracy of 10000 episodes of 5-way tasks on the tiered-ImageNet dataset test set

Approach	Backbone	One shot	Five shots
Reptile [18] <sup>†</sup>	Conv-4	48.97±0.21	66.47±0.21
SimpleShot (UN) [10]	Conv-4	51.02±0.20	68.98±0.18
MAML [16]	Conv-4	51.67±1.81	70.30±0.08
ProtoNet [7] <sup>†</sup>	Conv-4	53.31±0.20	72.69±0.74
RelationNet [13]	Conv-4	54.48±0.93	71.32±0.78
AD2AML+IR [56]	Conv-4	54.97±1.92	–
RCNet [57]	Conv-4	58.42±0.96	<b>74.17±0.78</b>
MTUNet (w/o PE)	Conv-4	57.02±0.58	70.94±0.52
MTUNet	Conv-4	<b>59.12±0.61</b>	73.31±0.65
MTUNet (workshop) [4]	ResNet-18	61.27±0.50	77.82±0.41
SimpleShot (UN) [10]	ResNet-18	62.69±0.22	79.69±0.15
MTUNet (w/o PE)	ResNet-18	60.21±0.42	77.26±0.41
MTUNet	ResNet-18	<b>63.83±0.53</b>	<b>82.07±0.46</b>
MTUNet (workshop) [4]	WRN	62.40±0.51	80.05±0.46
Meta SGD [19] <sup>†</sup>	WRN	62.95±0.03	79.34±0.06
SimpleShot (UN) [10]	WRN	64.35±0.23	85.69±0.15
LEO [60]	WRN	66.33±0.05	81.44±0.09
MTUNet (w/o PE)	WRN	62.11±0.30	78.40±0.35
MTUNet	WRN	<b>66.52±0.48</b>	<b>86.17±0.41</b>

The results of our method are in orange

<sup>‡</sup> Results are reported in [49]

The bold line is used to separate different settings for comparison

for the backbone CNNs. Thus, we do not adopt any post-processing methods like  $\ell_2$  normalization in [10]. For testing the model, we report our best model on  $\mathcal{D}_{val}$  by randomly sampling 10,000 1-shot and 5-shots tasks from  $\mathcal{D}_{test}$  in Tables 1, 2 and 3 over the three datasets. During testing, taking a 1-shot task for example, our model assigns the query image to one of the classes of support images. It is realized by (i) extracting regions from each of query and

support images and extracting features from these regions with PE and (ii) matching the features with PM. The results of MTUNet (w/o PE) means the model trained without the PE module. This model has a structure similar to ProtoNet [7] and is used to evaluate the impact of the PE.

As seen in the tables, the prediction accuracy of MTUNet outperforms most existing FSL methods in both one-shot and five-shots settings. This proves that our model can

**Table 3** Average accuracy of 10000 episodes of 5-way tasks on the CIFAR-FS dataset test set

Approach	Backbone	One shot	Five shots
RelationNet [13] <sup>‡</sup>	Conv-4	55.00±1.00	69.30±0.80
ProtoNet [7] <sup>‡</sup>	Conv-4	55.50±0.70	72.00±0.60
MAML [16] <sup>‡</sup>	Conv-4	58.90±1.90	71.50±1.00
GNN [12] <sup>‡</sup>	Conv-4	61.90	75.30
R2-D2 [49]	Conv-4	65.30±0.20	<b>78.30±0.20</b>
MTUNet (w/o PE)	Conv-4	62.55±0.51	74.62±0.54
MTUNet	Conv-4	<b>65.81±0.65</b>	77.42±0.60
MTUNet (w/o PE)	ResNet-18	65.32±0.37	79.54±0.34
MTUNet	ResNet-18	<b>67.47±0.43</b>	<b>82.81±0.41</b>
MTUNet (w/o PE)	WRN	67.29±0.39	82.98±0.35
MTUNet	WRN	<b>70.49±0.46</b>	<b>86.55±0.44</b>

The results of our method are in orange

The bold line is used to separate different settings for comparison



achieve high prediction accuracy for FSL tasks. We also find that the different architectures of the backbone CNNs affect the performance. With simple backbone structure, Conv-4 tends to produce a lower performance. The variants with WRN always have a better performance than those with Conv-4 and ResNet-18. Besides from the difference in the network architecture, the size of feature maps may be one of the factors. On the mini-ImageNet dataset, the WRN variants have  $20 \times 20$  feature maps, while the ResNet-18 variants have  $10 \times 10$ . Such larger feature maps not only provide more information to the PM module but also give a better basis of patterns as higher resolutions may help find more specific patterns. The results also demonstrate the learning ability of the PE. For all experiment settings, the PE can improve the model accuracy by approximately 2%-4% more than without the PE. This module filters useless features and focus on informative regions as it is designed to be. We will further analyse the importance of pattern number  $z$  and PE pre-training categories selection for training MTUNet in Section 4.5.1.

#### 4.4 Explainability

In this section we will qualitatively and quantitatively evaluate the explainability of MTUNet.

##### 4.4.1 Qualitative evaluation

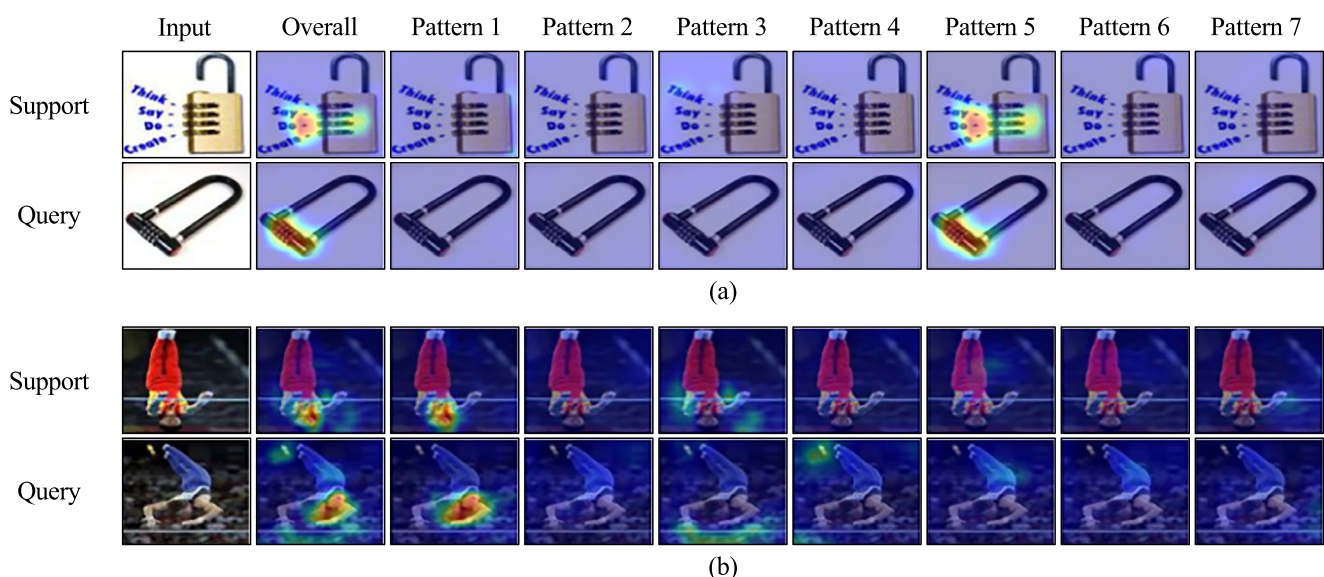
In addition to the classification performance, MTUNet is designed to be explainable in two different aspects. First, pattern-based visual explanation. MTUNet's decision

is based on certain combinations of learned patterns. These patterns are localized in both query and support images through  $A^{(T)}$ , which can be easily visualized. This visualization offers intuition on the learned patterns and how much these patterns are shared between the query and support images. Second, visualization of pairwise matching scores. Thanks to the one-to-one matching strategy formulated as a binary classification problem in (9), the distributions (or appearances) of learned patterns in query and support images give a strong clue on MTUNet's matching score  $s$ . In this combination, we may find the potential failure reasons by observing the matching matrix.

