Towards Better Representation and Interpretability for Deep Neural Networks on Visual Tasks

Submitted to
Graduate School of Information Science and Technology
Osaka University

January, 2023

Bowen WANG

Abstract

In recent years, Deep Neural Networks (DNNs) have shown their power over many research fields, and related applications are entering people's daily lives with unstoppable momentum. However, the large number of DNNs' training parameters causes difficulty in learning representation from real-world data efficiently, and the black-box nature harms its explainability. To accommodate actual demands, a DNN model should be adaptive (able to learn meaningful representation for different tasks) and trustworthy (interpretable of its decision). In this thesis, we will show how to design a DNN for better representation, as well as interpret its behavior for reliable artificial intelligence (AI). We first show the effect of designing DNNs according to target task. A well-designed DNN is adaptive to the target task, owning the ability to learn better representation. We also try to unveil the decision made by DNNs through explainable AI (XAI). By embedding a slot-attention-based XAI module, we find that a DNN model is interpretable, and the learning of representation can be benefited from this interpretability. XAI methods are further extended to find representation in a simple classification task. The found representation is transferred as training data for a complex object detection task, realizing weak supervision. In three different real-world scenarios, we evaluate that our proposal can encourage DNNs to learn better representation and let them be interpretable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep Neural Networks (DNNs), methods of machine learning (ML), were introduced to bring ML closer to its original goal - artificial intelligence (AI). DNNs have achieved many outstanding results in data mining [2, 3], machine translation [4, 5], natural language processing [6, 7], multimedia learning [8, 9], recommendation [10, 11], and other related fields. They have enabled machines to mimic human perceptions, such as seeing and hearing, solving many comprehensive pattern recognition challenges and achieving significant advances in AI-related applications.

To deal with more challenging tasks and obtain higher recognition performance, the architecture of DNNs is becoming deeper and more complex. However, such evolution inevitably brings two main issues that harm practical applications: i) DNNs may lack the ability to learn meaningful representation for target task [12]. ii) The decision made by DNNs is not explainable due to its black-box nature [13].

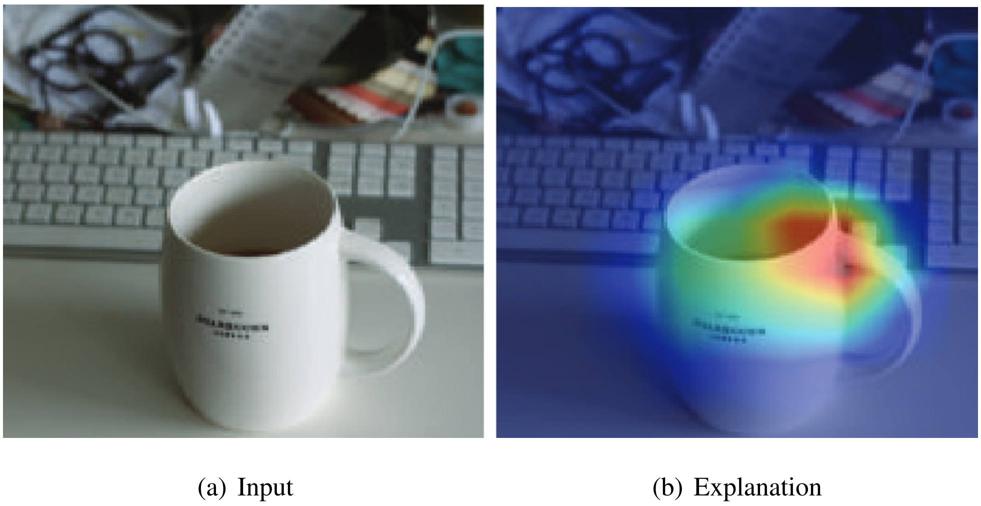
Learning better representation. The effectiveness of ML methods depends on learning representation from data [14]. Representation learning has been concerned with how to learn meaningful and superior representation, simplifying the complex raw data, removing invalid or redundant information from the raw data, and refining the valid information to form features for downstream tasks [12, 14].

Due to their powerful learning ability, DNNs are frequently used in recognition tasks for different types of data (e.g., text, image, video). However, as the depth of the model increases, DNNs become increasingly greedy [15, 16]; that is, a large quantity of data and annotations are needed for successful training. In the real world, there are many scenes (e.g., medical applications) for which there is not enough annotated data, and the cost of acquiring annotation is very high. Thus, the design (architectures and training patterns) for the DNNs to learn better representation is essential.

It is difficult for a plain DNN to learn meaningful representations for real needs. In order to obtain good representation, the design of DNNs requires careful consideration of the demands according to different tasks [14]. For example, a categorical loss should be adopted for multi-label image classification, and the spatial-temporal property of video should be introduced into DNNs' architecture for video recognition tasks. The popular direction of designing DNNs involves shared factors across tasks (e.g., multi-task [17], transfer learning [18, 19], domain adaption [20]), hierarchical organization of explanatory factors (e.g., hierarchical clustering [21]), spatial-temporal coherence (e.g., action recognition [22]), etc. Designing task-specialized DNNs can significantly improve their ability to learn representation and thus enables ML tasks to be much easier and more accurate.

Interpreting DNNs. In the early phase of ML, people used handcraft representation for the downstream task. It is easy to explain the decision of the model while the performance is not satisfying [23]. The excellent performance of DNNs stems from their huge trainable parameters [24,25]. Existing DNNs models often involve millions, even billions of parameters. However, even though researchers train a model that works well, they do not understand its reasoning. When DNNs are applied to sensitive areas closely related to human survival or safety (e.g., autonomous driving, finance, healthcare), it is crucial to explain their decision behind the black-box nature. For example, a highly accurate model that can only predict the possibility of a person having the disease is of little meaning to doctors – because they do not know how the prediction was derived.

Explainable artificial intelligence (XAI) [26, 27] has been gaining attention in the past few years. It can give a close look into the models' inference process. The most popular paradigm in



(a) Input (b) Explanation

Figure 1.1: Explanation for the classification of Coffee Mug using GradCAM [1].

XAI is to produce *a posteriori* explanations [1,28], providing a relevance map (attention map) that highlights the regions that affect the prediction. In Figure 1.1, we show the explanation for a DNN model recognizing "Coffee Mug" using the XAI method GradCAM [1]. It can be observed that the attention map covers the part of the cup handle. One drawback of these methods is that they can only provide such an explanation after the model training, lacking the ability to unveil DNNs in the forward calculation. Intrinsic methods [29,30] provide another way to interpret the behavior of DNNs. They build an interpretable learning module (e.g., attention mechanism [31]) inside the DNNs model and thus spontaneously explain how the model works.

A lot of work contributes to learning representations [12,14] or trying to interpret the behavior of DNNs [26,27]. However, the combination of these two factors is ignored. In this thesis, we try to unexplored the boundary research fields between representation learning and XAI. Our concept is built step by step. We start with work towards learning better representation and showing the impact of designing DNNs according to the target task. We then introduce the thought of XAI into the design of DNNs for a real-world scenario, which proves the possibility of combining representation learning and XAI. We further adopt XAI method as the tool to find useful representation that could be transferred between different tasks. It is an application of our concept.

In our first work, we show that a task-specialized DNN can contribute to learning better representation. Our DNN is designed to discriminate the multi-label, multi-task, and hierarchy properties for real-world data. We thus constructed a multi-task DNN model and adopted the hierarchy loss to regularize training. The learned representation shows an apparent hierarchical clustering consistent with human perception, and the retrieval performance is improved. These results prove that designing DNNs according to the target task can improve their ability to learn meaningful representation.

Next, we try to unveil the black-box nature of DNNs and bring this demand into the Few-shot Learning (FSL) task. FSL refers to learning from a small number of labeled samples, and the core is to learn the representation efficiently. Inspired by intrinsic XAI, we designed a self-explainable attention [31] module consisting of trainable patterns. The patterns are trained to learn meaningful representations from base classes (classes with adequate labeled images) and then transferred to accommodate novel classes (classes with few labeled images). Patterns can relieve the data requirement of DNNs for novel classes, and the learned representation from patterns can be interpreted by showing its attention, thus enabling a transparent FSL pipeline. On some FSL benchmarks, our method achieves better classification accuracy and interpretability.

Besides interpreting the DNNs, XAI can also help to realize the weakly-supervised nodule detection in Chest X-rays (CXR). We use XAI methods to find representations in a supervised nodule classification task and use them as the training data for a more challenging object detection task without annotations. The transfer of representation between tasks realizes weak supervision. In a medical imaging task of Chest X-ray (CXR) diagnosis, we show our method's potential to reduce the annotation burden for specialists.

The remainder of this thesis is organized as follows. Chapter 2 introduces the related studies of image retrieval, few-shot learning, XAI, and DNNs for medical imaging (e.g., CXR). Next, Chapter 3 details our multi-task hierarchy-aware model for a real-world ethnological museum application. In Chapter 4, we show the combination of FSL and XAI, enabling better representation and interpretability. We also demonstrate the potential of XAI technology realizing the weak supervision of nodule detection on CXR in Chapter 5. Chapter 6 discusses the combination of all three topics. Finally, Chapter 7 concludes the whole thesis.

Chapter 2

Related Work

Representation learning is one of the core problems of ML and has drawn attention through the past few decades [32, 33]. There are two types of methods for obtaining representation: hand-crafted and learning approaches. Popular hand-crafted methods include Scale-invariant Feature Transform (SIFT) [34] based on directional gradients reflecting texture characteristics, Histogram of Oriented Gradients (HOG) [35] describing the distribution of directional gradients, Shape Context [36] reflecting contour shapes, etc. Hand-crafted representations are comprehensible to humans for their transparent calculation. However, they do not work well on complex or large-scale visual tasks [23]. Due to the powerful ability to learn representation, the DNNs-based learning approaches have become the mainstream methods in recent years. This ability comes from the huge training parameter of DNNs while causing them to be greedy and unexplainable. Towards real-world applications, a DNN model requires well-designing and interpretability.

In this chapter, we will briefly introduce the works concerning our thesis. In Section 2.1, we go through the hashing-based retrieval and few-shot learning as the cases for representation learning. In Section 2.2, we introduce the paradigm of XAI. In Section 2.3, we summarize our contributions.

2.1 Representation Learning

Representation learning is a basic area that covers a large range of research fields. In order to demonstrate our theory, we take image retrieval (hashing-based), few-shot learning, and weak-supervision as our cases. We thus give a brief introduction to them.

2.1.1 Hashing-based Image Retrieval

Many hashing-based methods have been designed for approximate nearest-neighbor (ANN) search [37,38] in Hamming space for image retrieval. The target of these methods is to discriminate the similarity between images. They map high-dimensional images into compact binary codes with a preset number of bits, which can greatly reduce the calculation consumption and storage space. At the early stage, hashing-based methods focused on data independence [39], such as locality-sensitive hashing (LSH) and its variance [39,40]. The major drawback of LSH is that long code is necessary to achieve satisfactory search accuracy, which limits its application.

Recently, deep hashing methods [41–49] achieve great results in image hashing. Convolutional Neural Networks (CNNs) are adopted as feature extractors, and hash layers [43] are applied to generate hash codes. In Figure 2.1, we show the model structure of a classic method, HashNet [43]. Images are first fed into the CNNs backbone to extract features and then computed by the hash layer for hash code. Pairwise-similarity loss or triplet ranking loss are used to guide the training of CNNs. The target of this training is to maximize the Hamming distances between hash codes of dissimilar images (from different classes, 0) and minimize the Hamming distances of similar images (from the same class, 1). HashNet also adopts the Weighted Maximum Likelihood (WML) estimation to alleviate the severe data imbalance by adding weights in loss functions.

HashNet ignores the multi-label properties of some datasets, e.g., COCO [50]. One image could belong to multiple classes, however, works [41,43,45] like HashNet directly make image pairs similar if they share at least one class. IDHN [46] applied hard and soft similarity loss

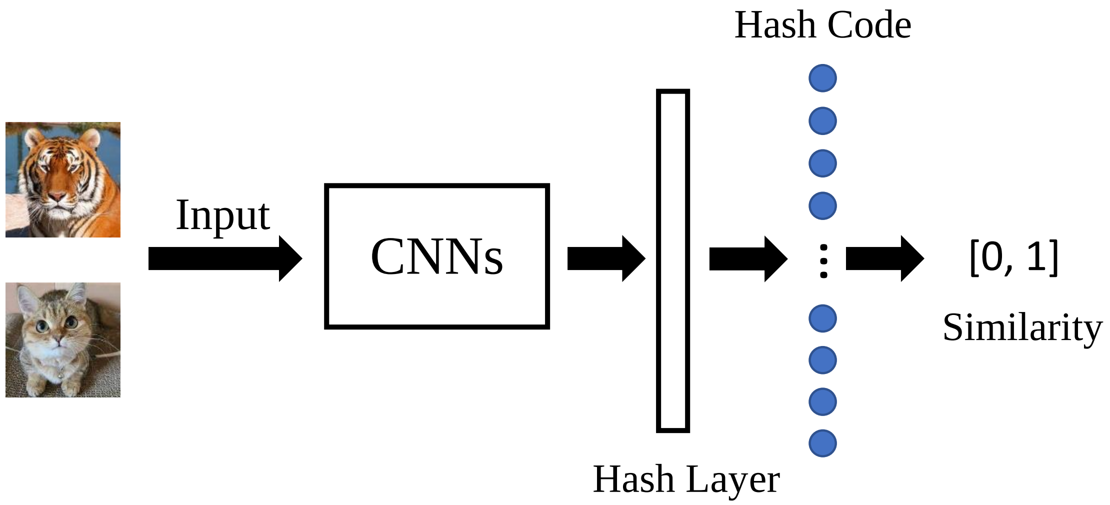


Figure 2.1: The pipeline of HashNet.

to solve the multi-label problem. They calculate the pairwise similarity between image pairs' labels by cosine similarity. This fine-grained pairwise loss can more effectively encode the image information from multi-label.

Existing research usually implements their experiments in some well-organized dataset, e.g., ImageNet [51], NUS WIDE [52], and MS COCO [50]. The images usually own simple labeling and cannot represent real needs. Some works [48, 49] try to solve the retrieval task for a real-world scene. CSA-Net [48] solves the task of searching for complementary products of fashion products. The model can better understand the relationship between products in the entire outfit by learning the representation of features in different subspaces. It is a retrieval task for fashion clothes, and their subspaces belong to similar semantic domains. For realistic applications, an image may have multiple labels belonging to different domains. Thus, the exploration of real-world demands needs further studies.

2.1.2 Few-shot Learning

Recently, due to the availability of a sufficient number of annotated images, DNNs 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 [53, 54], may not inherently have enough training samples. We require a new paradigm that allows training a model with a small number of labeled images, and FSL is proposed to solve those tasks.

Popular FSL model [55–57] serves as a testbed for certain aspects of such small tasks. In Figure 2.2, we demonstrate the pipeline of a classic FSL method, Matching Network [55]. The support set (shown on the left of the figure) contains annotated images from different classes, and the query image is the input we want to predict. g_θ and f_θ are DNN models (g_θ and f_θ share parameters) to extract feature vectors. Matching Network will calculate the distance between the query image to each of the support images via their feature vectors and thus know the class that the query image belongs to. The training is implemented in the base set (all classes own enough training data) by simulating an FSL scene. The testing is implemented on the novel set (a few training samples for each class). Notice that the base set and novel set are from similar domains. Recent efforts toward FSL are summarized as follows.

Many works focus on transforming images into vectors in an embedding space, where the distance between a pair of vectors represents the conceptual similarity. A Siamese network [58] uses a shared feature extractor to produce image embeddings for both support and query images. The weighted ℓ_1 distance is used for the classification criterion. Metric learning [55, 56] can offer a better way to map data into the embedding space, and some works try to improve the discriminatory power of image embeddings. Simple Shot [59] applies an ℓ_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 [60–64].

Another major approach of FSL is to optimize models. A simple way is to fine-tune the feature extractor using support images of novel classes [65]. However, due to very few support

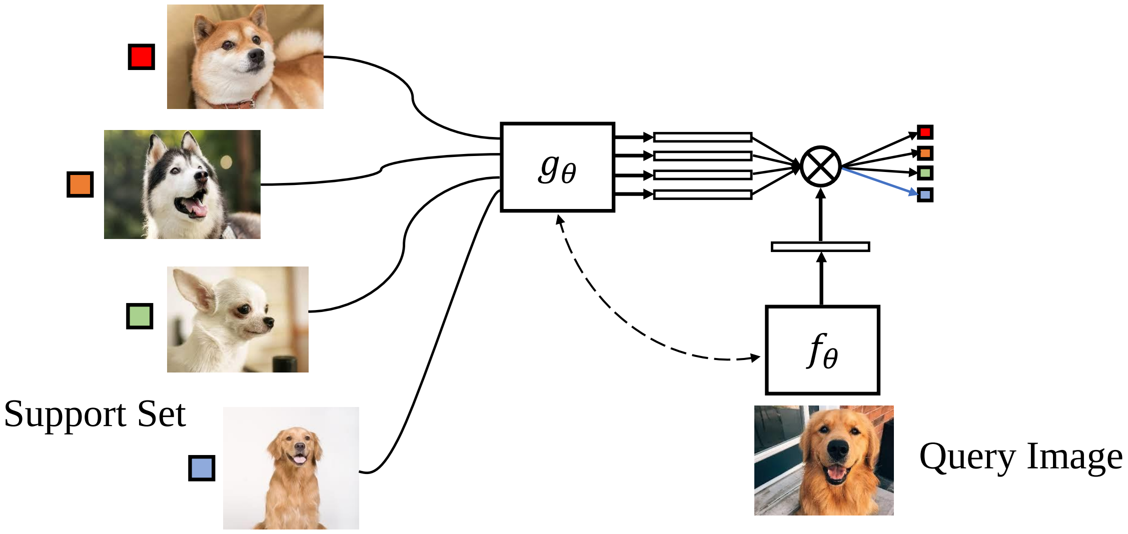


Figure 2.2: The pipeline of Matching Network.

samples, overfitting limited the training success. MAML [66] and its extensions [67,68] try to find the best initial parameters, and through one or more gradient adjustment steps, they can be easily adapted to a target task with only a small amount of data. Besides training good initial parameters, Meta-SGD [69] also trains the update direction and step size.