**Pattern-based visual explanation** MTUNet's decision is based on learned patterns, i.e., it is solely based on how much shared patterns (or features) appear in both query and support images. This design in turn means that, by pinpointing each pattern in the images, we can obtain an intuition behind the decision made by the model. This can be done by merely visualizing  $A^{(T)}$ .

Figures 4(a) and (b) show a pair of support and query images in the mini-ImageNet dataset for a 5-way task. The pairs (a) and (b) are of classes `lock` and `horizontal bar`, respectively. The second column shows the visualization of the aggregated overall attention, given by  $A'$ . The third to ninth columns are the visualization of the regions corresponding to the learned patterns in  $A^{(T)}$  (i.e., the  $i$ -th row vector of  $A^{(T)}$  represents the appearance of the  $i$ -th learned pattern at the respective spatial location).

For (a) with class `lock`, the support image is a small gold combination lock used for storage cabinets or post



**Fig. 4** Visualization of each pattern and the average features for a sampled task in the mini-ImageNet dataset. (a) is the lock class and (b) is the horizontal bar class. Overall is the overall attention among all

patterns. The third to ninth columns are the visualization of the regions corresponding to the learned patterns

boxes. Among all 7 patterns, only pattern 5 shows a strong response, whereas the others are not observed. We can see that pattern 5 pays attention to the discs of the lock in the support image. It also provides a strong response to the words on the left which shows similar morphological characteristics. The query image in (a) is a black combination lock often used for bicycles. The attention maps show almost the same distributions as the support, that is, only pattern 5 has a response on the discs. From these visualizations, we can infer that pattern 5 represents the character of the discs. MTUNet successfully finds a shared pattern although these two locks have a different appearance.

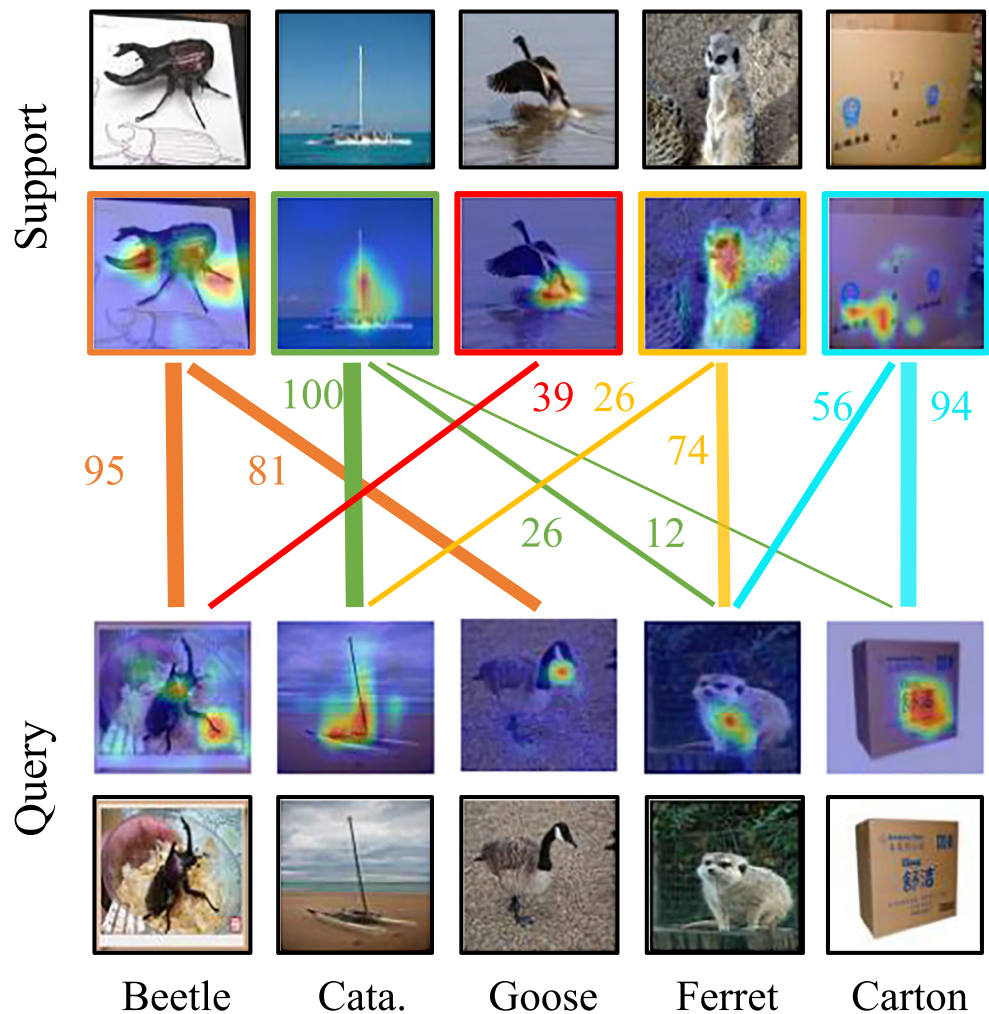
For (b), the support image is a gymnast wearing red. Multiple patterns are observed in the image. We can see that the visualization of pattern 1 identifies part of the human body (head), and pattern 3 appears around the hands grabbing the horizontal bar. The query image is a gymnast in blue. Patterns 1 and 3 respond in a similar way to the support image. Patterns 4 and 5 appear in the background and around other parts of the body, however, their responses are

relatively weak compared to patterns 1 and 3. Patterns 1 and 3 may be responsible for human heads and hands grabbing the horizontal bar, leading to the successful classification of the unseen classes.

**Visualization of pairwise matching scores** Figure 5 shows the visualized overall attentions  $A'$  and corresponding origin support and query images (a 5-way 1-shot task on the mini-ImageNet dataset). Through the pairwise matching module, the FSL task is cast into a binary classification problem. The output for each pair is a value between 0 and 1 due to the sigmoid function, whereas the scores are shown as percentages in the figure. The support images are marked with different colors to represent the classes. The thickness of coloured lines shows higher or lower matching score between each support and query. Only pairs with a score over 0 are shown in the figure.