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 [70–72]. There are also some works that try to use samples that are weakly labeled or unlabeled [73,74]. ICI [75] introduces a judgment mechanism to enhance the training set by utilizing unlabeled data with confidently predicted labels.

Transductive or semi-supervised approaches [76,77] 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.

2.1.3 Weakly Supervised Object Localization

Weakly Supervised Object Localization (WSOL) aims to localize objects with image-level supervision. Most existing works [78, 79] mainly rely on XAI methods (e.g., CAM) derived from a classification model. They use such attention areas to localize the foreground object that concerns the classification results. However, they are often implemented for benchmark datasets, e.g., ImageNet [80], lacking the exploration of real-world application.

In this thesis, inspired by the thought of WSOL, we use XAI to find the important region for the classification of nodules for chest X-rays. We use the CXR report as auxiliary information to filter the representation (attention areas) found by XAI. Different from previous works solely based on XAI methods, our localization is constrained to be meaningful. The filtered representation is accurate and close to human annotation. We thus used them for the training of a more complex object detection task.

2.2 Explainable AI

DNNs are considered black-box technology, and XAI is a series of attempts to unveil them. There are mainly three XAI paradigms [81], *post-hoc*, *intrinsic*, and *distillation*. The post-hoc paradigm usually provides a heat map highlighting important regions for the decision (e.g., [1, 82]). The heat map is computed beside the forward path of the model. The intrinsic paradigm explores the important piece of information within the forward path of the model, e.g., as attention maps (e.g., [29, 83–86]). *Distillation methods* are built upon model distillation [87]. The basic idea is to use an inherently transparent model to mimic the behaviors of a black-box model (e.g., [88, 89]).

The post-hoc paradigm has been extensively studied among XAI methods. The most popular type of method is based on back-propagation or perturbation. Back-propagation methods include CAM [90], GradCAM [1], DeepLIFT [91], and their extensions [92–96]. Popular perturbation methods include RISE [28], meaningful perturbations [97], real-time saliency [98], extremal perturbations [99], I-GOS [100], IBA [82], etc. These methods basically give *attribu-*

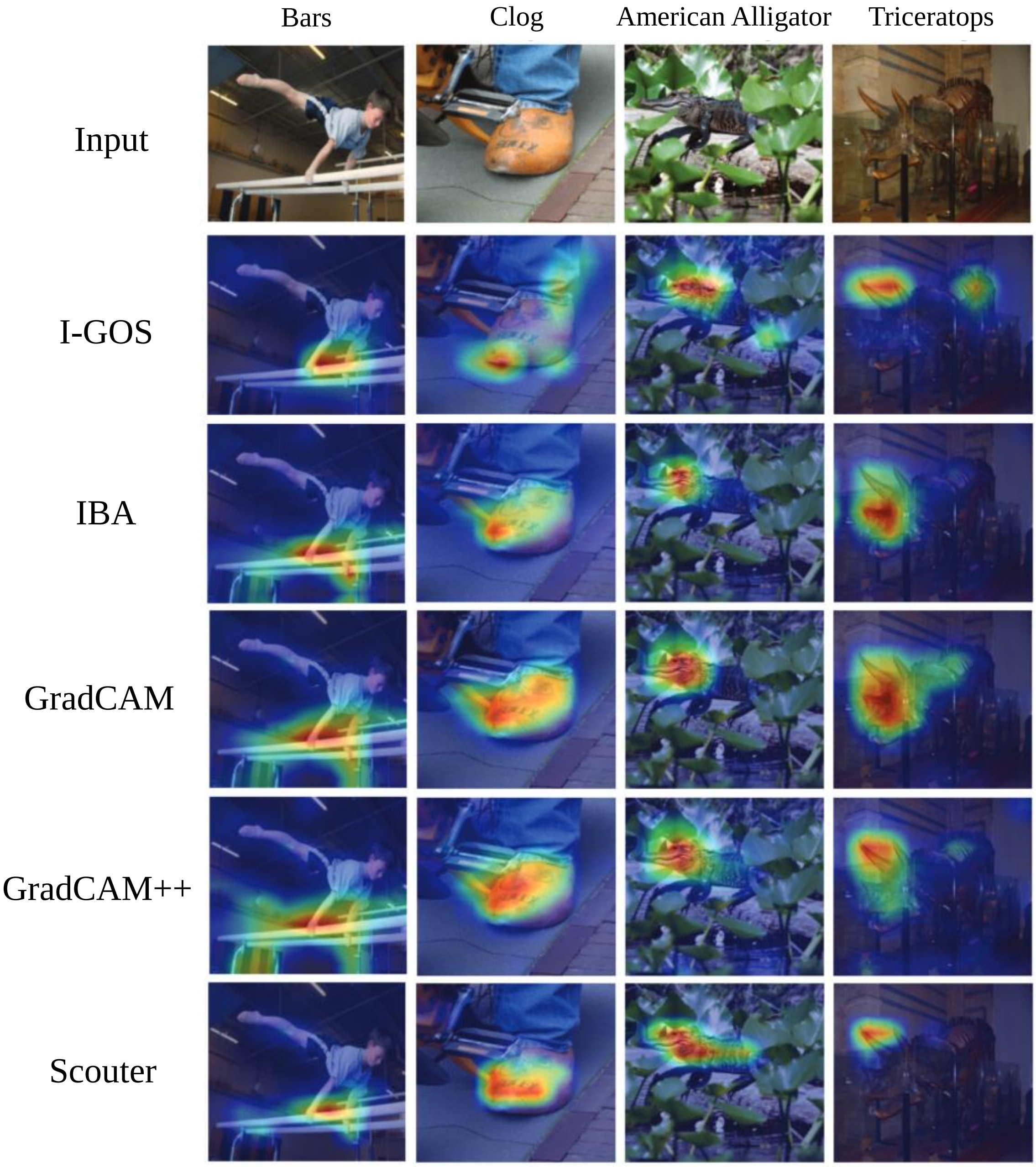


Figure 2.3: Visualization of XAI methods.

tive explanation, which visualizes support (decision relevant) regions of learned patterns for each category l in the set of all possible categories $\mathcal{L}$. This visualization can be done by finding regions in feature maps or the input image that have a large impact on the score for predicting ground truth.

A new type of *intrinsic* XAI, coined Scouter [29], has been proposed, which applies a slot-attention mechanism [101] to the classifier. This method can extract the attention for each class during training, which makes classification results explainable. In Figure 2.3, we demonstrate the visualization of some XAI methods (two post-hoc methods GradCAM [1] and GradCAM++ [92], two perturbation methods I-GOS [100] and IBA [82], an intrinsic method Scouter [29]) interpreting the prediction of class “Bars,” “Clog” “American Alligator,” and “Triceratops” from ImageNet [80]. We can observe that the explanation of different methods is similar, while the intrinsic method Scouter seems to give a more accurate attention area.

XAI methods have been widely applied to many deep learning tasks [102]. However, a few works [30, 103–105] have tried XAI for FSL tasks. Geng *et al.* [104] uses a knowledge graph to make an explanation for zero-shot tasks. Sun *et al.* [103] adopt layer-wise relevance propagation (LRP) [106] to explain the output of a classifier. StarNet [105] 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 [55]). Most of them are not evaluated on FSL benchmark datasets, which makes these methods not comparable. Thus, an FSL model which has both high classification accuracy and interpretability is important.

2.3 Our Contributions

Previous works either focus on better representation improving the performance of DNN for the target task, or just trying to interpret the existing DNN model for reliable AI. The combination of these two factors is ignored. In this thesis, we specify the design of DNNs for the target task to enhance their learning ability and interpret the decision of DNNs. Our contributions are summarized as follows:

1. We present that designing DNNs according to the target task can enhance their ability to learn representation. Our model’s superiority is proved in real-world data derived from

an actual database provided by an ethnological museum.

2. An XAI method is proposed for unveiling the black-box nature of DNNs. Our model enables a transparent FSL pipeline, as well as efficient knowledge transfer, improving classification accuracy.

3. We realize weak supervision of nodule detection in CXR via XAI. The object detection model is trained using the annotation generated from a simple classification task. Our method is of potential to relieve the annotation burden of specialists.

Chapter 3

Multi-task Hierarchy-aware Deep Hashing for Image Retrieval

3.1 Overview

Deep hashing has been widely used to approximate nearest-neighbor searches for image retrieval tasks. Most of them are trained with image-label pairs without any inter-label relationship, which may not make full use of real-world data. This research presents deep hashing, named HA^{2}SH [107], that leverages multiple types of labels with hierarchical structures that an ethnological museum assigns to its artifacts. Museums have large image databases to record their collections of artifacts. For example, the British Museum made their database with 1.9M images available online[1]. Each artifact (or equivalent image) often comes with rich metadata, including codes encoding taxonomic classification, as shown in Figure 3.1. Such an image database can facilitate the experience in the museum by, e.g., providing a handy and easy-to-use app to retrieve relevant artifacts (or images) by taking an image of the artifact in an exhibition, allowing exploration of relevant artifacts on the artifact for visitors.

In order to implement such an app, a powerful and efficient approach to image retrieval is

[1] `https://www.britishmuseum.org/collection`

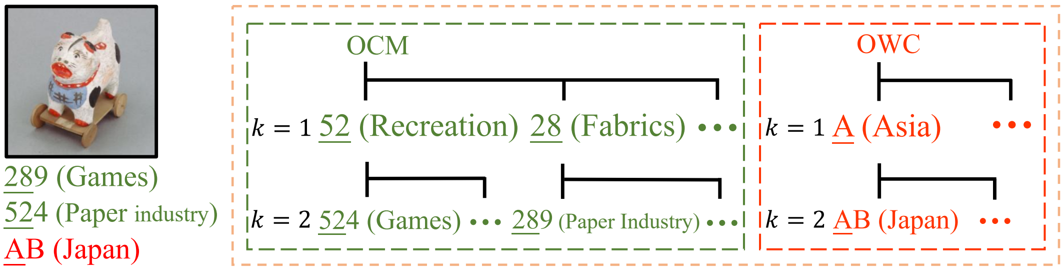


Figure 3.1: An example of artifact (or image) in our dataset. It comes with OCM and OWC codes to roughly represent the functionalities of the artifact and where it originates. k is the hierarchical level of the labels.

necessary. Because of its high computational and storage efficiency, hashing [39, 40, 108, 109] can be the possible choice. Deep hashing [41–48, 110–113] adopts deep convolutional neural networks (CNNs) [114, 115] as backbone networks to learn a nonlinear hash function. It allows large-scale retrieval of images [41, 116] and videos [117–120].

Previous works often use existing datasets, such as ImageNet and COCO [49–52], to train the image retrieval models; however, this may not be very coherent with actual image retrieval scenario for, e.g., the museum uses. In real-world data, an artifact (or an image) can have multiple taxonomic classifications to describe different aspects of the artifact. For example, the artifact in Figure 3.1 originated from Japan, was used as a toy, and was made of paper, which is encoded into multiple classification codes. Furthermore, such codes can also encode taxonomic hierarchy, e.g., Japan is in Asia. This inherently leads to the multi-task, multi-label, and hierarchical nature of this image retrieval task.

We present a new approach for hierarchy-aware hashing, called HA^{2}SH, which can handle real-world data derived from an actual database provided by an ethnological museum. There are two label spaces associated with each artifact, where one of them can have multiple labels and both of them have hierarchical structures. In order to generate hashes dedicated to the two label spaces, our model uses a shared CNN followed by two branches with a respective hash layer to generate hashes. Multi-task learning with respective losses is adopted for better image representation, whereas the losses take the multiple labels and their hierarchy into account. Our three main contributions are as follows: 1) We propose HA^{2}SH, which is trained in a multi-task and multi-label paradigm for hierarchy-aware hashing. 2) We design a flexible retrieval system that allows controlling the importance of different hashes to meet actual users' needs. 3) We evaluate HA^{2}SH with a real-world dataset derived from an actual database provided by an ethnological museum.

3.2 Our Dataset

Under our collaborative project with an ethnological museum, we were granted access to the database of its collection of artifacts, which contains images and metadata of each artifact. We extracted these images and associated metadata to build our dataset, containing 450,443 images (127,337 artifacts) in total. The metadata includes various information on the artifact, and we used as a label the outline of cultural materials (OCM) and the outline of world cultures (OWC) defined by Human Relations Area Files[2], where OCM and OWC roughly describe the function and culture of the artifact.

One important aspect of OCM and OWC is their hierarchical structures. The semantics is encoded in a few-digit code, representing a certain category and its subcategory; for example, OCM's three-digit label `524` stands for "game", where the first two digits `52` means recreation. OWC label `AB06` stands for the culture of "Ainu," where the first digit and the first two digits mean "Asia" and "area of Japan." We used only the first two digits of the OWC label to identify the region where the artifact originated. The first and second levels of OCM have 31 and 80 classes, whereas those of OWC has 8 and 50 classes. Each image has at least one OCM and OWC label; many images have multiple OCM labels, and an artifact can serve multiple functions. An example of data is shown in Figure 3.1.

[2] https://ehrafworldcultures.yale.edu/ehrafe/

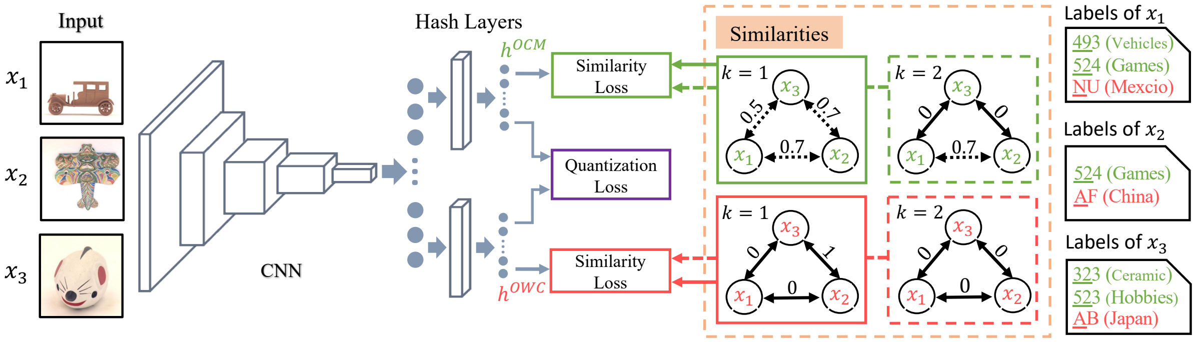


Figure 3.2: Overview of HA^{2}SH with OCM and OWC branches to generate respective hashes. The cosine similarity is used to define the hierarchical image-image similarities, which provide hard (solid lines) and soft (dotted lines) similarity losses.

3.3 Hierarchy-aware Hashing

Given the dataset above, we design a deep hash that takes into account the multi-task, multi-label, and hierarchical nature of our dataset for image retrieval.

3.3.1 Problem Formulation

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ be the set of images in our dataset, from which images similar to a query image are retrieved. HA^{2}SH finds a mapping from an image to a Q-bit binary code $\mathcal{B} = \{b_i\}_{i=1}^N$, where $b_i \in \{-1, 1\}^Q$. Code b_i is trained to be locality-sensitive, and its neighbors may be semantically similar to each other. Following the previous work, [43, 46, 47], instead of generating binary codes, we adopt continuous relaxation $h_i \in [-1, 1]^Q$, which can be easily mapped to a binary code by making the sign of each element of h_i; therefore, HA^{2}SH learns mapping f from x_i to h_i.

The semantic similarity between a pair of images is defined based on their labels. The labels for image $x_i \in \mathcal{X}$ can be represented by a multi-hot vector $z_i^{(u,k)}$, where $u \in \{\text{OCM}, \text{OWC}\}$ (for OWC, the vector is often reduced to a one-hot vector) and k is the level in the label hierarchy (k is either 1 or 2 for both OCM and OWC). We adopt the same strategy as IDHN [46] to define similarity $s_{ij}^{(u,k)}$ between images x_i and x_j with labels $z_i^{(u,k)}$ and $z_j^{(u,k)}$ using the cosine similarity, i.e.,

$$s_{ij}^{(u,k)} = z_i^{(u,k)} \cdot z_j^{(u,k)} / (\|z_i^{(u,k)}\| \; \|z_j^{(u,k)}\|), \tag{3.1}$$

where "·" is the operator for the inner product. This definition quantifies a fine-grained semantic similarity, taking the multi-label nature of our dataset by allowing similarity in-between 0 and 1. The similarities in the different levels are fused through the loss function to generate hierarchy-aware multi-level deep hashes. For notation simplicity, we omit u and k unless it is ambiguous.

Figure 3.2 shows the pipeline of our model. A CNN is used as the backbone for feature extraction. HA^{2}SH branches after the global average pooling to generate different hashes for OCM and OWC. Each branch has a hash layer, consisting of an FC layer and the $tanh(\cdot)$ nonlinearity, to generate hash $h_i^{(u)}$.

3.3.2 Learning from Similarities

For a pair of hashes h_i and h_j, we use the inner product $h_i \cdot h_j$ to measure the distance between them, which is proved to be a good alternative to the Hamming distance used for binary hashes to quantify the pairwise similarity [42, 46, 116]. We train our mapping f for label category u so that generated hashes h_i and h_j well encode our label-based similarity s_{ij} for image pair (x_i, x_j).