Among all pairwise combinations, the combination of the support and query images of the *catamaran* obtains a full score (100%). The visualization of the overall attention covers the hulls, especially the masts, in both images, which

**Fig. 5** Matching point of one sampled task in the mini-ImageNet dataset. We only show the connection between pairs with a score over 0, and the scores are shown as percentages



are the main characteristics of this class. Class `goose` gets a low matching score. The query is a close-up of a goose on the ground from its front side, which captures the goose's blackhead or beak. The support image is an overall view of a goose about to fly and the visualization of the overall attention captures the leg. With this combination, finding a shared pattern may not be easy, although these two extracted patterns are both representative parts of a bird. This problem stems from differences in viewing angles, which can be relieved in 5-shot tasks, giving more support from different viewing angles. Surprisingly, the query image for `goose` obtains 81% for the support image for `beetle`. This may suggest that one of the patterns responds to black regions and this pattern is solely used as the clue of `goose`. This is a negative result for the FSL task but clearly demonstrates MTUNet's explainability on the relationship between visual patterns and the matching scores.

We also provide more visualization samples in Appendix A.

#### 4.4.2 Quantitative evaluation

Our method is designed to interpret FSL tasks, and we think it necessary to compare the explainability of MTUNet with previous XAI methods using existing metrics. We adopt MTUNet without the PE with ResNet-18 as the baseline model and use existing XAI methods for explanations (We consider our PE module as the explainable module. After removing the PE, our model has a similar structure to ProtoNet). We conduct 10000 episodes of 5-way 1-shot tasks, obtain the visual explanations for each task using several types of XAI methods, and compare these explanations to the overall attention map  $A'$  generated by our method (MTUNet ResNet-18).

We adopt three evaluation metrics for comparison. (i) Precision: We donate an input image as  $x$  and the foreground bounding box by  $\bar{x}$  (provided by ImageNet [61]). Thus, we can compute the area ratio of explanation within the bounding box by the Precision =  $\sum_{p \in \bar{x}} A'(p) / \sum_{p \in x} A'(p)$ , where

$A'(p)$  is the attention value in  $A'$  at pixel  $p$  and  $A'$  is resized to the same size as the input image. (ii) Insertion area under the curve (IAUC) [62]: This metric calculates the accuracy gain of the model when gradually adding image pixels in the order of importance given by the explanation. (iii) Deletion area under the curve (DAUC) [62]: This metric measures the accuracy drop when gradually removing important pixels from the input image. As shown in Table 4, the explanation of MTUNet outperforms existing XAI methods in all three evaluation metrics, which demonstrates the strong explainability of the proposed method. We think our *intrinsic* method has the advantage for the interpretation of FSL tasks. Due to the FSL sampling training strategy, both *back-prop* and *perturbation* methods may lack the ability to analyze such complex scenarios. While our method can provide an explanation within a simple inference step.

## 4.5 Discussion

### 4.5.1 Pattern setting

The pattern number  $z$  and categories selected for PE pre-training are important elements for training the whole MTUNet. In this section, we will analyse them from these two aspects.

**The number  $z$  of patterns** The number of patterns can be another crucial factor for MTUNet. Intuitively, a larger  $z$  makes the model more discriminative. To show the impact of  $z$ , we uniformly sample classes in  $C_{\text{base}}$  (i.e., defaulting to sampling every  $I$  classes from the class list, where  $I = 10, 8, 7, 5, 4, 3, 2$ , and 1); thus,  $I=1$  uses all classes in  $C_{\text{base}}$ .

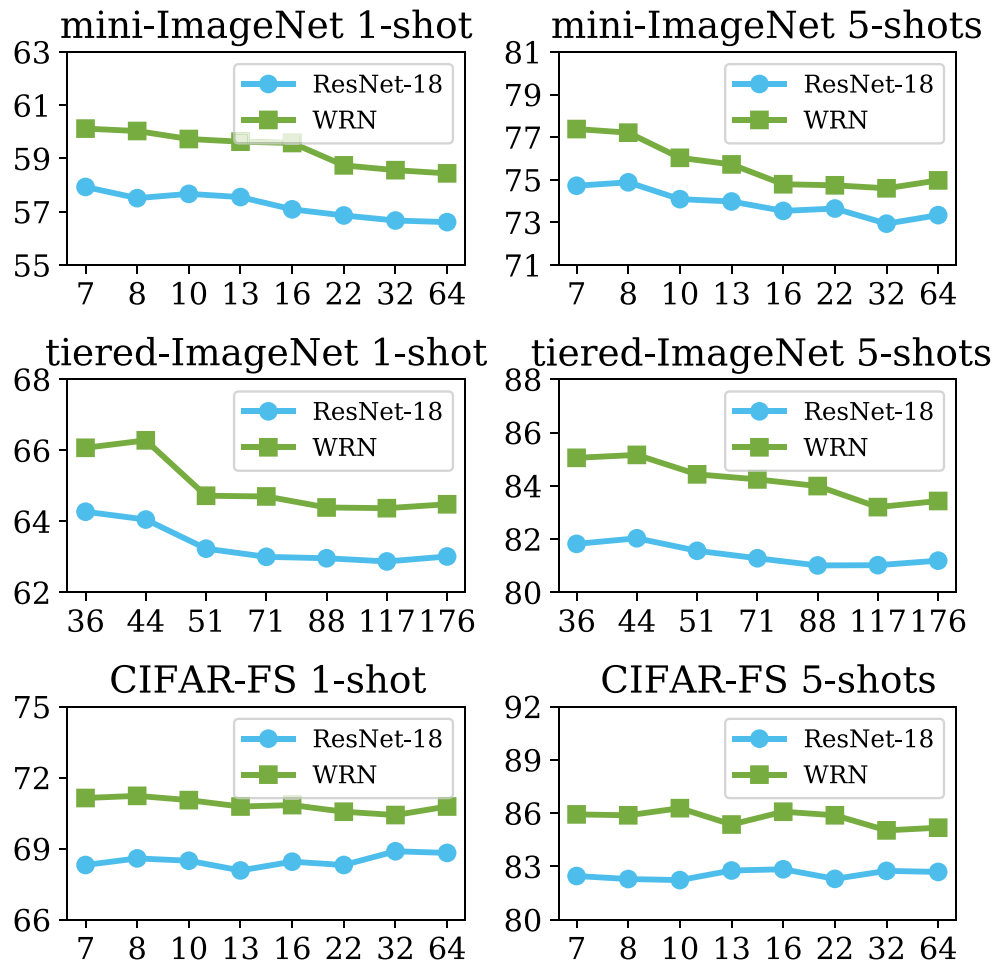
The test accuracies are shown in Fig. 6 for 5-way 1-shot and 5-way 5-shot tasks on 10,000 sampled episodes over  $D_{\text{test}}$  of the three datasets. The horizontal axis represents the number of patterns and the vertical axis represents the average accuracy. We would say that the performance has

**Table 4** Evaluation of MTUNet and existing XAI methods using explainability metrics

Methods	Type	mini-ImageNet		
		Precision $\uparrow$	IAUC $\uparrow$	DAUC $\downarrow$
DeepLIFT [63]	Back-Prop	0.728	0.680	0.131
GradCAM [35]	Back-Prop	0.807	0.712	0.116
GradCAM++ [36]	Back-Prop	0.826	0.735	0.107
Score-CAM [34]	Back-Prop	0.811	0.702	0.110
SS-CAM [64]	Back-Prop	0.791	0.720	0.114
RISE [62]	Perturbation	0.757	0.753	0.098
IBA [37]	Perturbation	0.871	0.764	0.096
MTUNet	Intrinsic	<b>0.902</b>	<b>0.793</b>	<b>0.091</b>