Hard similarity loss. Let $\mathcal{S}_1$ and $\mathcal{S}_0$ be the sets of image indices pairs (i, j) whose (multiple) labels are exactly the same (i.e., $s_{ij} = 1$) or completely different (i.e., $s_{ij} = 0$), respectively. Pairs in these sets give a strong signal that corresponding hashes h_i and h_j are close to or far from each other. To encode this, similar to HashNet [43], we define the probability of the similarity given a pair of hashes as

$$p(s_{ij} \mid h_i, h_j) = \begin{cases} \sigma(h_i \cdot h_j) & \text{for } (i,j) \in \mathcal{S}_1 \\ 1 - \sigma(h_i \cdot h_j) & \text{for } (i,j) \in \mathcal{S}_0 \end{cases} \tag{3.2}$$

where $\sigma(\cdot) \in [0, 1]$ is the sigmoid function. Generally, the number of image pairs with the same set of labels is far less than those of completely different sets of labels. We therefore introduce a weight w_{ij} that gives γ for $(i, j) \in \mathcal{S}_1$ and $1 - \gamma$ for $(i, j) \in \mathcal{S}_0$ to mitigate the imbalance and define the loss function as

$$\ell_{\text{H}} = - \sum_{(i,j) \in \mathcal{S}_1 \cup \mathcal{S}_0} w_{ij} \log p(s_{ij} \mid h_i, h_j). \tag{3.3}$$

Soft similarity loss. For pairs (i, j) that have partially matched sets of labels, we use the loss defined in IDHN [46]. Let $\mathcal{S}'$ denote the set of indices pair (i, j) such that $s_{ij} < 1$. The soft similarity loss is given by:

$$\ell_{\text{S}} = - \sum_{(i,j) \in \mathcal{S}'} \left(\frac{h_i \cdot h_j + Q}{2} - s_{ij} Q \right)^2. \tag{3.4}$$

This loss enforces the correlation between $h_i \cdot h_j$ and s_{ij} to take into account the multiple labels assigned to a single image. As we discussed in 3.3.1, the soft similarity can compute the distance between different samples in a continuous way, which has the advantage of multi-label than previous binary settings.

Quantization loss. We use $tanh(\cdot)$ to squash the output of the hash layer to be in $[-1, 1]$, but this does not guarantee that the resulting hash has values closer to either 1 or -1. We thus use the quantization loss, given by

$$\ell_{\text{Q}} = \sum_i \| |h_i| - \mathbf{1}_Q \|^2, \tag{3.5}$$

where $|h_i|$ gives the absolute value element-wise and $\mathbf{1}_Q$ is a vector with all its Q elements being 1.

Overall loss for hierarchical training (HT) Due to the hierarchical structure of our labels, the hard and soft similarity loss can be defined for respective levels of the hierarchies. Therefore, the loss for branch u is given by combining the losses as:

$$L^{(u)} = \sum_k \ell_{\text{H}}^{(u,k)} + \delta \sum_k \ell_{\text{S}}^{(u,k)} + \lambda \ell_{\text{Q}} \tag{3.6}$$

where δ and λ are weights to control the soft similarity and quantization losses, respectively. The model is trained in the multi-task learning framework, in which the following loss is used to train mappings f^{OCM} and f^{OWC}:

$$L = \sum_{u} L^{(u)} = L^{\mathrm{OCM}} + L^{\mathrm{OWC}}. \tag{3.7}$$

3.3.3 Retrieval

Given query image q, we retrieve similar images in $\mathcal{X}$, for which we preliminary compute the set $\mathcal{H}^{(u)} = \{h_i^{(u)}\}_{i=1}^{N}$ of hashes. The pairwise distance between q and x_i can be given by

$$D_i^{(u)}(q) = f^{(u)}(q) \cdot h_i^{(u)}. \tag{3.8}$$

We combine the distances for OCM and OWC with weight $\alpha \in [0, 1]$ to provide flexible retrieval that can take both aspects of images into account:

$$D = \alpha D^{\mathrm{OCM}} + (1 - \alpha) D^{\mathrm{OWC}}. \tag{3.9}$$

Images in $\mathcal{X}$ are ranked according to D.

3.4 Experiments

We implemented HA^{2}SH with PyTorch, using a CNN backbone ResNet-50 [115] pre-trained on the ImageNet classification task [114]. The model was trained for 30 epochs with AdamW [121], which started with a learning rate 10^{-4}, decreased by a factor of 10 at the 20-th epoch. The learning rate of hash layers is set to be 5 times greater than the backbone network. Based on the data statistic, we set γ and δ to 0.9 and 1. λ is set to 0.1. For learning the hierarchical structures in the labels, we only used the first-level ($k = 1$) for the first 10 epochs and then added the second-level's loss.

For evaluation, 2% of the images are randomly picked out as query images for evaluation and the rest are used as image database $\mathcal{X}$. We randomly sampled 50% of the image database for training. We use the mean Average Precision (mAP) for evaluating our model.

Table 3.1: Similarity learning experiments with 32-bits and 64-bits of hash codes. Evaluated with mAP@1000.

Branch			OCM		OWC	
OCM	OWC	Soft Loss	32-bit	64 bit	32 bit	64-bit
	✓		0.233	0.256	0.585	0.625
	✓	✓	0.227	0.254	0.588	0.623
✓			0.629	0.701	0.300	0.342
✓		✓	0.644	0.713	0.319	0.334
✓	✓		0.645	0.711	0.600	0.632
✓	✓	✓	0.652	**0.726**	0.597	**0.635**

3.4.1 Effects of Multi-task and Multi-label Losses

To evaluate the collective effect of two branches (OCM and OWC) and multiple labels for 32-bit and 64-bit hashes, we used only second-level ($k = 2$) labels with removing some losses (the losses for OCM and OWC branches; and the soft similarity loss). As shown in Table 3.1, the performance by multi-task losses with the soft similarity loss was better than those of individual tasks'. Interestingly, the model trained only for OCM labels can still give relevant images for OWC and vice versa. This implies the correlation between OCM and OWC labels. The soft similarity loss worked well for the OCM labels, while in the OWC space, there are not many multi-label cases, and this loss serves slightly. In Table 3.2, we compare our method to previous works on OCM and OWC, respectively. The results show our superiority in retrieval correct answers.

3.4.2 Effects of Hierarchy Awareness

The hard/soft similarity losses encourage images with the same label to form a cluster in the hash space. Our hierarchy-aware hashing is for learning a better hash space, which forces the

Table 3.2: Performance comparison to previous works. Evaluated with mAP@1000.

	OCM		OWC	
Methods	32-bit	64-bit	32-bit	64-bit
HashNet [43]	0.626	0.688	0.581	0.622
IDHN [46]	0.633	0.710	0.587	0.625
CSQ [47]	0.645	0.721	0.595	0.630
HA^{2}SH	**0.652**	**0.726**	**0.597**	**0.635**

Table 3.3: Performance of Hierarchy-aware hashing in both branches and hierarchical level k.

	OCM		OWC	
Label Setting	$k = 1$	$k = 2$	$k = 1$	$k = 2$
Only First-level	0.791	-	0.834	-
Only Second-level	0.754	0.726	0.769	0.634
All Levels	0.788	0.712	0.829	0.601

model to put images with semantically similar (i.e., the first level $k = 1$) classes closer to each other. This gives an extra value for image retrieval. We used UMAP [122] to visualize the 64-bit hashes trained with OCM in a 2-D space. We sampled some images (not multi-label) with the chosen classes that share the first-level labels. For example, labels `532` (*Representative art*) and `534` (*Musical instruments*) belong to the first-level class `53` (*Art*).

Figure 3.3 shows the visualizations. The classes belonging to the same first-level label are assigned with the same color scheme. When the model is only trained with second-level ($k = 2$) labels (left), the first-level labels appear to be randomly placed. The red color clusters of `532` and `534` are far from each other. However, `534`'s cluster is close to blue scheme clusters. With this hash space, retrieved images can be semantically irrelevant. With hierarchical training, the first-level labels bring images with similar semantic meanings closer, and then training with the

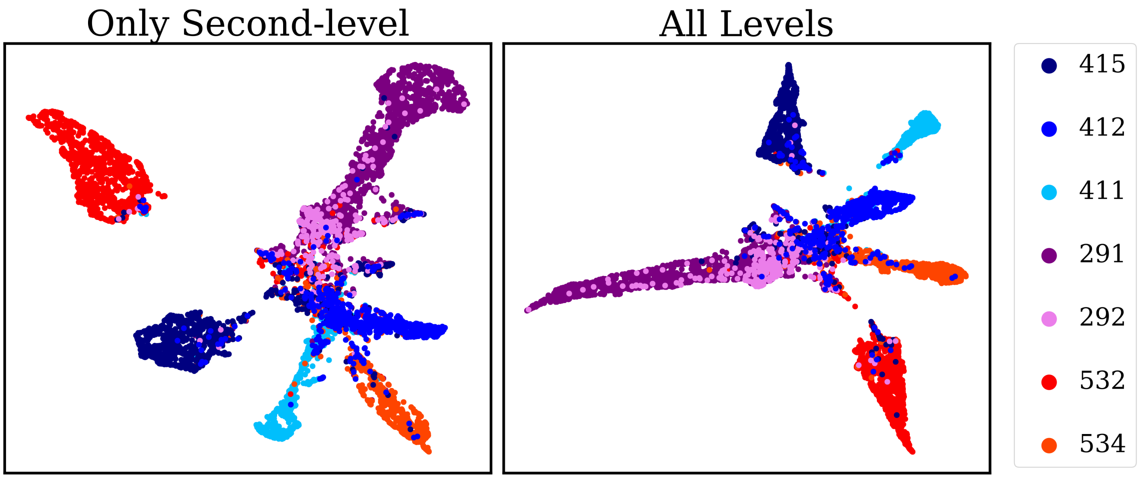


Figure 3.3: The visualization of UMAP for classes share similar semantic meanings

Table 3.4: The performance for different α values, evaluated with respect to OCM, OWC, and their union, in mAP@1000.

α	0	0.25	0.5	0.75	1
OCM	0.313	0.681	0.690	0.698	0.712
OWC	0.601	0.598	0.591	0.584	0.392
Union	0.216	0.559	0.544	0.551	0.342

second-level labels refines the clusters. The clustering of images in the hash space roughly takes this hierarchical structure into account, as shown in Figure 3.3 (right). The clusters of the same color scheme are aggregated.

Table 3.3 shows the performances of two models: one only trained with second-level labels ($k = 2$) and the other with all levels labels. Both models have two branches and use soft similarity loss. They are evaluated with OCM and OWC labels at both first ($k = 1$) and second ($k = 2$) levels. The results show that for both OCM and OWC, the performances improved for the first level with all levels training at the cost of the second-level performance.

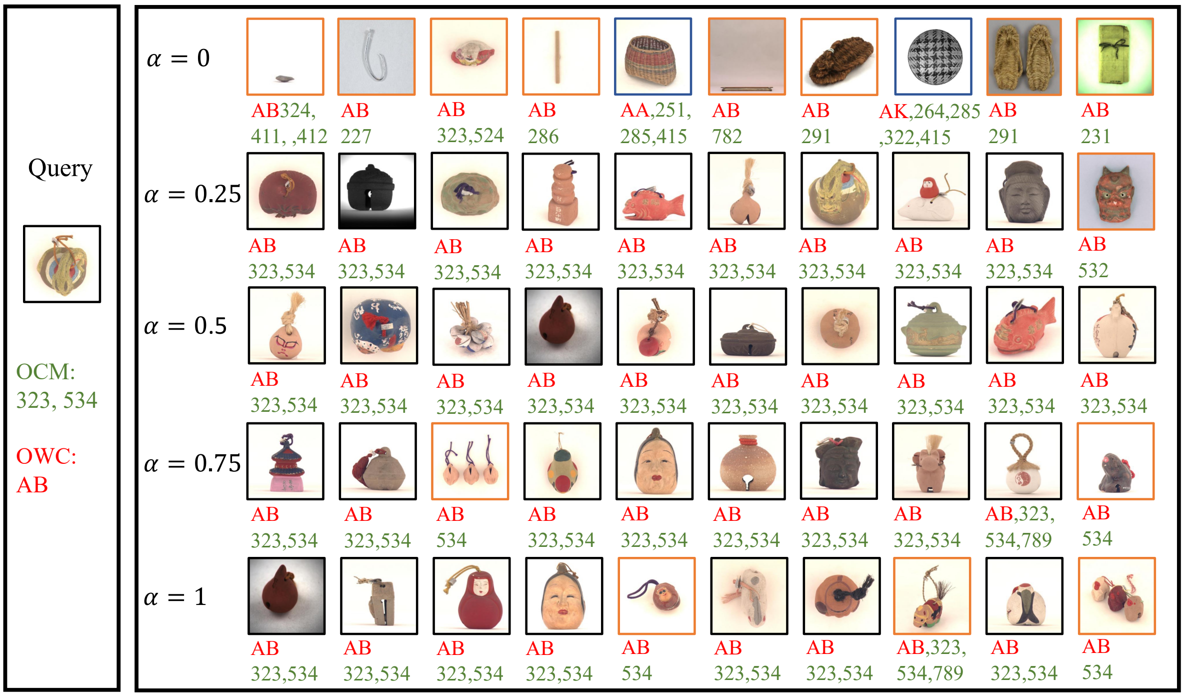


Figure 3.4: Demonstration of top 10 retrieval answers with different interest factors α.

3.4.3 Evaluation of Retrieval Performance

The hyper-parameter α controls the users' preference for OCM and OWC. We evaluated our model with different values of α. Table 3.4 summarizes the performances for different α, evaluated with respect to OCM, OWC, and their union (an image is counted as correct if both OCM and OWC labels are the same as a query), where the model is trained with the two branches, the soft similarity loss, and all levels labels. The results demonstrate that, as expected, $\alpha = 0$ gives a higher OWC performance, while $\alpha = 1$ gives a higher OCM performance. We can also observe that the best Union mAP is achieved by setting $\alpha = 0.25$. This may imply that some information on OCM has been encoded into the OWC branch. Although OCM and OWC are different tasks, there are some common concepts shared between them.

Figure 3.4 gives examples of retrieval results for different α. The query has OCM label *323* (*Ceramic technology*) and *534* (*Musical instruments*), as well as OWC label `AB` (*Japan*). Images marked with black boxes are with the exact same labels (of both OCM and OWC) as the query. The orange and blue boxes represent partial matches and complete mismatches, respectively. For $\alpha = 1$, all retrieved images are with OWC label `AB`. OCM labels are almost correct, which may imply a high correlation between OWC and OCM labels. Meanwhile, for $\alpha = 0$, which fully focuses on OWC, HA^{2}SH gave relatively diverse images. When $\alpha = 0.5$, all retrieved images are with the same labels as the query.

3.5 Summary

In this research, we proposed HA^{2}SH for image retrieval, targeted at the ethnological museum database. Our results demonstrated that HA^{2}SH could leverage multiple labels and their hierarchical structures to learn a better hash. It fuses hash codes learned from different types of labels to offer a flexible retrieval system. We believe HA^{2}SH provides a good user experience in museum apps. Our future work includes a subjective evaluation to show the usability of the retrieval system in some application scenarios.

Chapter 4

Visual Explainable Few-shot Learning

4.1 Overview

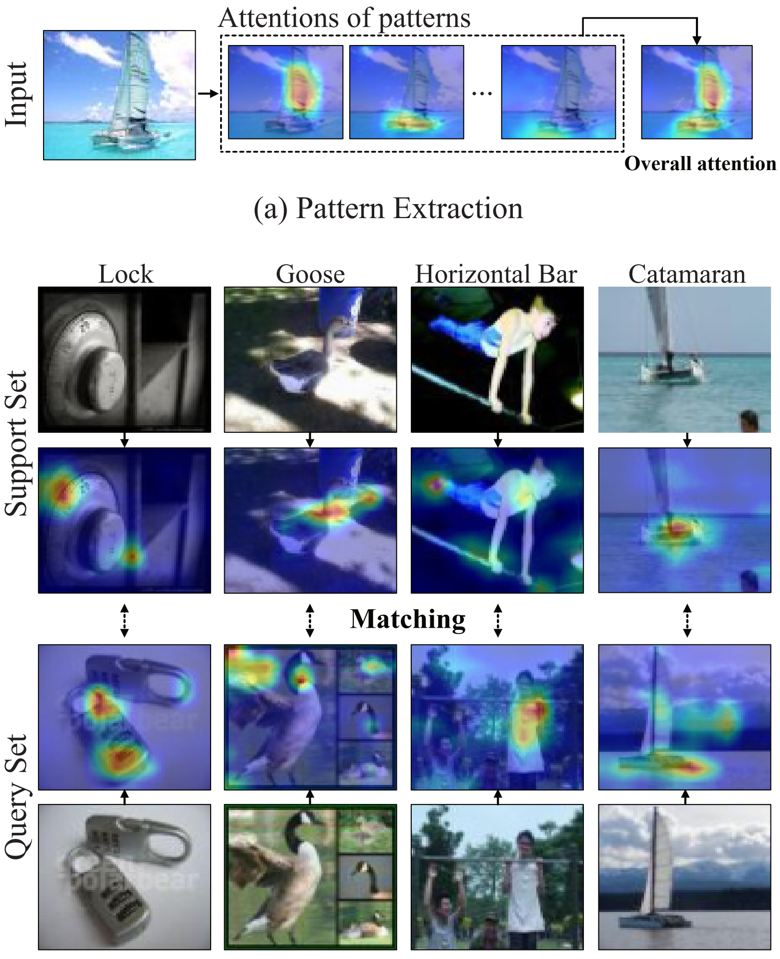
Few-shot learning (FSL) is of great significance for at least the following two scenarios [123]: First, FSL can relieve the heavy needs for data gathering and labeling, which can boost the ubiquitous use of deep learning techniques, especially for users without enough resources. Second, FSL is an important 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 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 pre-trained backbone convolutional neural networks (CNNs), which compute the features of an input image, may sometimes be useless when novel classes have significant visual differences from base class images [124]. 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.