The bold line is used to separate different settings for comparison

**Fig. 6** Results of pattern number settings for the mini-ImageNet, tiered-ImageNet, and CIFAR-FS dataset. The horizontal axis represents the number of patterns and the vertical axis represents the average accuracy. We report the results with 10,000 sampled 5-way episodes in the novel test set



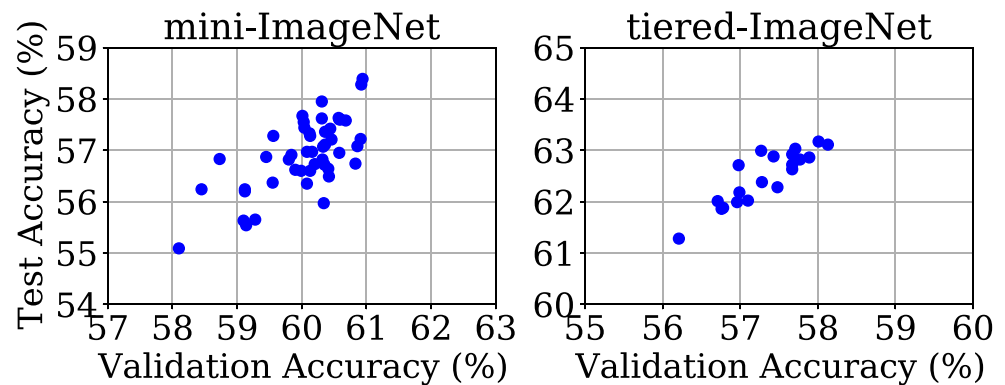
no obvious changes on the CIFAR-FS dataset as the number of  $z$  changes, whereas it has slightly decreased results on the mini-ImageNet dataset (approximately 1% for 1-shot and 2% for 5-shots). For the tiered-ImageNet dataset, when setting the pattern number to 51, an obvious performance drop is observed for the WRN backbone (approximately 3.5% for 1-shot), while this does not happen for the 5-shot setting. In general, tuning  $z$  may help gain performance, but its impact is not significant. It requires tuning the number  $z$  of patterns for each backbone and dataset. Since a small value of  $z$  can provide both high classification accuracy and convince the visualization of each pattern (e.g. Fig. 4), we recommend setting  $z$  to a small value according to the class number of the dataset. However, it might be an interesting research direction to estimate  $z$ , e.g., based on the number of classes in a given FSL task.

**Selection of classes for PE pre-training** Our PE module is supposed to learn common visual patterns. We use images of a certain subset of classes in  $C_{base}$  to learn the initialization of such patterns in our experiments. The selection of this subset thus affects the performance of downstream FSL tasks. To clarify the impact of the choice of the subset,

we randomly sample 7 classes 50 times in  $C_{base}$  of the mini-ImageNet dataset, and 36 classes 20 times in the tiered-ImageNet dataset, and use the corresponding images for the training PE on top of ResNet-18. The trained PE is used for training MTUNet, which is evaluated over 2,000 episodes of FSL tasks with both the validation and test sets.

Figure 7 left shows a scatter plot of the validation accuracies and corresponding test accuracies. The mean and the 95% confidence interval over the 50 test accuracies for the mini-ImageNet dataset are 56.83% and 0.18%, respectively. This implies that our model benefits from a better choice of classes for PE pre-training. For this choice, we only have access to the validation set; since the validation set and the test set have disjointed classes, the best choice for the validation set is not necessarily the best choice for the test set. While, the plot empirically shows that the validation and test accuracies are highly correlated to each other, with a Pearson’s correlation coefficient of 0.71. We also implemented the experiments on the tiered-ImageNet dataset with 20 random samplings of 36 classes, which shows similar results. The results above lead to the conclusion that MTUNet is sensitive to the PE pre-training, however, we can use the validation set to find the best choice.

**Fig. 7** Performance of random classes sampling for PE pre-training of patterns. All experiments are implemented on the mini-ImageNet and tiered-ImageNet dataset using ResNet-18 as the backbone



#### 4.5.2 Selection of metric learning methods

In our experiments, we find that a learnable metric by an MLP achieves the best FSL classification performance over commonly used predefined metrics, such as the Euclidean distance and the cosine similarity. As shown in Table 5, we can observe that the MLP performs the best for all backbone settings on the mini-ImageNet dataset. The accuracy difference is small for Conv-4 but noticeable for ResNet-18 and WRN. We can infer that the MLP better deals with features extracted from a larger backbone.

#### 4.5.3 Limitations and future work

Our experiments have shown that training MTUNet from scratch (i.e., without pre-training) was infeasible and that it even required two pre-training steps. The number  $z$  of patterns to be learned is a hyperparameter to be tuned for the given dataset. Pre-training of PE is sensitive to selection of classes. To address these drawbacks, we will study the relationship between the numbers of classes and patterns. This also requires to investigate the impact of different datasets in the training process. We will keep working on improving the training strategy to make it more agnostic to class selection in pre-training.

The core of MTUNet's explainability lies in observing the combination of pattern-based visualization and matching scores. However, evaluation of this aspect is not straightforward because the patterns are learned in the course of training without supervision and thus there is no ground-truth.

Due to this, we relied on quantitative evaluation to demonstrate the usability of MTUNet's explainability. This problem may be mitigated by using or building a dataset with fine-grained annotation on possible patterns.

An interesting future direction of MTUNet is to extend it to different types of real-world data, other than images, such as videos and 3D medical images. Research in this direction has been already explored for some tasks, such as shot boundary and key frame detection [65, 66] and lesion localization [67, 68]. MTUNet's extracted patterns may offer better explanation for tasks in these domains.

## 5 Conclusion

In this paper, we proposed MTUNet designed for explainable FSL classification tasks. Our model achieved higher classification performance than existing FSL methods on three benchmark datasets. The PE module serves to only include informative regions of image features extracted by CNNs backbone. It can learn better representations and is proved to be a necessary structure for improving prediction accuracy.

Our experiment results also quantitatively and qualitatively demonstrated MTUNet's strong explainability through patterns in images. Compared to the heatmap-alone explanations provided by existing methods, our explanation can be realized through the combination of pattern-based visual explanation and pairwise matching scores which offer a better proof basis for model decision analysis. With this

**Table 5** Performance of different metric learning methods

Methods	Conv-4		ResNet-18		WRN	
	One shot	Five shots	One shot	Five shots	One shot	Five shots
Cosine Similarity	53.47±0.27	67.44±0.29	56.72±0.35	70.96±0.38	58.23±0.40	73.15±0.42
Euclidean Distance	53.25±0.22	67.12±0.28	56.01±0.32	71.54±0.36	57.85±0.35	74.79±0.38
MLP	<b>54.01±0.37</b>	<b>69.43±0.46</b>	<b>58.13±0.44</b>	<b>75.02±0.43</b>	<b>60.12±0.45</b>	<b>79.23±0.42</b>

All the experiments are implemented on the min-ImageNet dataset  
The bold line is used to separate different settings for comparison

combination, we can further manually analyse the reason for failure cases, which is important to some high-risk areas (e.g. medical tasks). In addition, the approach taken in our model might be analogous to humans as we usually try to find shared patterns when making a match between images of an object that has never seen before. This can be advantageous since the explanation given by MTUNet can provide an intuitive interpretation (*intrinsic*) of what the model does.