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 to 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 [30, 125]), 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 Figure 4.1 (a). This directly means that we have a way of seeing 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 Figure 4.1 (b) and is used for matching. In Figure 4.1 (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 cat-



(a) Pattern Extraction

(b) Pairwise Matching with overall attention

Figure 4.1: Few-shot learning using pair-matching.

egory 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 relate the visualization and model decision further. The main contributions of our work include: 1) We propose a new explainable FSL model that achieves high classification accuracy, qualitatively and quantitatively showing its explainability. 2) 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. 3)

A PM mechanism that can relate the visual explanations with the model decision using matching scores, which may help find potential prediction failures. 4) Our method combines several techniques and concepts, e.g., FSL, attention, feature representation, and explainable AI, which can inspire future research.

4.2 Material and Methods

4.2.1 Problem Definition

This study addresses an inductive FSL task (*c.f.*, and a transductive task [76, 77]), in which we are given two disjointed sets $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{novel}}$ of samples. The former is a base set of many labeled base class images, whereas the latter is a novel set of a few labeled novel class images, where the disjointed sets of base and novel classes are denoted by $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$, respectively. The FSL task is to find a mapping from a novel image $x \in \mathcal{D}_{\text{novel}}$ to the corresponding class $y \in \mathcal{C}_{\text{novel}}$, with the images in $\mathcal{D}_{\text{base}}$ 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 $\mathcal{C}_{\text{base}}$ and serves as the basis for the 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 $\mathcal{D}_{\text{novel}}$.

4.2.2 Overview

The overall process is illustrated in Figure 4.2. 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_θ, where θ is the set of learnable parameters. F is then fed into the *pattern extractor* (PE) module, f_ϕ,

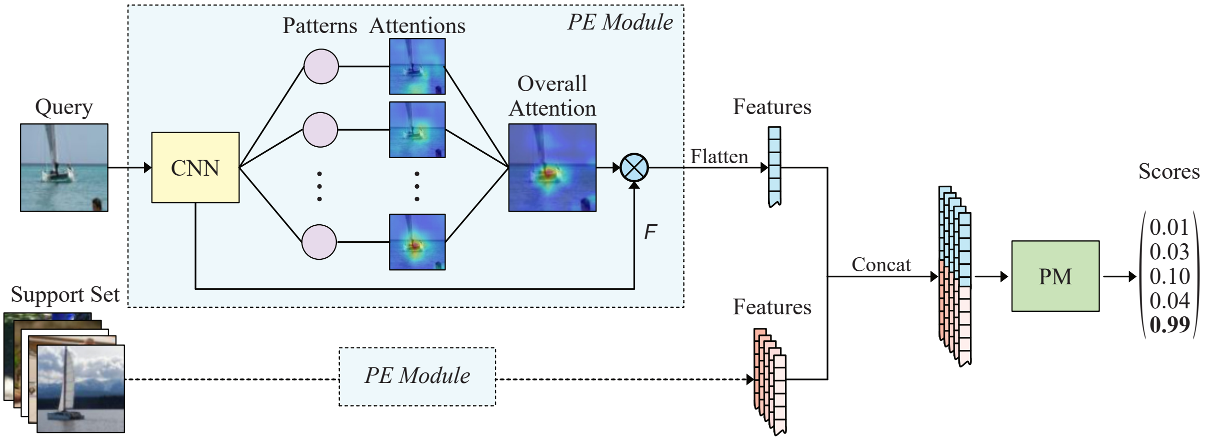


Figure 4.2: Overall structure of MTUNet. One query is processed by the CNN backbone and *pattern extractor* (PE) to provide exclusive patterns and then turned into overall attention. The query is concatenated to each support to make a pair for final discrimination through pairwise matching (PM). The dotted line represents each support image undergoing the same calculation as the query.

with learnable parameter set ϕ. 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 $\mathcal{S}$.

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 $\mathcal{D}_{\text{base}}$. Consequently, the patterns are more likely to be shared among $\mathcal{D}_{\text{novel}}$ given that $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{novel}}$ are from similar domains.

4.2.3 Pattern Extractor

Figure 4.3 shows the structure of our PE module. The input feature map F is first fed into a 1×1 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 [29, 126, 127] is added to the features, *i.e.*, $\tilde{F} = F' + P$.

The slot-attention [31] 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 $t = 1, 2, \ldots, T$ and $W^{(1)} = W$ is the learnable parameters. The nonlinear transformations for $W^{(t)}$ and $\tilde{F}$ are given by

$$g_{\mathrm{Q}}(W^{(t)}) \in \mathbb{R}^{z \times d}, \quad g_{\mathrm{K}}(\tilde{F}) \in \mathbb{R}^{d \times l}. \tag{4.1}$$

The attention is given using a normalization function ξ as

$$\bar{A}^{(t)} = g_{\mathrm{Q}}(W^{(t)}) g_{\mathrm{K}}(\tilde{F}) \tag{4.2}$$

$$A^{(t)} = \xi(\bar{A}^{(t)}) \quad \in (0, 1)^{z \times l}, \tag{4.3}$$

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

$$U^{(t)} = A^{(t)} F'^{\top} \tag{4.4}$$

$$W^{(t+1)} = \mathrm{GRU}\,(U^{(t)}, W^{(t)}). \tag{4.5}$$

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

$$A^{(t)} = \xi(\bar{A}^{(t)}) = \sigma(\bar{A}^{(t)}) \odot \mathrm{Softmax}_{\mathrm{R}}(\bar{A}^{(t)}), \tag{4.6}$$

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 Figure 4.1 (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.*,

$$A' = \frac{1}{z} A^{(T)} \mathbf{1}_z \tag{4.7}$$

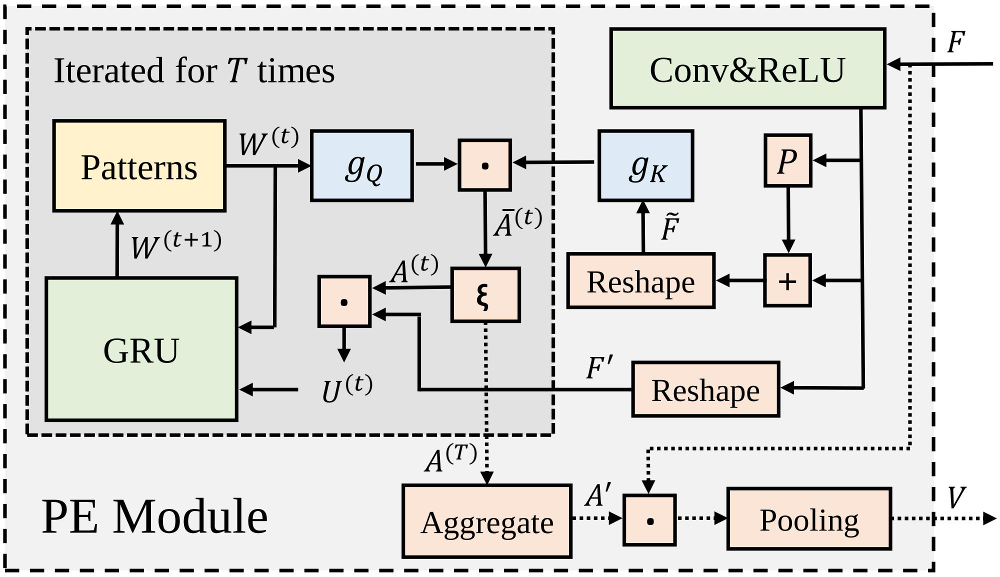


Figure 4.3: The structure of our pattern extractor module.

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

$$V = \frac{1}{hw}\sum_{ij} A'_{ij} F_{ij}, \tag{4.8}$$

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 $j = 1,2,\dots,w$).

4.2.4 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 [55, 56] to find matches between a query and the supports, and the cosine similarity or the ℓ_2 distance are typical choices [59, 128]. Learnable distances are another popular choice for metric learning-based FSL methods [60–62]. We use a learnable distance with an MLP (refer to Section 4.3.5).

Let V^{q} and $\{V_{kn}\}$ be features obtained by applying the PE to query image $x^{\mathrm{q}} \in \mathcal{Q}$ 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_γ with learnable parameters γ:

$$s(V^{\mathrm{q}}, \bar{V}_k) = \sigma(f_\gamma([V^{\mathrm{q}}, \bar{V}_k])), \tag{4.9}$$

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

$$k^* = \arg\max_k s(V^{\mathrm{q}}, \bar{V}_k). \tag{4.10}$$

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 [56].

4.2.5 Training

For training, we sample a set $\mathcal{Q} = \{(x^{\mathrm{q}}_{km}, y^{\mathrm{q}}_{km}) \mid i = 1, \dots, K \times M\}$ 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:

$$L = -\sum_{(x^{\mathrm{q}}, y^{\mathrm{q}}) \in \mathcal{Q}} \sum_{k=1}^{K} y^{\mathrm{q}}_k \log(\bar{s}(V^{\mathrm{q}}, \bar{V}_k)), \tag{4.11}$$

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

4.3 Experiments

4.3.1 Datesets

We evaluate our approach on three commonly-used datasets, mini-ImageNet [55], tiered-ImageNet [70], and CIFAR-FS [129]. **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 4.2.1. The images in miniImageNet are of size 84×84. As all recent work, we adopt the same splits of [55] **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 84×84. **CIFAR-FS** is a dataset with images sampled from CIFAR-100 [130]. This dataset contains 100 classes with 600 images each. We follow the split given in [129], which are 64, 16, and 20 classes for the base, novel validation, and novel test sets.

4.3.2 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_θ, which are often used for FSL tasks, namely Conv-4 [56], WRN-28-10 [131] and ResNet-18 [132]. For ResNet-18, we remove the first two down-sampling layers and change the kernel of the first 7×7 convolutional layer to 3×3. 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 [59]. We train the backbone CNNs with all images in $\mathcal{D}_{\text{base}}$. The performance of a simple nearest-neighbor-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. The corresponding number of classes' (a subset of $\mathcal{C}_{\text{base}}$) images are selected to pre-train the module as a normal classification task similar to [29]. The importance of this choice is discussed in Section 4.3.5. Both g_{Q} and g_{K} have three FC layers with ReLU nonlinearities between them. All the parameters in the backbone f_θ 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 [133] is adapted as an optimizer. Input images are resized to 80×80, and we applied data augmentation including random flip and affine transformations, following [59]. A GPU workstation with two NVIDIA Quadro GV100 (32GB memory) GPUs are 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 could easily reproduce our results.

4.3.3 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 for the backbone CNNs. Thus, we do not adopt any post-processing methods like ℓ_2 normalization in [59]. For testing the model, we report our best model on $\mathcal{D}_{\text{val}}$ by randomly sampling 10,000 1-shot and 5-shots tasks from $\mathcal{D}_{\text{test}}$ in Tables 1–3 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 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) mean the model trained without the PE module. This model has a structure similar to ProtoNet [56] and is used to evaluate the impact of 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 achieve high prediction accuracy for FSL tasks. We also find that the different architectures of the backbone CNNs affect the performance. With a 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. Asides 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 20×20 feature maps, while the ResNet-18 variants have 10×10. Such larger feature maps not only provide more information to the PM module but also give a better basis for 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 focuses on informative regions as it is designed to be. We will further analyze the importance of pattern number z and PE pre-training categories selection for training MTUNet in Section 4.3.5.

[†] Results are reported in [59]

[‡] Results are reported in [129]

Table 4.1: Average accuracy of 10000 episodes of 5-way tasks on the mini-ImageNet dataset test set.

Approach	**Backbone**	**One shot**	**Five shots**
SimpleShot (UN) [59]	Conv-4	33.17±0.17	63.25±0.17
MetaLSTM [67]	Conv-4	43.44±0.77	60.60±0.71
MatchingNet [55]	Conv-4	43.56±0.84	55.31±0.73
MAML [66]	Conv-4	48.70±1.84	63.11±0.92
ProtoNet [56]	Conv-4	49.42±0.78	68.20±0.66
GNN [61]	Conv-4	50.33±0.36	66.41±0.63
RelationNet [62]	Conv-4	50.44±0.82	65.32±0.70
Meta SGD [69]	Conv-4	50.47±1.87	64.03±0.94
RCNet [134]	Conv-4	**54.85±0.84**	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	**69.43±0.46**
MAML [66]‡	ResNet-18	49.61±0.92	65.72±0.77
R2-D2 [129]‡	ResNet-18	51.20±0.60	68.20±0.60
RelationNet [62]‡	ResNet-18	52.48±0.86	69.83±0.68
ProtoNet [56]‡	ResNet-18	54.16±0.82	73.68±0.65
Gidaris [128]	ResNet-15	55.45±0.89	70.13±0.68
SNAIL [57]	ResNet-15	55.71±0.99	68.88±0.92
adaCNN [135]	ResNet-18	56.88±0.62	71.94±0.57
SimpleShot (UN) [59]	ResNet-18	57.81±0.21	**80.43±0.15**
MTUNet (w/o PE)	ResNet-18	55.27±0.33	67.51±0.39
MTUNet	ResNet-18	**58.13±0.44**	75.02±0.43
SimpleShot (UN) [59]	WRN	57.26±0.21	78.99±0.14
Qiao [136]	WRN	59.60±0.41	73.74±0.19
MTUNet (w/o PE)	WRN	56.41±0.33	69.55±0.39
MTUNet	WRN	**60.12±0.45**	**79.23±0.42**

Table 4.2: Average accuracy of 10000 episodes of 5-way tasks on the tiered-ImageNet dataset test set.

Approach	**Backbone**	**One shot**	**Five shots**
Reptile [68]‡	Conv-4	48.97±0.21	66.47±0.21
SimpleShot (UN) [59]	Conv-4	51.02±0.20	68.98±0.18
MAML [66]	Conv-4	51.67±1.81	70.30±0.08
ProtoNet [56]‡	Conv-4	53.31±0.20	72.69±0.74
RelationNet [62]	Conv-4	54.48±0.93	71.32±0.78
AD2AML+IR [137]	Conv-4	54.97±1.92	-
RCNet [134]	Conv-4	58.42±0.96	**74.17±0.78**
MTUNet (w/o PE)	Conv-4	57.02±0.58	70.94±0.52
MTUNet	Conv-4	**59.12±0.61**	**73.31±0.65**
SimpleShot (UN) [59]	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	**63.83±0.53**	**82.07±0.46**
Meta SGD [69]‡	WRN	62.95±0.03	79.34±0.06
SimpleShot (UN) [59]	WRN	64.35±0.23	85.69±0.15
LEO [138]	WRN	66.33±0.05	81.44±0.09
MTUNet (w/o PE)	WRN	62.11±0.30	78.40±0.35
MTUNet	WRN	**66.52±0.48**	**86.17±0.41**

Table 4.3: Average accuracy of 10000 episodes of 5-way tasks on the CIFAR-FS dataset test set.

Approach	**Backbone**	**One shot**	**Five shots**
RelationNet [62]‡	Conv-4	55.00±1.00	69.30±0.80
ProtoNet [56]‡	Conv-4	55.50±0.70	72.00±0.60
MAML [66]‡	Conv-4	58.90±1.90	71.50±1.00
GNN [61]‡	Conv-4	61.90	75.30
R2-D2 [129]	Conv-4	65.30±0.20	**78.30±0.20**
MTUNet (w/o PE)	Conv-4	62.55±0.51	74.62±0.54
MTUNet	Conv-4	**65.81±0.65**	77.42±0.60
MTUNet (w/o PE)	ResNet-18	65.32±0.37	79.54±0.34
MTUNet	ResNet-18	**67.47±0.43**	**82.81±0.41**
MTUNet (w/o PE)	WRN	67.29±0.39	82.98±0.35
MTUNet	WRN	**70.49±0.46**	**86.55±0.44**

4.3.4 Explainability

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

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 Eq. (4.9), 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 mental 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 4.4 (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 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

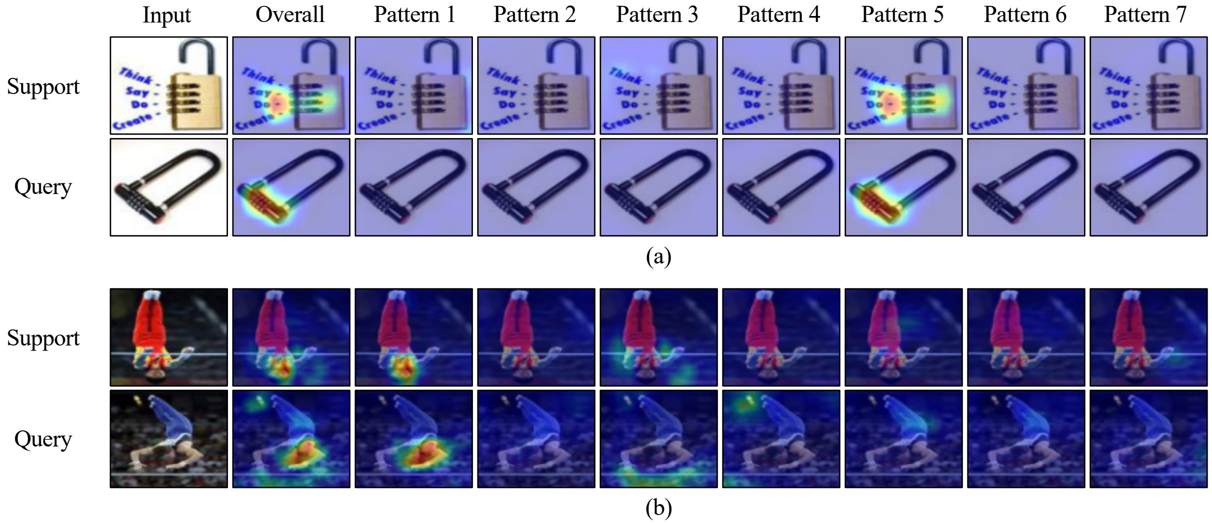


Figure 4.4: visualization of each pattern and the average features for a sampled task in the mini-ImageNet dataset. (a) is the lock class and (b) is the horizontal bar class. Overall is the overall attention among all patterns. The third to ninth columns are the visualization of the regions corresponding to the learned patterns.