### Appendix A: Qualitative results of MTUNet

We provide visualization of patterns for 3 randomly sampled 5-way 1-shot tasks with a single query image per class in the mini-ImageNet dataset. The pattern-based visualization (Figs. 8, 10, 12) and the pairwise matching scores (Figs. 9, 11, 13, row and column are consistent with the overall attention visualization for support and query of

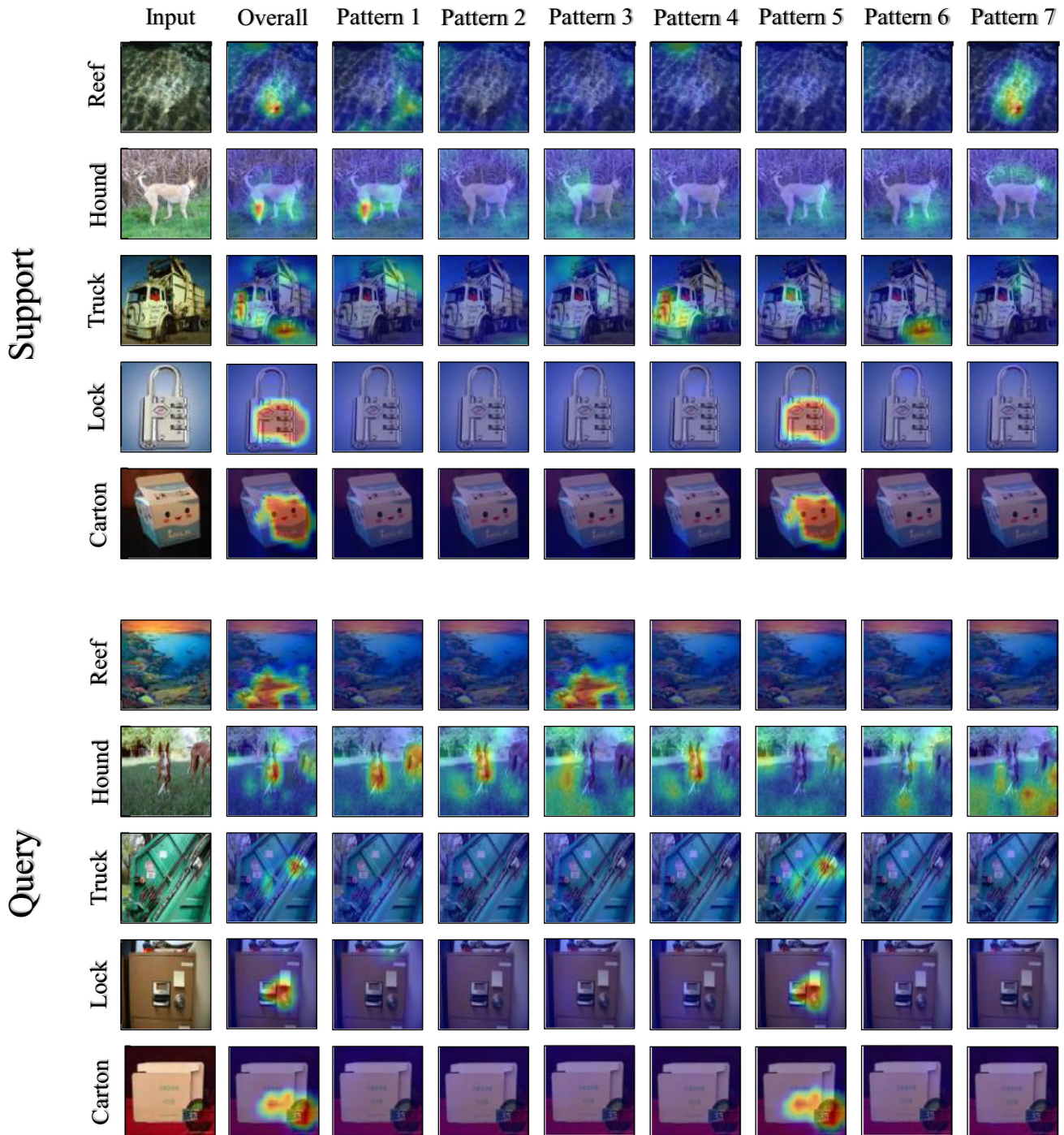
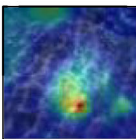



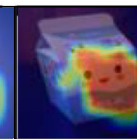
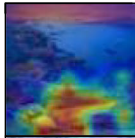
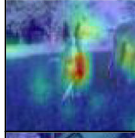
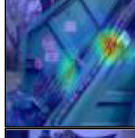
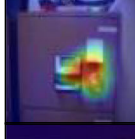
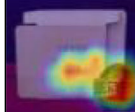


Fig. 8 Pattern-based visualization of sample 1

**Fig. 9** Pairwise matching of sample 1

	Reef	Hound	Truck	Lock	Carton
Support					
Query					
	100	0	0	0	0
	0	100	0	0	1
	0	0	47	8	62
	0	0	0	96	100
	0	0	2	93	100

each category, with the scores shown as percentages) are shown for samples 1–3, respectively. We also provide some discussion on the respective samples.

**Sample 1** By observing the matching matrix in Fig. 9, we find there are two confusing categories of *lock* and *carton*. They all obtain a high score for each other category. The visualization in Fig. 8 shows that pattern 5 is responsible for both the letters (or a face of a character) on the carton and the discs of the lock. We would say that the letters and the discs share some similar structures, which causes the confusion.

**Sample 2** As shown in Fig. 11, the pairwise matching scores for this sample find proper matches except for *poncho*. In Fig. 10, the *poncho* support image is a baby girl wearing a poncho, while the query image is just a poncho with black color on a white background. The query

image for *poncho* yields high scores for the support images of *poncho*, *skirt*, and *beetle*. The highest score of *beetle* may be due to the black colour. Interestingly, the support and query images for *skirt* shows the attention over the door behind the person but not over the skirt itself. This is a good example of the importance of explanation for FSL.

**Sample 3** In Figs. 12 and 13, we find both the query and support give attention to the body part of the goose, but the differences in the perspective and the number of objects may make matching difficult. As a result, the query *goose* gets low scores for all support images. This also happens for *carton* in this sample. On the contrary, for the prediction of *truck*, it obtains a high score of 94. We can observe pattern 5 catch the wheel part for both the support and query images.

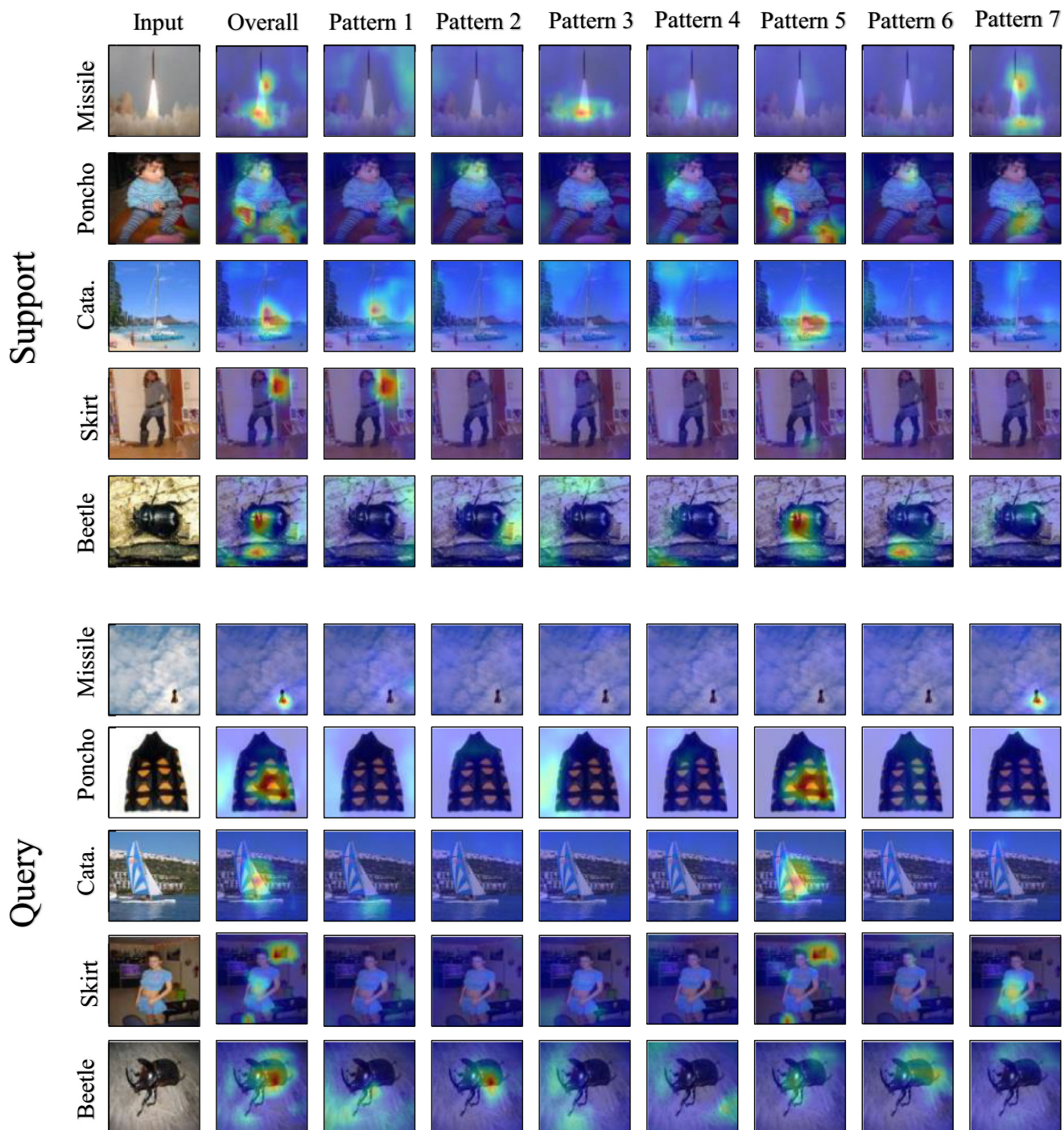
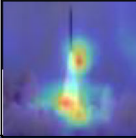



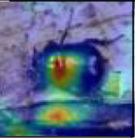

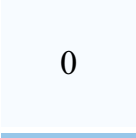
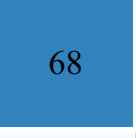
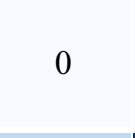
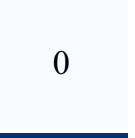

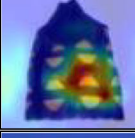


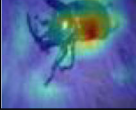


Fig. 10 Pattern-based visualization of sample 2



**Fig. 11** Pairwise matching of sample 2

	Missile	Poncho	Cata.	Skirt	Beetle
Support					
Query					
	100	0	68	0	0
	0	47	0	74	99
	28	0	100	0	0
	0	20	0	92	0
	0	4	0	33	94

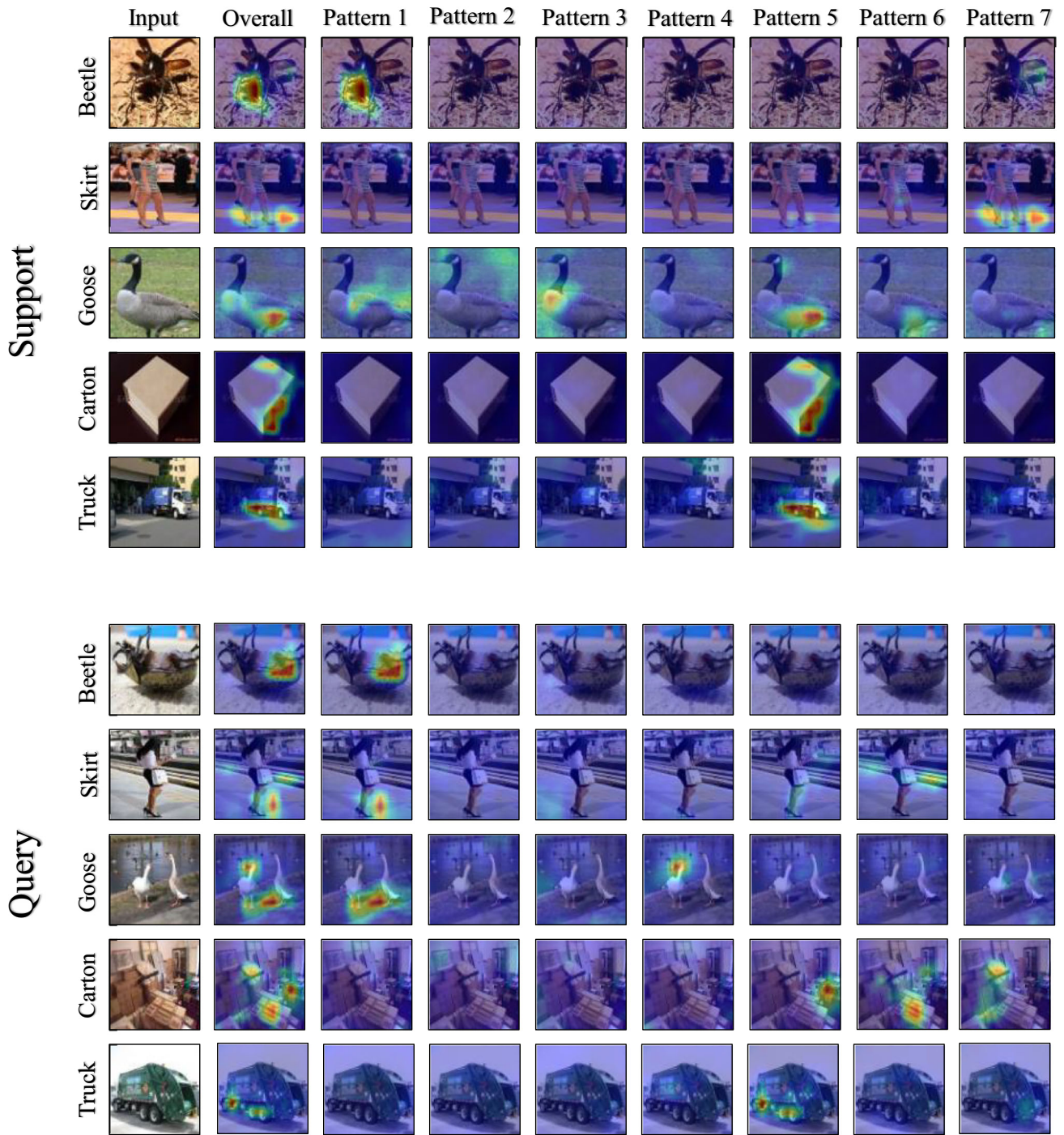
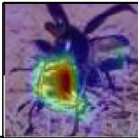