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 appearances.

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

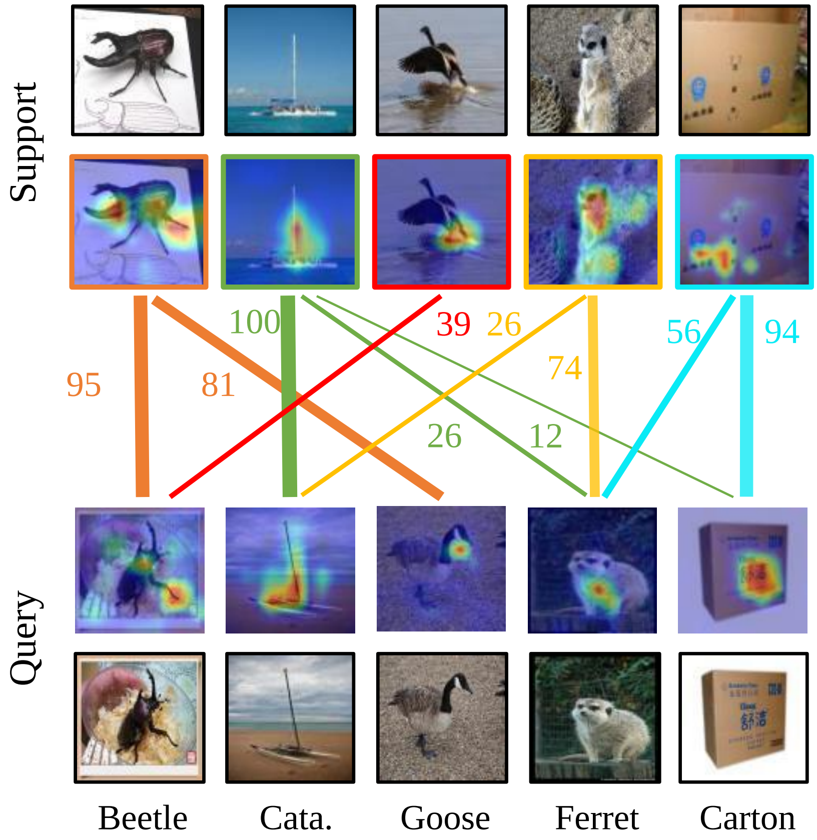


Figure 4.5: Matching point of one sampled task in the mini-ImageNet dataset. We only show the connection between pairs with a score over 0, and the scores are shown as percentages.

the unseen classes.

Visualization of pairwise matching scores Figure 4.5 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 colored lines shows a 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 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 of the relationship between visual patterns and the matching scores.

We also provide more visualization samples in Section 4.5.

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 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 [80]). 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

Table 4.4: Evaluation of MTUNet and existing XAI methods using explainability metrics.

		mini-ImageNet		
Methods	Type	Precision ↑	IAUC ↑	DAUC ↓
DeepLIFT [91]	Back-Prop	0.728	0.680	0.131
GradCAM [1]	Back-Prop	0.807	0.712	0.116
GradCAM++ [92]	Back-Prop	0.826	0.735	0.107
Score-CAM [94]	Back-Prop	0.811	0.702	0.110
SS-CAM [95]	Back-Prop	0.791	0.720	0.114
RISE [28]	Perturbation	0.757	0.753	0.098
IBA [82]	Perturbation	0.871	0.764	0.096
MTUNet	Intrinsic	**0.902**	**0.793**	**0.091**

image. (ii) Insertion area under the curve (IAUC) [28]: 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) [28]: This metric measures the accuracy drop when gradually removing important pixels from the input image. As shown in Table 4.4, 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.

4.3.5 Discussion

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 analyze 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 $\mathcal{C}_{\text{base}}$ (*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 $\mathcal{C}_{\text{base}}$.

The test accuracies are shown in Figure 4.6 for 5-way 1-shot and 5-way 5-shot tasks on 10,000 sampled episodes over $\mathcal{D}_{\text{test}}$ of the three datasets. The horizontal axis represents the number of patterns and the vertical axis represents the average accuracy. We can observe that for all settings a performance drop when only using one pattern. We would say that the performance has 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. Figure 4.4), 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 $\mathcal{C}_{\text{base}}$ 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 $\mathcal{C}_{\text{base}}$ 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 4.7 left shows a scatter plot of the validation accuracies and corresponding test ac-

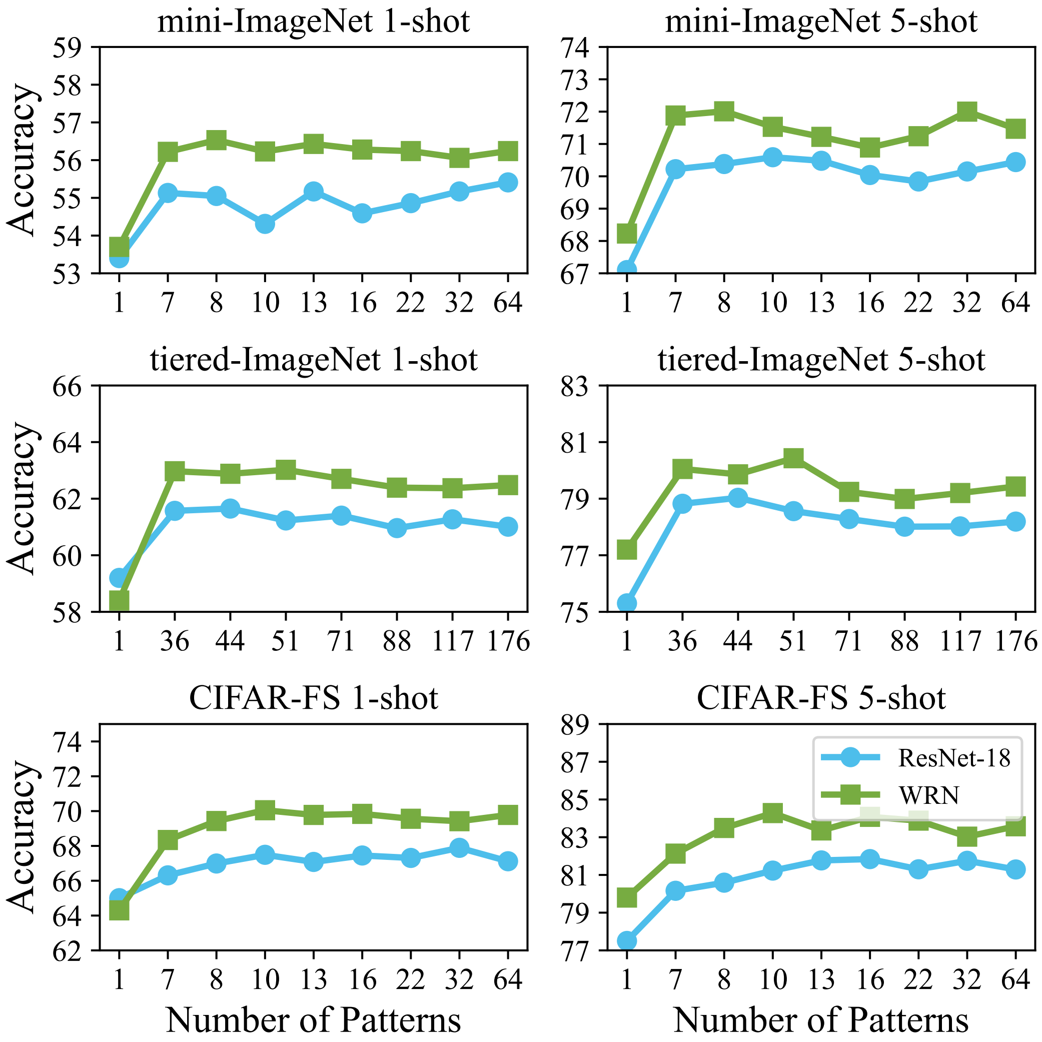


Figure 4.6: Results of pattern number settings for the mini-ImageNet, tiered-ImageNet, and CIFAR-FS dataset. The horizontal axis represents the number of patterns, and the vertical axis represents the average accuracy. We report the results with 10,000 sampled 5-way episodes in the novel test set.

curacies. 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

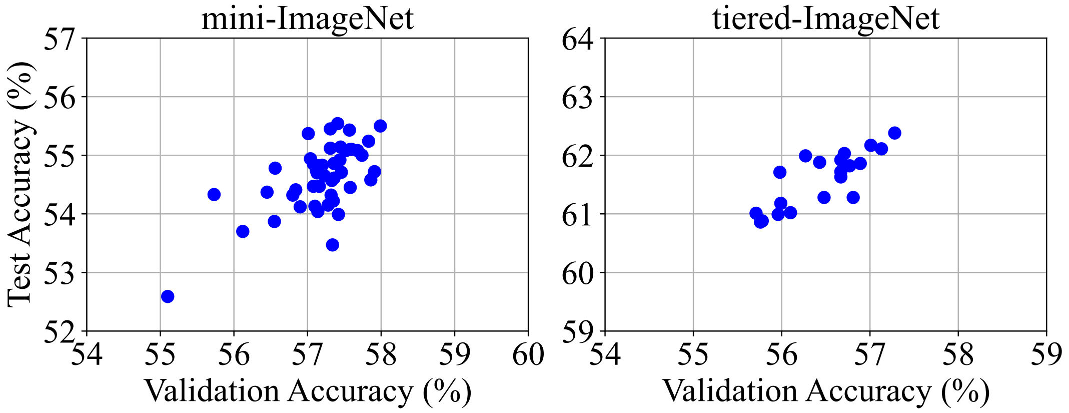


Figure 4.7: Performance of random classes sampling for PE pre-training of patterns. All experiments are implemented on the mini-ImageNet and tiered-ImageNet datasets using ResNet-18 as the backbone.

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 PE pre-training. However, we can use the validation set to find the best choice.

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 4.5, 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.

Table 4.5: Performance of different metric learning methods. All the experiments are implemented on the min-ImageNet dataset.

	Conv-4		ResNet-18		WRN	
Methods	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	**54.01±0.37**	**69.43±0.46**	**58.13±0.44**	**75.02±0.43**	**60.12±0.45**	**79.23±0.42**

4.4 Summary

In this chapter, 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 the CNNs backbone. It can learn better representations and is proven 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 combination, we can further manually analyze 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 been seen before. This can be advantageous since the explanation given by MTUNet can provide an intuitive interpretation (*intrinsic*) of what the model does.

4.5 Supplementary 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 (Figures 4.8, 4.10, 4.12) and the pairwise matching scores (Figures 4.9, 4.11, 4.13, row and column are consistent with the overall attention visualization for support and query of 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 Figure 4.9, we find there are two confusing categories of `lock` and `carton`. They all obtain a high score for each other category. The visualization in Figure 4.8 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 confusion.

Sample 2 As shown in Figure 4.11, the pairwise matching scores for this sample find proper matches except for `poncho`. In Figure 4.10, 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 color. 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 an explanation for FSL.

Sample 3 In Figure 4.12 and 4.13, 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.

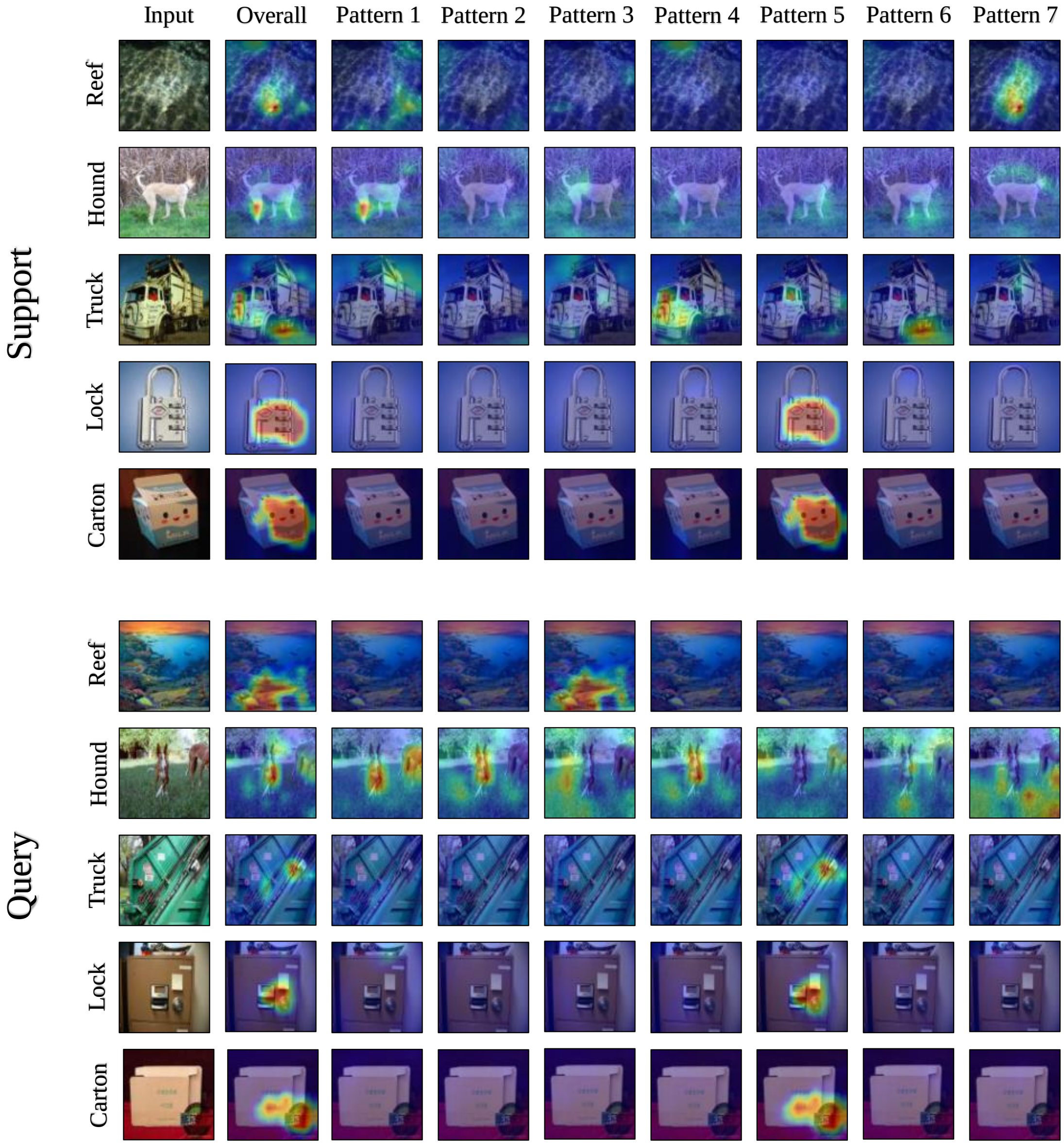
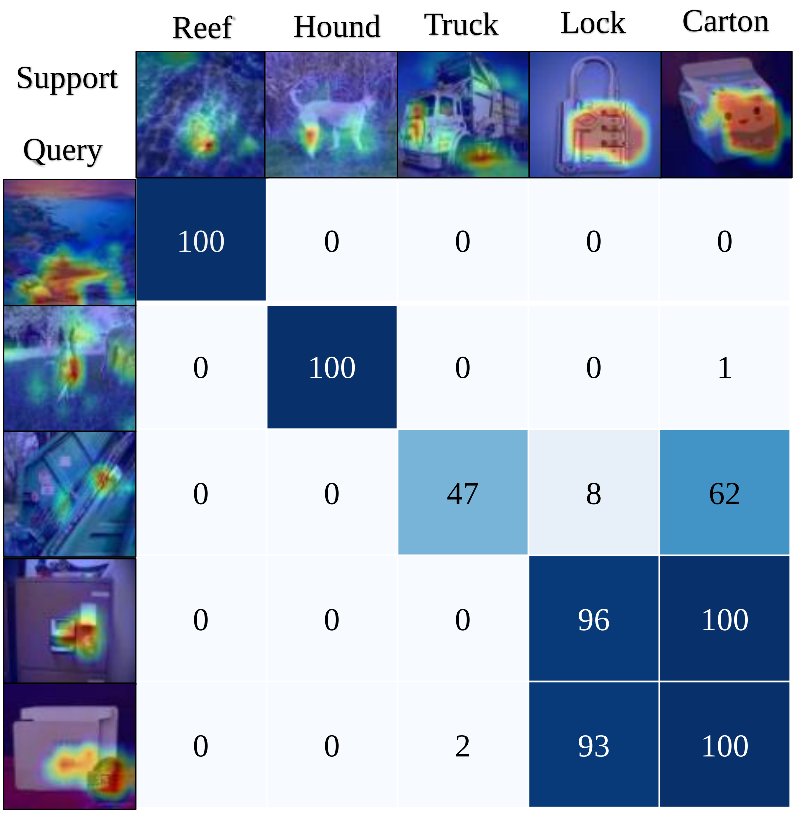


Figure 4.8: Pattern-based visualization of sample 1.



	Reef	Hound	Truck	Lock	Carton
	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

Figure 4.9: Pairwise matching of sample 1.