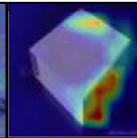


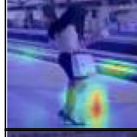

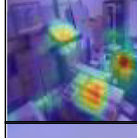



Fig. 12 Pattern-based visualization of sample 3

**Fig. 13** Pairwise matching of sample 3

	Beetle	Skirt	Goose	Carton	Truck
<b>Support</b>					
<b>Query</b>					
	98	0	96	0	0
	3	69	1	0	25
	0	0	11	0	1
	6	60	4	0	97
	0	1	1	0	94

**Acknowledgements** This work was supported by the Council for Science, Technology and Innovation (CSTI), the Cross-ministerial Strategic Innovation Promotion Program (SIP), the “Innovative AI Hospital System” (Funding Agency: National Institute of Biomedical Innovation, Health and Nutrition (NIBIOHN)), and the JST FOREST Grant No. JPMJFR216O. This work was also supported by JSPS KAKENHI Grant Number 21K17764 and 19K10662.

**Data Availability** All the data used in this paper are open-source data.

## Declarations

**Competing interests** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Wang Y, Yao Q, Kwok JT, Ni LM (2020) Generalizing from a few examples: a survey on few-shot learning. *ACM Comput Surv* 53(3):1–34
2. Yue Z, Zhang H, Sun Q, Hua X-S (2020) Interventional few-shot learning. *NeurIPS* 33:2734–2746
3. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. In: *Proceeding NeurIPS*, pp 3630–3638
4. Wang B, Li L, Verma M, Nakashima Y, Kawasaki R, Nagahara H (2021) MTUNET: few-shot image classification with visual explanations. In: *Proceeding CVPR workshops*, pp 2294–2298
5. Prabhu VU (2019) Few-shot learning for dermatological disease diagnosis. PhD thesis, Georgia institute of technology
6. Feyjje AR, Azad R, Pedersoli M, Kauffman C, Ayed IB, Dolz J (2021) Semi-supervised few-shot learning for medical image segmentation. *IEEE Int Conf Bioinform Biomed*
7. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In: *Proceeding NeurIPS*, pp 4077–4087
8. Mishra N, Rohaninejad M, Chen X, Abbeel P (2018) A simple neural attentive meta-learner ICLR
9. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. In: *ICML deep learning workshop*, vol 2
10. Wang Y, Chao W-L, Weinberger KQ, Van Der Maaten L (2019) Simpleshot: revisiting nearest-neighbor classification for few-shot learning. [arXiv:1911.04623](https://arxiv.org/abs/1911.04623)
11. Kim J, Kim T, Kim S, Yoo CD (2019) Edge-labeling graph neural network for few-shot learning. In: *Proceeding CVPR*, pp 11–20

12. Garcia V, Bruna J (2018) Few-shot learning with graph neural networks. ICLR
13. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: relation network for few-shot learning. In: *Proceeding CVPR*, pp 1199–1208
14. Wang J, Song B, Wang D, Qin H (2022) Two-stream network with phase map for few-shot classification. *Neurocomputing* 472:45–53
15. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: *Proceeding NeurIPS*, pp 3320–3328
16. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. ICML
17. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning. ICLR
18. Nichol A, Achiam J, Schulman J (2018) On first-order meta-learning algorithms. arXiv:1803.02999
19. Li Z, Zhou F, Chen F, Li H (2017) Meta-SGD: learning to learn quickly for few-shot learning. ICML
20. Hu Z, Li Z, Wang X, Zheng S (2022) Unsupervised descriptor selection based meta-learning networks for few-shot classification. *Pattern Recogn* 122:108304
21. Simon C, Koniusz P, Harandi M (2022) Meta-learning for multi-label few-shot classification. In: *Proceeding WACV*, pp 3951–3960
22. Ren M, Triantafillou E, Ravi S, Snell J, Swersky K, Tenenbaum JB, Larochelle H, Zemel RS (2018) Meta-learning for semi-supervised few-shot classification. ICLR
23. Chen Z, Fu Y, Wang Y-X, Ma L, Liu W, Hebert M (2019) Image deformation meta-networks for one-shot learning. In: *Proceeding CVPR*, pp 8680–8689
24. Douze M, Szlam A, Hariharan B, Jégou H (2018) Low-shot learning with large-scale diffusion. In: *Proceeding CVPR*, pp 3349–3358
25. Pfister T, Charles J, Zisserman A (2014) Domain-adaptive discriminative one-shot learning of gestures. In: *Proceeding ECCV*. Springer, pp 814–829
26. Wang Y, Xu C, Liu C, Zhang L, Fu Y (2020) Instance credibility inference for few-shot learning. In: *Proceeding CVPR*, pp 12836–12845
27. Hu Y, Gripon V, Pateux S (2020) Leveraging the feature distribution in transfer-based few-shot learning. *Int Conf Artif Neural Netw*
28. Dhillon GS, Chaudhari P, Ravichandran A, Soatto S (2020) A baseline for few-shot image classification. ICLR
29. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: *Proceeding CVPR*. IEEE, pp 951–958
30. Wang X, Ye Y, Gupta A (2018) Zero-shot recognition via semantic embeddings and knowledge graphs. In: *Proceeding CVPR*, pp 6857–6866
31. Kampffmeyer M, Chen Y, Liang X, Wang H, Zhang Y, Xing EP (2019) Rethinking knowledge graph propagation for zero-shot learning. In: *Proceeding CVPR*, pp 11487–11496
32. Hayashi T, Fujita H, Hernandez-Matamoros A (2021) Less complexity one-class classification approach using construction error of convolutional image transformation network. *Inf Sci* 560:217–234
33. Hayashi T, Fujita H (2021) Cluster-based zero-shot learning for multivariate data. *J Ambient Intell Humanized Comput* 12(2):1897–1911
34. Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, Mardziel P, Hu X (2020) Score-CAM: score-weighted visual explanations for convolutional neural networks. In: *Proceeding CVPR workshops*, pp 24–25
35. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: *Proceeding ICCV*, pp 618–626
36. Chattopadhyay A, Sarkar A, Howlader P, Balasubramanian VN (2018) Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: *Proceeding WACV*, pp 839–847
37. Schulz K, Sixt L, Tombari F, Landgraf T (2020) Restricting the flow: information bottlenecks for attribution. In: ICLR
38. Li L, Wang B, Verma M, Nakashima Y, Kawasaki R, Nagahara H (2021) SCOUTER: slot attention-based classifier for explainable image recognition. *Proc ICCV*
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *NeurIPS*, pp 5998–6008
40. Hsu PY, Chen CT, Chou C, Huang SH (2022) Explainable mutual fund recommendation system developed based on knowledge graph embeddings. *Appl Intell*
41. Sun J, Lapuschkin S, Samek W, Zhao Y, Cheung N-M, Binder A (2020) Explain and improve: cross-domain few-shot-learning using explanations. arXiv:2007.08790
42. Geng Y, Chen J, Ye Z, Zhang W, Chen H (2020) Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs. *SWJ*
43. Karlinsky L, Shtok J, Alfassy A, Lichtenstein M, Harary S, Schwartz E, Doveh S, Sattigeri P, Feris R, Bronstein A et al (2021) StarNet: towards weakly supervised few-shot detection and explainable few-shot classification. *Proc AAAI*
44. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one* 10(7):0130140
45. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *Proceeding NeurIPS*, pp 5998–6008
46. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: *Proceeding CVPR*, pp 1492–1500
47. Locatello F, Weissenborn D, Unterthiner T, Mahendran A, Heigold G, Uszkoreit J, Dosovitskiy A, Kipf T (2020) Object-centric learning with slot attention. *Proc neurIPS*
48. Gidaris S, Komodakis N (2018) Dynamic few-shot visual learning without forgetting. In: *Proceeding CVPR*, pp 4367–4375
49. Bertinetto L, Henriques JF, Torr PH, Vedaldi A (2019) Meta-learning with differentiable closed-form solvers. ICLR
50. Krizhevsky A (2009) Learning Multiple Layers of Features From Tiny Images. University of Toronto, Master's thesis
51. Zagoruyko S, Komodakis N (2016) Wide residual networks. arXiv:1605.07146
52. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceeding CVPR*, pp 770–778
53. Zhuang J, Tang T, Ding Y, Tatikonda S, Dvornik N, Papademetris X, Duncan J (2020) Adabelief optimizer: adapting stepsizes by the belief in observed gradients. *Proc neurIPS*
54. Grant E, Finn C, Levine S, Darrell T, Griffiths T (2018) Recasting gradient-based meta-learning as hierarchical bayes. ICLR
55. Finn C, Xu K, Levine S (2018) Probabilistic model-agnostic meta-learning. In: *Proceeding NeurIPS*, pp 9516–9527
56. Widhianingsih TDA, Kang D-K (2021) Augmented domain agreement for adaptable meta-learner on few-shot classification. *Appl Intell*:1–17
57. Li L, Jin W, Huang Y (2021) Few-shot contrastive learning for image classification and its application to insulator identification. *Appl Intell*:1–16