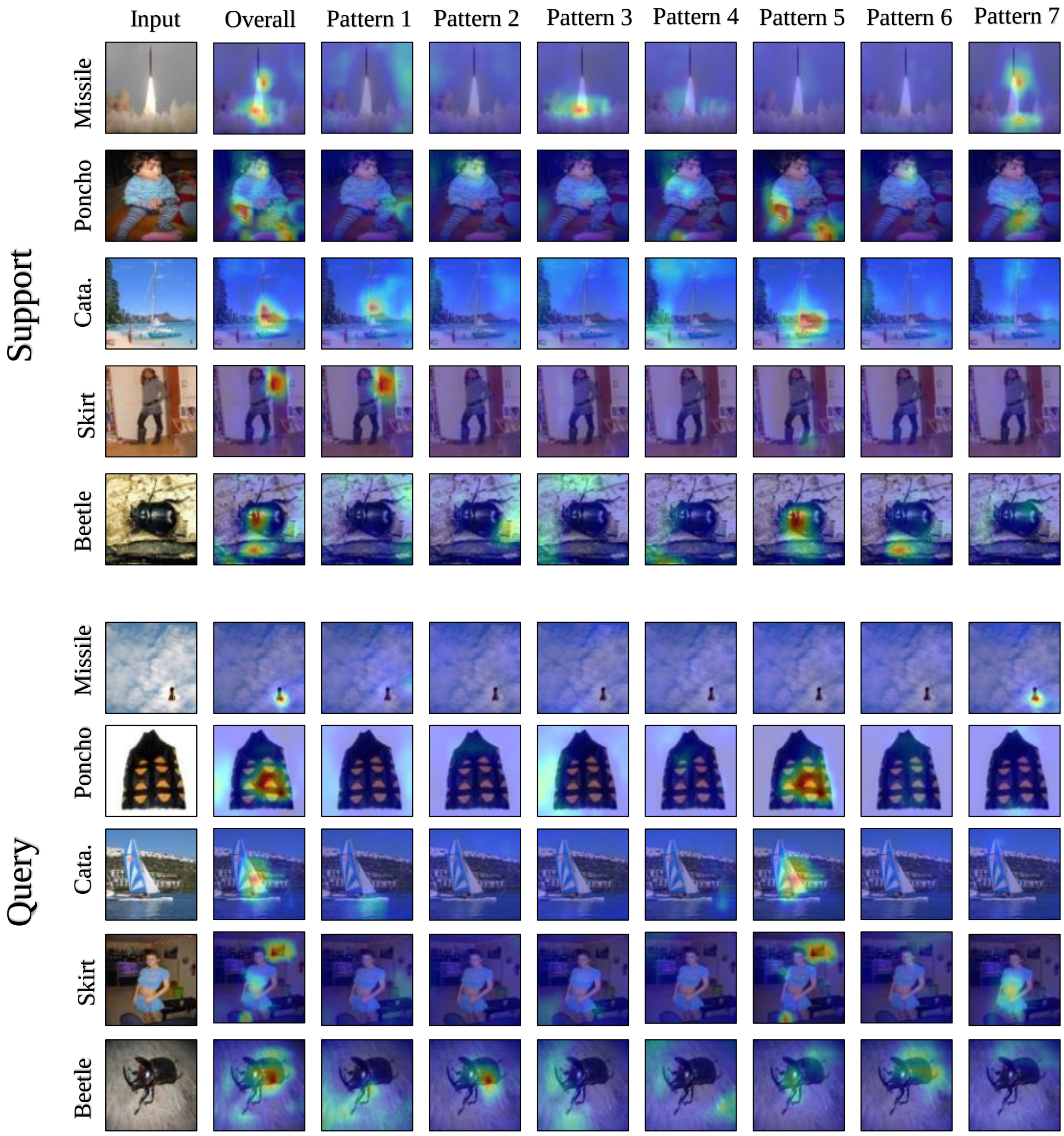
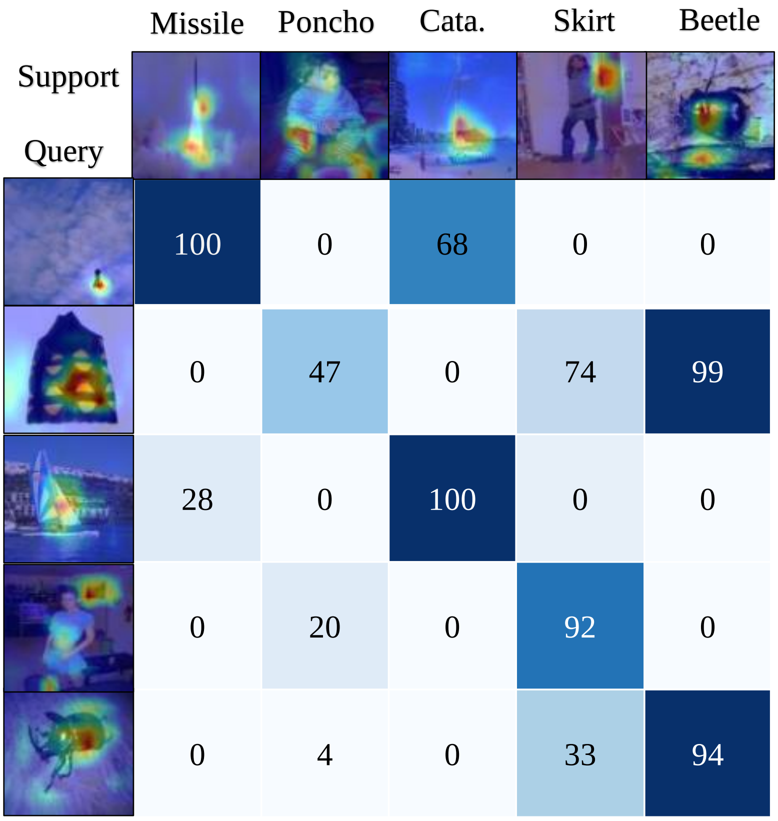


Figure 4.10: Pattern-based visualization of sample 2.



Support Query	Missile	Poncho	Cata.	Skirt	Beetle
	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

Figure 4.11: Pairwise matching of sample 2.

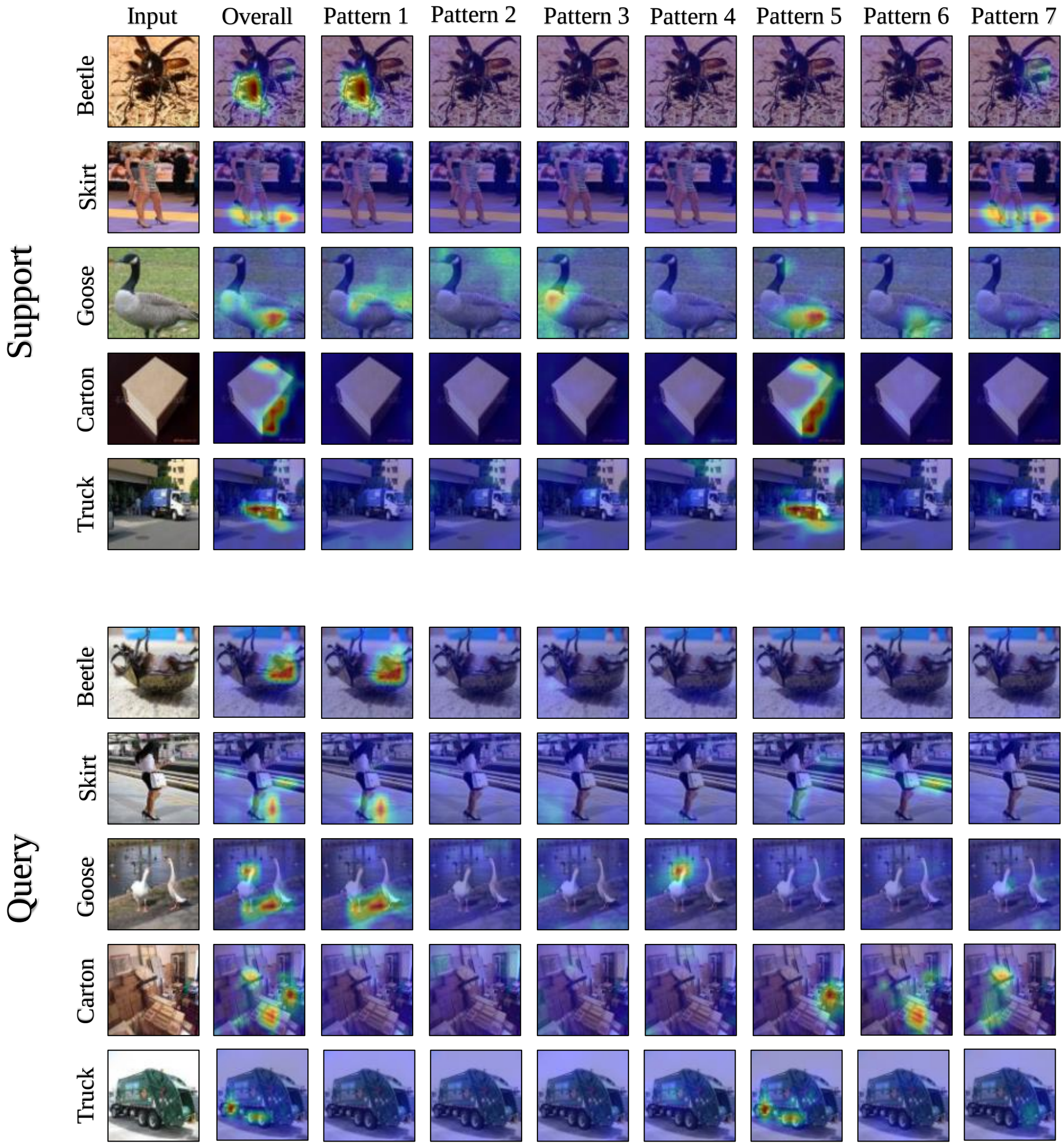
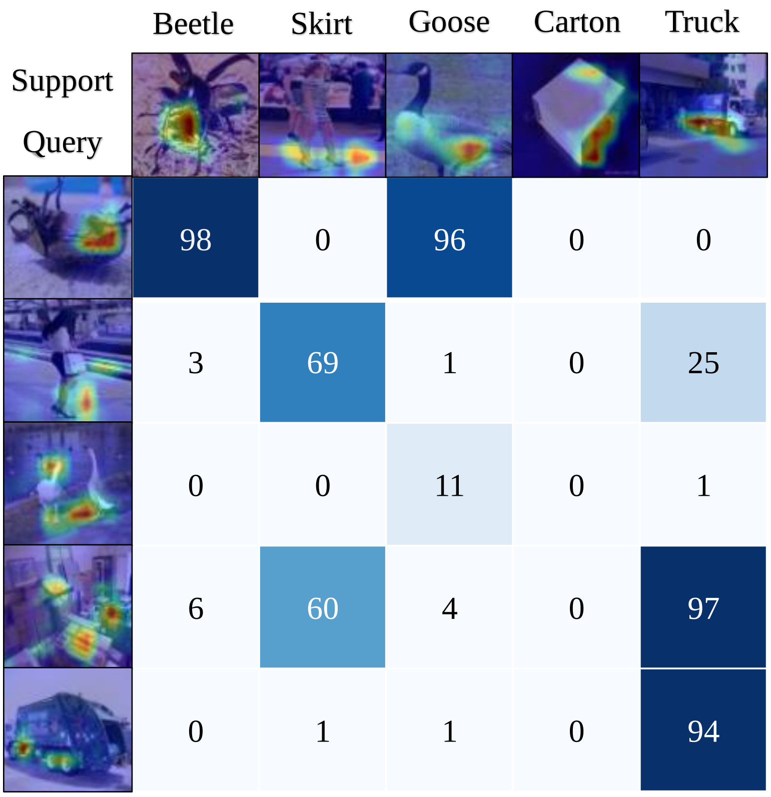


Figure 4.12: Pattern-based visualization of sample 3.



	Beetle	Skirt	Goose	Carton	Truck
	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

Figure 4.13: Pairwise matching of sample 3.

Chapter 5

Weakly Supervised Nodule Detection Via XAI

5.1 Overview

Chest X-ray (CXR) is a basic procedure in radiology for disease prediction and diagnosis. CXR images are taken as a composite shadow of the lungs, heart, mediastinum, and bones, and reading them is not an easy task. In the interpretation of CXR findings, double interpretation by independent doctors, including a radiological specialist, is desired. In Japan, numerous CXR scans are performed at healthcare or health checkup facilities, but there are too few radiological specialists available to cover all facilities. It is not uncommon for radiological specialists not to interpret CXR images, even in facilities where they interpret all computed tomography and magnetic resonance imaging scans. As such, patients may be at serious risk of important findings in CXR images being overlooked.

Positional information on disease lesions is provided by applying a bounding box to each imaging finding. Normally, these tasks must be performed manually on each image by medical professionals, which is a burdensome task. In hospitals, radiology reports are created by radiological specialists to communicate with referring clinicians. In these radiology reports, the clinical findings along with the positional information are described in the free-text form, along with positive and negative expressions (e.g., “There is a nodule in the left upper lung”). The positional information of the lung field is described in accordance with the circulation image technology (CITEC [139]) standards. In the CITEC standards, the left and right lung fields are divided into five fields: the apical lung field (above the collarbone), the upper lung field (between the collarbone and the second rib), the middle lung field (between the second and fourth ribs), and the lower lung field (below the fourth rib) and hilar portions (perihilar). If the information on the clinical findings and the positional information for each image could be extracted from a radiology report, the DCNN would have higher-resolution input with reference to positional information.

At Osaka University Hospital, radiographic images have been stored electronically in the Picture Archiving and Communication System (PACS) since 1999. As of December 2019, more than 1,000,000 images have been stored in the PACS, with over 400,000 of them including radiology reports written by radiological specialists. The radiographic images themselves are not labeled with clinical findings or position information. If this information can be labeled accurately based on radiological reports using information extraction techniques, our CXR images may constitute a good data set for machine learning that exceeds the Chest X-ray 14 data set. We previously reported on the extraction of information from CXR reports written in free-text form using an NLP technique [140]. We comprehensively extracted terms related to “ clinical findings ” and their “ positional information ” using the bi-directional long short-term memory (LSTM [141]) with the conditional random field (CRF) model and combined them to create structured data. Spelling variations and synonyms were then converted into representative terms. We have used 319,130 CXR reports stored in the diagnostic imaging report system of Osaka University Hospital. In terms of the accuracy of term extraction by machine learning, the average F1-score was 0.94, with 1,788 positional expressions and 8,807 clinical finding expressions obtained from all reports. The structured data of 824,539 records were ultimately extracted from the CXR reports.

In this study, we attempted to create a deep learning method [142] that detects disease lesions from CXR images using a data set annotated with extracted CXR report information. We set the nodule and mass shadow (hereafter, all together expressed as nodule) as the target disease lesion.

5.2 DNNs in Chest X-ray Diagnosis

With the development of AI, DNNs have demonstrated remarkable strength in a variety of tasks during medical image analyses. A highly accurate computer-aided diagnosis (CAD) is expected to become a powerful tool assisting doctors in interpreting medical images.

One popular topic for medical imaging is the analysis of Chest X-rays. The National Institute of Health (NIH) released a CXR dataset of 14 common thorax disease categories (Chest X-ray 14 [143]) in 2017. This data set contains 112,120 CXR images of 30,805 patients with an image resolution of 1024 $\times$ 1024. In the Chest X-ray 14 dataset, every CXR image is multi-labeled with 14 different diseases. The labels are generated by natural language processing (NLP) from the diagnostic reports, and the accuracy is over 90%.

With the Chest X-ray 14 dataset, many DNN-based approaches [143–148] have been proposed to diagnose thoracic diseases in CXR images automatically. However, none of these studies produced good results for diseases without an obvious finding (e.g., nodules). In the Chest X-ray 14 data set, fewer than 1000 images have positional information of the disease lesion, with most images having only the tag information of the diseases. This means that when using the Chest X-ray 14 data set, the entire image must be used in low resolution for the model input. However, image compression results in the loss of information. Wang *et al.* [143] evaluated several deep convolutional neural network (DCNN) architectures, reporting an area under the receiver operating characteristics (ROC) curve (AUC) of 0.75 on average. In their work, high-resolution CXR images are compressed before they are used as model input, restricting the performance of DNNs.

An important direction of chest disease diagnosis is nodule detection [149]. This target requires the finding of each nodule, and a straightforward way is using object detection methods

[150–152]. Many works contribute to automatic nodule detection using bounding boxes [153–155]. However, they are designed for analyzing CT or MRI, and the application of object detection for CXR still needs to be studied. One reason that hinders the research for detecting nodules in CXR is the annotation difficulty. Many nodules are small, and physicians may miss annotating them as it is not easy to find. Additionally, annotation itself is time costing and expensive. To overcome the insufficient data problem, many methods have been proposed (e.g., synthetic data [156], attention mechanism [157]). However, it is hard to say they can satisfy the need of real-world demands. In this thesis, we design a method that makes use of the CXR report record to realize a weakly-supervision of nodule detection on CXR images.

5.3 Our Dataset

The CXR images and their reports are taken from 2010 to 2019 at Osaka University Hospital were included in the study. The information on patient ID, examination date, clinical findings, and their location were extracted from CXR reports using NLP. The corresponding CXR images identified with the patient ID and examination date were downloaded from our PACS. The ID along with the date confirmed the CXR image as unique. We downloaded about 6,340 CXR images with nodules (other diseases such as cardiomegaly, pneumonia, and atelectasis are also potentially present) in DICOM format. We also randomly download about 6,636 CXR images without nodules (other diseases present) as negative samples. In addition, about 2,180 completely normal CXR images were also added as negative samples. All CXR images were taken in the radiography room in the standing frontal (P-A) position. The equipment used for radiography was DB BENEO (Fuji Film Medical), Digital Diagnost TH/VM, Digital Diagnost SRN (Phillips), Definium 800 (GE Healthcare), and RADspeed Pro (Shimadzu). All of these DICOM images were transformed into PNG, which retains the original resolution. All these images construct our original data set. It was used for nodule classification and detection.

Based on the CITEC standard, we divided the lung field into 8 divisions: the left and right apical lung field, upper lung field, middle lung field, and lower lung field [139]. We used

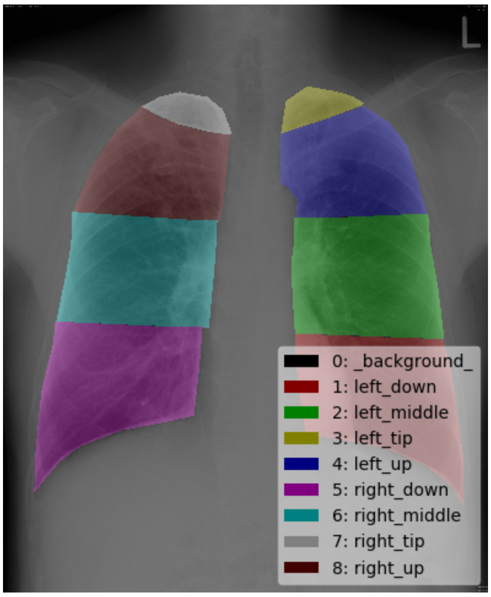


Figure 5.1: Example of masked CXR image. One CXR image was masked into eight lung fields. Each lung field will give different color which is shown in our sample. The background will be assigned by the color black.

“ Labelme ” as the data annotation software program. The RSCR contains 250 anonymized CXR images randomly taken from Osaka University Hospital. The DICOM format images were converted to PNG format and retained the original resolution. All labels were manually made and converted to binary image format. 80% were used for training, 10% were used for validation, and the remaining 10% were used for testing. One sample of the annotated image is shown in Figure 5.1.

For nodule detection evaluation, we also obtained 426 CXR images with 1,280 nodule objects annotated by professional physicians. All these patients do not appear in the training or validation set of the classification part.

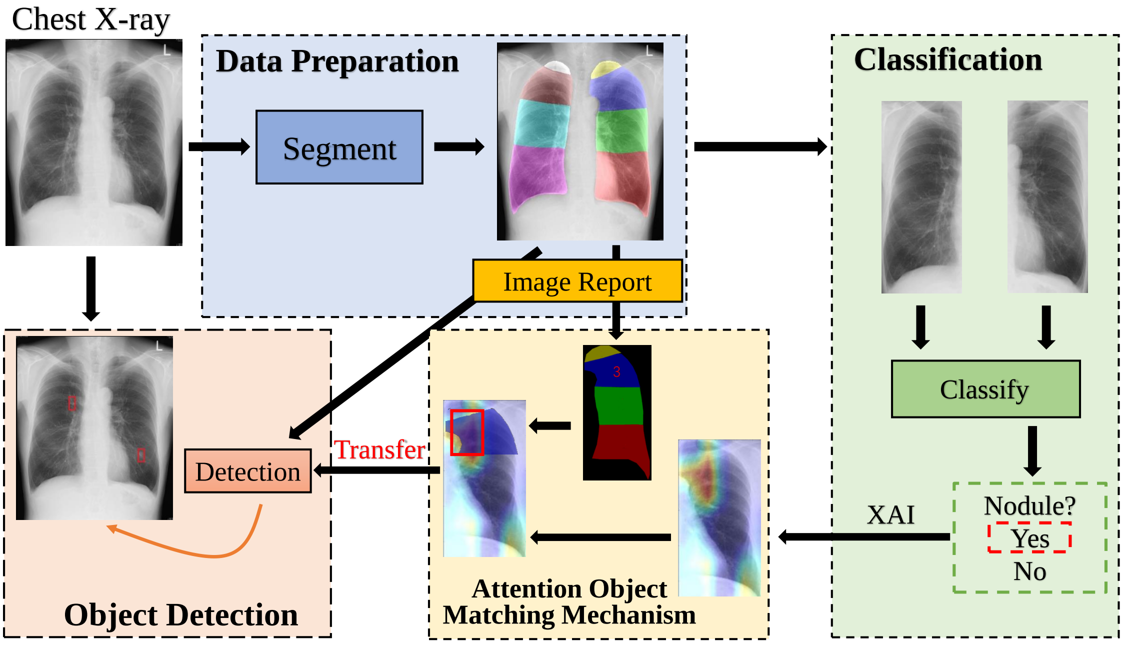


Figure 5.2: Pipeline of Our Weakly Supervised Nodule Detection

5.4 Method

5.4.1 Pipeline of Our Method

The overall flowchart is shown in Figure 5.2. First, by combining the CXR images and the information extracted from corresponding CXR reports, we made a data set names Regional Segmentation in Chest Radiographs (RSCR). Based on CITEC [139] standards, the whole lung was divided into eight parts described in CXR imaging reports. We used PSPNet [158] for the segmentation of lung images. Next, we adopted a classification model, ResNeSt-50d [159] to predict nodules in the segmented lung fields. We also created an attention map using the Grad-CAM [1] algorithm. If the area of attention matched the area annotated by the CXR report, the coordinate of the bounding box was considered as a possible nodule area. Finally, all possible nodule candidates are transferred to the object detection model for further prediction. For the object detection model, we chose the Faster-RCNN [150]. The bounding boxes generated by

Faster-RCNN were filtered to satisfy the location extracted from CXR reports.

Our weak supervision consisted of two aspects: i) eight-part lung field mask and nodule indicator from the CXR report. ii) attention area generated by the XAI method in classification. They are combined to generate training data for object detection of nodules (detailed in Section 5.4.4).

5.4.2 Lung Field Segmentation

Semantic segmentation is an important branch of image analysis. For our project, we need this technology to segment the lung field into eight parts corresponding to the report record. Previous research [160] usually uses U-Net [161] for binary lung field segmentation. Although it performs well in many medical tasks, we tried some commonly used methods for our multi-classes segmentation. We evaluated the results with mean intersection-over-union (IoU) of eight lung fields in RSCR with U-Net [161], PSPNet [158], DeepLab-V3 [162], FCN [163], and DA-Net [164]. All the input images are resized into 512x512. PSPNet gets the best outcome in the test set and is chosen for field mask generation. It is adopted to segment the left and right lungs for the continuous classification module.

5.4.3 Recognition of Nodules in The Left and Right Lungs

Classification Considering the amount of data, we applied nodule recognition in segmented left and right lungs instead of eight lung fields. Original CXR data will be divided into train and validation data. The number of classification datasets is shown in Table 1. All the positive data come from nodule images' segments according to CXR reports. Images in which a nodule was noted in either lung field were not used as negative data, even if there was no nodule noted in the other lung field. Both lung segments from non-nodule images and totally normal images are used for negative data.

We provide two training modes for classification called "All" and "Separate". In "All" mode, both left and right lung images shared one classification model. Therefore, the total training data

Table 5.1: Classification dataset.

	Left Lung		Right Lung	
	Train	Validation	Train	Validation
Positive	2943	538	3785	668
Negative	7493	1323	7493	1323

for a model is the sum of the train left and right image numbers recorded in Table 5.1. In this way, one model can own more training data. This model is evaluated by both left and right validation data. On the contrary, in "Separate" mode, each lung field will have its own model. The total training data for the left model are 2,943 positive plus 7,493 negative. And for the right model, there are 3,785 positive plus 7,493 negative. Training separately will relieve the recognition difficulty caused by the difference between the left and right lungs. Two models are evaluated in the validation set for the left and right lung respectively.

We tried some popular classification models including ResNeSt-50d [159], DenseNet-169 [165], Inception-V4 [166], and EfficientNet-E5 [167]. All the input segments are resized into 260×260. We trained 20 epochs for each model with the same setting parameters. The results are evaluated by AUC and F1 scores. The best performance in validation is achieved by ResNeSt-50d. It will be used to generate an attention map for the nodule area.

Attention Map In DCNNs, after multiple convolutions and pooling, the final convolution layer contains the most abundant spatial and semantic information. The next structures are the fully connected layer and the softmax layer. The information contained in these layers is difficult for humans to understand or be displayed visually. Therefore, to help CNN deliver a reasonable explanation of its classification results, the final convolutional layer must be fully utilized. For the segmented lung images recognized as having nodules, we used the GradCAM algorithm [1] to create an attention map. The last convolution of the CNN will generate a feature map with fixed image size (8×8 in this study) and channel number (2,048 in this study). Global

Average Pooling translates this structure into a full connection form for the final prediction. The GradCAM algorithm uses gradient information to obtain weights on feature maps and make adjustments. The outcome is then processed by the relu activate function and subjected to normalization and resizing in order to convert it to a mode suitable for humans to review. We also tried some other visualization methods like Deeplift [91], and GradCAM++ [92], which show no obvious difference from GradCAM. We only made an attention map for the segmented lung images recognized as having nodules.

5.4.4 Attention Object Matching Mechanism (AOMM)

Self-Intersection for CXR Reports To correctly capture the lesion of the predicted nodule, we created an attention map and determined the centers of all areas of attention with bounding boxes. First, all of the attention maps were processed by binarization with a threshold in order to reduce the proportion of the area of attention (fixed on the strong center). The pixel of a strong area of attention was then assigned a value of 255. In this study, we set the binarization threshold (**BT**) as 25, 75, and 125. A rectangle that is outlined as an intercept of the attention area that exceeds the binarization threshold was set as the bounding box (a green rectangle shown in Figure 5.3).

Next, using the lung segmentation model, we obtained the segmented areas of the eight lung fields in line with the CXR reports (blue area shown in Figure 5.3). We, therefore, use the segmented lung fields information and strong areas of attention to observe whether or not the prediction of the nodule by our classification model truly finds the nodule described in the CXR reports. This process computes a Self-Intersection (**SI**) value, which uses the following formula:

$$\mathrm{SI} = \frac{\mathrm{target} \cap \mathrm{attention}}{\mathrm{attention}}, \tag{5.1}$$

where “target” is the area of the segmented lung fields recorded with nodules and “attention” is the area of the strong area of attention. This function aims at computing the location consistency between computer prediction and report record. The larger the **SI** value, the higher the

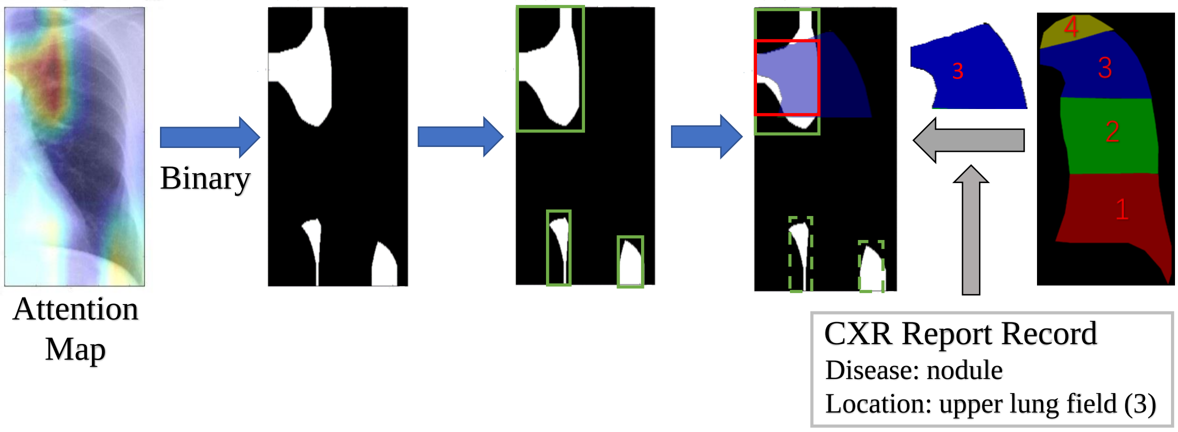


Figure 5.3: Attention Object Matching Mechanism (AOMM). The Attention map obtained by Grad-Cam was binarized, and the region defined by the binarization threshold was set as the attention area. If the attention area matched the region annotated by the CXR report, the coordinate of the bounding box was considered as a possible nodule area.

probability that the found area actually included a nodule. In this study, we set the threshold of **SI** as 0.7. If the calculated **SI** value was over 0.7, a rectangle was generated as an attention bounding box (a red rectangle shown in Figure 2) to outline the overlapping area of the attention area and segmented nodule area. Hereafter, we call this process the **SI** process.

Precision for Generated Attention Bounding Boxes We obtained 426 CXR images with nodules annotated by professional physicians. Manual annotation (bounding boxes) are usually small in size, while attention bounding boxes are usually large in size. We sought to detect the coincidence degree (**CD**) between the manual annotation bounding box and the attention bounding box. The formula is as follows:

$$\mathrm{CD} = \frac{\text{manual} \cap \text{generation}}{\text{manual}}, \tag{5.2}$$

where “manual” is the area of the manually annotated bounding boxes, and “generation” is the area of the generated attention bounding boxes. We defined that an attention bounding box was correctly generated if the attention bounding box enclosed more than 50% of the area of a manual bounding box. We defined (**Precision**) as the fraction of attention bounding boxes that satisfy the above definition. The formula is as follows:

$$\text{Precision} = \frac{\text{correct detection}}{\text{all detection}}, \tag{5.3}$$

where "all detection" is the total number of the generated attention bounding box and "correct detection" is the number of correctly generated boxes.

We also use a **Recall** metric to quantify the ratio of manual annotations covered by the generated boxes based on **CD**.

5.4.5 Object Detection of Nodule

We used the attention bounding boxes to train an object detection model. We built a new data set according to the format of Microsoft COCO [50] which is a common data structure in object detection. The Json file was used to record the coordinate of attention bounding boxes for each image. For the object detection model, we chose the Faster-RCNN [150].

The ResNet-50 [115] model was adopted as the pre-training model for the network. We also applied Feature Pyramid Networks (FPN [168]) for better feature generation. The next part is a region proposal network [150] analyzing the feature maps and proposing candidate nodule regions. This network estimated the probability of nodule/non-nodule based on a fixed set of anchors at each position of the feature map. The position and size of each anchor obtained by bounding box regression were then fine-tuned. We used three anchor scales (64, 128, and 256) and three anchor ratios (1:2, 1:1, and 2:1) in the present study. The feature maps and nodule proposals were then sent to a region of interest (RoI) pooling layer, which set all feature maps in a proposal at a fixed size (7×7 in this study). Finally, a 1024D feature vector is sent to two full connection layers to predict the bounding box regression for further fine-tuning and the confidence scores for each nodule proposal. Non-maximum suppression (NMS) is applied to the bounding boxes to decide the final predictions. Based on the priority of confidence for each detection in one image, NMS will drop the bounding box overlapped around a certain area. The IoU threshold for NMS is 0.7 for training and 0.2 for inference.

The output of Faster-RCNN is the bounding boxes possibly containing nodules. Each box has its confidence (scores calculated by Faster-RCNN) as a nodule candidate. By setting a threshold for confidence, the generation number of nodule detection is different. In this study, it is important to ensure that the generated bounding box contains nodules. Therefore, it is preferred to set the confidence level high. In this study, we set the confidence to 0.7. In addition, during the test, all the boxes which satisfied the setting confidence threshold were then filtered to match the location extracted from CXR reports using **SI** process (we use the whole bounding box area as attention for **SI** process). We confirm that this operation makes the pipeline not purely weakly supervised learning. For object detection of nodules, it is detection with auxiliary information. However, it can remove meaningless detection and improve accuracy, which is important for the application of nodule annotation. Hereafter, the generated bounding box is called the detected bounding box.

Table 5.2: Segmentation results.

Model	U-Net [161]	FCN [162]	DA-Net [164]	DeepLab-V3 [162]	PSPNet [158]
mIoU (%)	0.832	0.788	0.863	0.878	**0.889**

5.5 Results

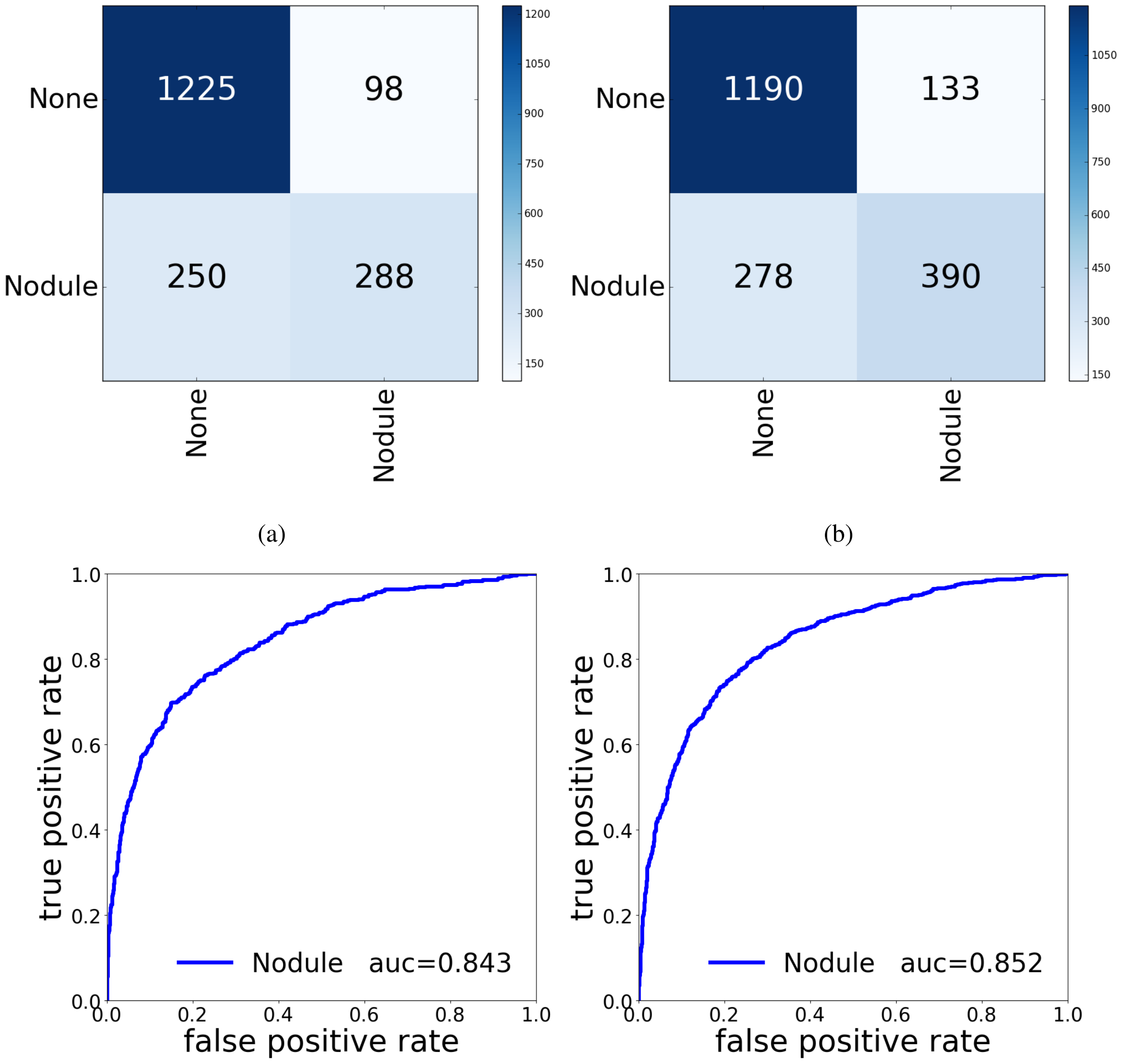
5.5.1 Lung field Segmentation Analysis

We used different models to conduct comparative experiments. The mean IoU was used to evaluate the outcome. The results are shown in Table 5.2. We find PSPNet outperforms other models.