58. Munkhdalai T, Trischler A (2018) Metalearning with hebbian fast weights. arXiv:1807.05076
59. Qiao S, Liu C, Shen W, Yuille AL (2018) Few-shot image recognition by predicting parameters from activations. In: Proceeding CVPR, pp 7229–7238
60. Rusu AA, Rao D, Sygnowski J, Vinyals O, Pascanu R, Osindero S, Hadsell R (2019) Meta-learning with latent embedding optimization. ICLR
61. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. IJCV 115(3):211–252. <https://doi.org/10.1007/s11263-015-0816-y>
62. Petsiuk V, Das A, Saenko K (2018) Rise: randomized input sampling for explanation of black-box models. BMVC
63. Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: ICML, pp 3145–3153
64. Wang H, Naidu R, Michael J, Kundu SS (2020) SS-CAM: smoothed score-CAM for sharper visual feature localization. arXiv:2006.14255
65. Kumar N, Sukavanam N (2019) Keyframes and shot boundaries: the attributes of scene segmentation and classification. In: Harmony search and nature inspired optimization algorithms, pp 771–782
66. Caelles S, Maninis K-K, Pont-Tuset J, Leal-Taixé L, Cremers D, Van Gool L (2017) One-shot video object segmentation. In: Proceeding CVPR, pp 221–230
67. Yang C, Rangarajan A, Ranka S (2018) Visual explanations from deep 3d convolutional neural networks for alzheimer's disease classification. In: AMIA annual symposium proceedings. American medical informatics association, vol 2018, p 1571
68. Wang X, Jiang L, Li L, Xu M, Deng X, Dai L, Xu X, Li T, Guo Y, Wang Z (2021) Etal: joint learning of 3d lesion segmentation and classification for explainable covid-19 diagnosis. IEEE Trans Med Imaging 40(9):2463–2476

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



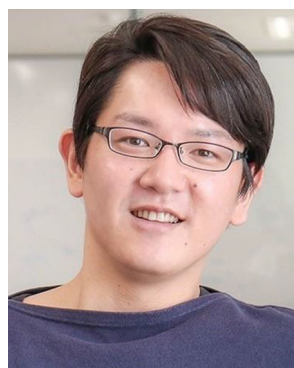
**Bowen Wang** was born in China and received his B.CS in computer science from Anhui University, China. He received his M.M. of medical information research at Osaka University. He is now a Ph.D. candidate at Osaka University Institute for Datability Science. His research interest includes computer vision and medical AI research. He has received the best paper award from APAMI 2020.



**Liangzhi Li** is a researcher with Institute for Datability Science, Osaka University, Japan. He received the B.Sc and M.Eng degrees in Computer Science from South China University of Technology (SCUT), China, in 2012 and 2016, respectively, and Ph.D. degree in Engineering from Muroran Institute of Technology, Japan, in 2019. His main fields of research interest include computer vision, deep learning, and medical imaging. He has received the best paper award from FCST 2017 and IEEE Sapporo Section (2018).



**Manisha Verma** received the BSc degree from Rajasthan University, India, in 2009, and the MSc and PhD degrees in mathematics from the Indian Institute of Technology Roorkee, India, in 2012 and 2016, respectively. She is currently a researcher at the Institute for Datability Science, Osaka University. From 2017 to 2018, she was a postdoc with the Indian Institute of Technology Gandhinagar, India. Her research interests include computer vision and deep learning.



**Yuta Nakashima** received the B.E. and M.E. degrees in communication engineering and the Ph.D. degree in engineering from Osaka University, Osaka, Japan, in 2006, 2008, and 2012, respectively. From 2012 to 2016, he was an Assistant Professor at the Nara Institute of Science and Technology. He is currently an Associate Professor at the Institute for Datability Science, Osaka University. He was a Visiting Scholar at the University of North Carolina at Charlotte in 2012 and at Carnegie Mellon University from 2015 to 2016. His research interests include computer vision and machine learning and their applications. His main research includes video content analysis using machine learning approaches. Prof. Nakashima is a member of ACM, IEICE, and IPSJ.



**Ryo Kawasaki** is a Clinician-Scientist in Ophthalmology at Osaka University Hospital. He has a strong passion aiming to achieve the ultimate goal of blindness prevention using epidemiology, data science, behavioral science and AI. He is jointly appointed as an adjunct professor at Southern Denmark University for medical data science in Ophthalmology. He has published more than 200 peer-reviewed papers, and serves as an editorial board member of Ophthalmic Epidemiology and Japanese Journal of Ophthalmology.

thalmic Epidemiology and Japanese Journal of Ophthalmology.



**Hajime Nagahara** received the Ph.D. degree in system engineering from Osaka University, Suita, Japan, in 2001. Since 2017, he is a Professor with the Institute for Data-bility Science, Osaka University. He was a Research Associate with the Japan Society for the Promotion of Science from 2001 to 2003. He was an Assistant Professor with the Graduate School of Engineering Science, Osaka University, from 2003 to 2010. He was an Associate Professor with the

Faculty of Information Science and Electrical Engineering, Kyushu University, from 2010 to 2017. He was a Visiting Associate Professor with CREA University of Picardie Jules Verns, in 2005. He was a Visiting Researcher with Columbia University in 2007–2008 and 2016–2017. His research interests include computational photography and computer vision. He was the recipient of an ACM VRST2003 Honorable Mention Award in 2003, IPSJ Nagao Special Researcher Award in 2012, ICCP2016 Best Paper Runners-up, and SSII Takagi Award in 2016.