5.5.2 Nodule Classification Analysis

Table 5.3 shows the results of each classification model with the different training modes. “All” training mode tends to provide a better result, which shows the superiority of joint training. Although Inception-V4 and EfficientNet-E5 got higher F1 scores, they showed serious overfitting in the training set. We show the train results of the “All” mode in Table 5.4. In the continuing AOMM module, we need a classification model to generate an attention map for nodule object matching. An overfitting model will provide inaccurate features which influence object detection. ResNeSt-50d got two top ranks in evaluation. So, we choose it as our final classification model.

The confusion matrixes of the best model ResNeSt-50d in “All” mode are shown in Figure 5.4. The AUC of ResNeSt-50d was 0.843 for the left lung and 0.852 for the right lung. No marked difference in the AUC of the lung fields was noted. According to the confusion matrix, both the left and right models had a strong recognition ability for non-nodule cases.



(a) (b)

(c) (d)

Figure 5.4: Confusion Matrix and ROC Curve of ResNeSt-50d. (a) confusion matrix for the left lung, (b) confusion matrix for the right lung, (c) ROC curve for the left lung, (d) ROC curve for the right lung. None referred to the non-nodule imaging data, while nodule referred to images with nodules.

Table 5.3: Classification results of the validation set.

	All Mode				Separate Mode			
	Left Lung		Right Lung		Left Lung		Right Lung	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
DenseNet-169 [165]	0.841	0.622	0.837	0.651	0.824	0.637	0.839	0.680
Inception-V4 [166]	0.836	0.646	0.842	**0.689**	0.831	0.620	0.829	0.671
EfficientNet-E5 [167]	0.835	**0.676**	0.833	0.678	0.835	0.674	0.841	0.678
ResNeSt-50d [159]	**0.843**	0.623	**0.852**	0.655	0.826	0.627	0.841	0.649

Table 5.4: Classification results of the training set in “All” mode.

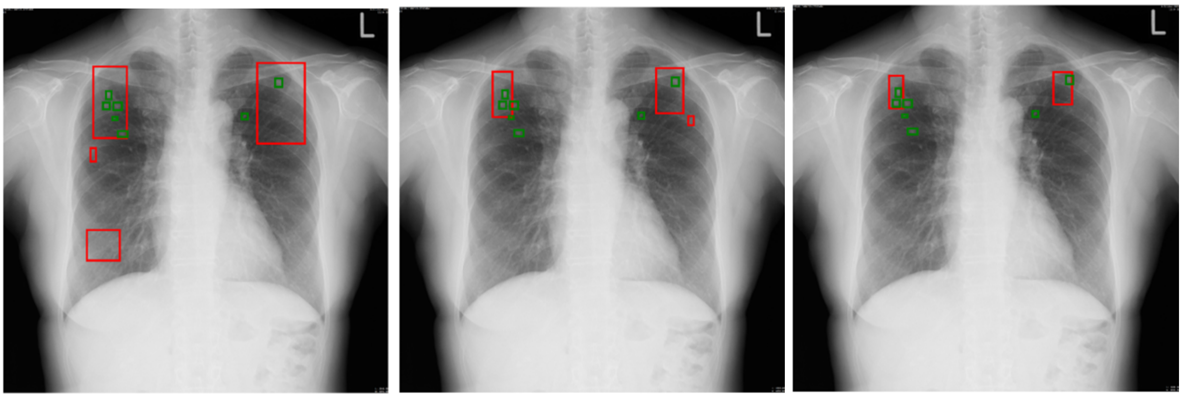
	Left Lung		Right Lung	
	AUC	F1	AUC	F1
DenseNet-169 [165]	0.890	0.687	0.885	0.712
Inception-V4 [166]	0.931	0.778	0.924	0.783
EfficientNet-E5 [167]	0.992	0.986	0.998	0.991
ResNeSt-50d [159]	0.881	0.680	0.873	0.713

5.5.3 Analysis of AOMM

Through AOMM, attention bounding boxes of nodules are obtained. With a different set of **SI** and **BT**, the generation will show different qualities. We fix the **SI** value as 0.7, and **BT** is tried with 25, 75, and 125. The samples of the attention bounding box are shown in Figure 5.5. In Table 5.5, we show the results of attention prediction and human annotations (BT: binarization threshold, Correct: the number of correctly predicted attention bounding boxes, Total: The total number of generated attention bounding boxes, Size: the average size of correctly predicted attention bounding boxes (the total pixel amount in original resolution CXR image). About half of the generated attention bounding boxes include the nodules. A lower binary setting tends to generate larger bounding boxes which make it easier to satisfy **CD** condition. On the contrary,

Table 5.5: Precision of attention bounding box generated by AOMM.

BT	Precision (correct/total)	Recall	Size (K)
25	0.531 (297/559)	0.625	241
75	0.490 (244/498)	0.452	138
125	0.341 (160/469)	0.418	95



(a) BT-25

(b) BT-75

(c) BT-125

Figure 5.5: Samples of attention bounding boxes. Red boxes are predicted by the model and green boxes are annotated by professional physicians. The binarization threshold (**BT**) was set as 25, 75, and 125. **SI**, which represents the overlap with the field extracted from the CXR report, was set as over 0.7

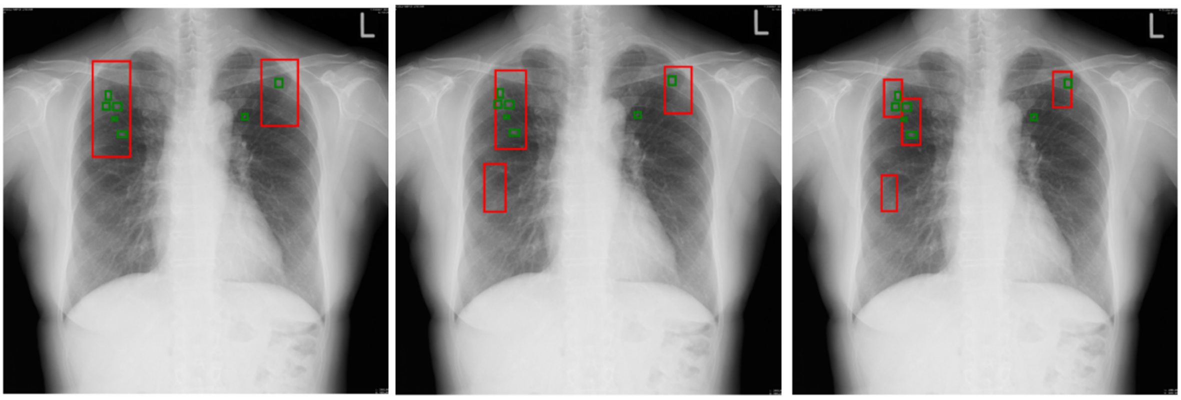
a high binary setting can pinpoint the attention area but performs worse prediction results. We also provide the average size of correctly predicted bounding boxes. It represents the total pixel amount owned by them in the original resolution CXR image (using unit K). In our original data set, all the CXR images have a resolution of over 2,000 x 2,000. The average size of boxes annotated by professional physicians is 18K.

5.5.4 Nodule Detection Analysis

If the binarization threshold for the attention map is set low, the area of generated bounding box becomes large. The larger the area of the bounding box, the greater the possibility that the area

Table 5.6: Precision of detected bounding box generated by Faster-RCNN and filtered to satisfy the location extracted from CXR reports.

Generation	Precision (correct/total)	Recall	Size (K)
BT-25	0.800 (284/355)	0.603	221
BT-75	0.776 (180/232)	0.412	128
BT-125	0.567 (51/90)	0.251	90



(a) BT-25 (b) BT-75 (c) BT-125

Figure 5.6: Samples of detected bounding boxes The bounding boxes generated by Faster-RCNN were filtered to satisfy the location extracted from CXR reports. Red boxes are predicted by the model and green boxes are annotated by professional physicians.

will contain nodules. Therefore, the detection precision will be high but will also include extra areas without nodules. When training object detection with the attention bounding boxes, it is uncertain which BT setting will produce better prediction results. For this reason, generated attention bounding boxes with a binary setting of **BT-25**, **BT-75**, and **BT-125** were used for the object detection training. Samples of the detected bounding boxes by Faster-RCNN are shown in Figure 5.6.

We also provide the prediction results for each BT setting. In order to evaluate the precision of model detection, we use the **Precision** standard defined in AOMM analysis (Table 5.6). In every binary setting, the Precision of the detected bounding boxes is improved compared to that of the attention bounding box. Best Precision results boost to 0.800 in **BT-25**, 0.776 in **BT-75**, and 0.567 in **BT-125**. In **BT-25** and **BT-75**, the size of the detected bounding box becomes smaller than that of the attention bounding box. On the other hand, only slight changes in box size were observed in the **BT-125**.

5.6 Discussion

In CXR reports, the location of the pulmonary findings is often described in the left and right apical lung field, upper lung field, middle lung field, and lower lung field. Therefore, the CXR report can be used as annotation information if the left and right pulmonary fields can be divided into these eight regions on the CXR image. In the present study, PSPNet was able to segment the lung field.

For the classification of the nodule images, we planned to use the PSPNet to segment the CXR image into eight lung fields. However, considering that the amount of data was not sufficient to divide the data into eight regions and that the nodular shadows may occasionally cross over two regions, we classified the left and right lung fields according to the presence or absence of nodules. According to the confusion matrix of ResNeSt-50d, the recall was insufficient. One reason may be due to the imbalance in the data for nodules and non-nodules. Another reason may be the size of some nodules being too small to be detected. Although our segmentation process provides a higher-resolution image, resizing is also necessary for classification model input. Compared with the original high-resolution CXR image, some small nodule findings become difficult to discriminate. In addition, smaller nodules that are detected in CXR images but may not be pathological were sometimes described in the CXR report and sometimes not. These smaller nodules may be hard to recognize with our classification model. This problem may be solved by only including mass shadows in the study, as a mass is larger than the nodule will always be included in the CXR report.

In the present study, we utilized the attention map, which provides the basis for the classification of nodules, to assign a bounding box to nodules. It is well known that the basis for computer judgments may differ from that of human perception. Therefore, we assumed that the attention region that corresponds to the location of the nodule in the CXR report was the region where the nodule shadow was recognized and set the bounding box at the corresponding region. For attention generation, we applied GradCAM [1] this time. It tends to generate one strong attention center and is not suitable for a situation where many nodules appear in one image.

The detection precision of the attention bounding boxes is not high enough. However, through the object detection process and **SI** process, the detection precision of the detected bounding boxes is improved. Moreover, the size of detected bounding boxes is also decreased in the binary setting of **BT-25** and **BT-75**. The fact that the percentage of bounding boxes that correctly enclose a nodule is increasing while the size of the bounding box is decreasing means that the accuracy of the bounding box generation is greatly increased by this process.

The size of the detected bounding box generated in this study differs depending on the binarization threshold for the attention map. If the area of the bounding box becomes large, the detection precision of containing nodules becomes high. However, the bounding box also includes extra areas without nodules. Considering the use of bounding boxes as training data for machine learning, we need to discuss which should be prioritized: precision or box size. The generation of detected bounding boxes from attention bounding boxes by Faster-RCNN showed an improvement in detection precision of 51% for **BT-25**, 58% for **BT-75**, and 66% for **BT-125**. This fact suggests that the balance between precision and box size is important. As the detection precision was not significantly different between the **BT-25** and **BT-75**, and the box size of the **BT-75** was 60% of that of the **BT-25**, our best result was thought to be obtained with a **BT-75** and confidence with 0.7.

Finally, we achieved a precision of 77.6% in generating a bounding box containing nodules. The generated bounding box still contains a bounding box without nodules, which may not be accurate enough to be used as train data. A possible solution to this problem is to have the specialist visually check the generated bounding boxes. The CXR images are already marked with bounding boxes, and the specialist only has to exclude the nodule-free bounding boxes visually. Therefore, the workload of specialists is significantly reduced compared to the manual annotation process, where the specialist diagnoses the nodules one image at a time and assigns a binding box to each image.

In the CXR report, various lesions, such as pneumonia, atelectasis, and pleural effusion, are described with their locations. Our proposed method is expected to create a bounding box for other lung lesions as well as a nodule. In Osaka University Hospital, more than 400,000 CXR images and corresponding CXR reports are stored in the PACS and the data warehouse. We also have a system for outputting radiographic images stored in PACS in a batch by describing the extraction conditions [169]. If it is possible to create a dataset for these large images in which various lung lesions described in CXR reports are assigned a bounding box, it will be possible to create a huge dataset that exceeds the Chest X-rays 14 data set.

5.7 Summary

We proposed a machine-learning method to generate bounding boxes with nodules on CXR images based on the positional information of the diseases extracted from the CXR reports. Based on the attention map generated from the nodule classification model and the nodule location information extracted from the CXR report, a bounding box could be assigned to the nodules on the CXR image. Through the object detection process and **SI** process, the detection precision of the bounding boxes can be improved. Our method has the potential to provide bounding boxes for various lung lesions, which can reduce the annotation burden for specialists.

Chapter 6

Overall Discussion

In this thesis, our core concept is to explore the potential of combining the thought of learning representation and XAI. Representation learning is a broad field of study that is involved in various fields such as machine learning, image processing, etc. In order to present our theoretical basis in a concrete way, we explore how to improve the DNNs' ability to learn representations in terms of image retrieval, few-shot learning, and weak-supervision.

In chapter 3, we proposed our thought that DNNs should be designed according to the target task. In this image retrieval work, we want to prove that task-specific DNNs contribute to better learning of representation. We introduce the multi-task, multi-label, hierarchy characters of our dataset to the design of DNNs. The experiment results support our thought. Next, in chapter 4, we try to organize a few-shot learning task with the idea inheritance from the last chapter. Additionally, we adopt the advantage of XAI when designing DNNs. We use a learnable XAI module to enable learning better representation and interpreting the behavior of the few-shot pipeline. This work proves our core concept of the combination between representation learning and XAI. In chapter 5, we implement our concept in an application for chest X-ray diagnosis. This work is a weak-supervision integrated with multiple technologies under the principle of representation and XAI combination. Our results demonstrate the potential of such a combination for the real world application. The whole thesis is constructed step by step, from the thought of designing DNNs to our core concept and, finally, an application.

Chapter 7

Conclusion

In this thesis, we improved the DNNs technology, aiming at learning better representation and interpreting the behavior of DNNs. Our theory is demonstrated step-by-step through three different scenarios. We first proved that designing DNN according to the target task is essential for learning representation. We then applied the thought of XAI to the design of DNNs. Our model shows better ability in learning representation and is interpretable to its decision. Finally, we further adopted XAI methods to realize weak supervision for nodule detection in the diagnosis of CXR.

In Chapter 3, we implement our retrieval experiments on a dataset for ethnological museum exhibitions. The design of DNNs is according to the multi-task, multi-label, and hierarchy properties of this dataset. Using a multi-task DNN structure, a soft-similarity loss, and a hierarchy loss, the learned representation from our model is constrained to be meaningful. The retrieval performance is also much better than a usual DNN, showing that a task-specialized DNN can accommodate real needs.

Next, we discussed whether DNNs could be interpretable to its decision while owning a better ability to learn representation in chapter 4. We embedded trainable patterns into the pipeline for an FSL task. Our designed patterns can learn representation from base classes and transfer them to novel classes efficiently. The slot-attention structure of the patterns enables a transparent FSL inference, enhancing its potential for actual uses. Additionally, the high FSL classification performance and XAI evaluation prove that a DNN model is interpretable, and the learning of representation benefited from this interpretability.

In Chapter 5, we explore the possibility of using XAI methods for weak supervision of nodule detection. In the diagnosis of CXR, we applied XAI methods to a nodule classification task to generate bounding boxes that possibly locate nodules. The generated bounding boxes are then utilized for training a nodule detection task successfully. XAI methods realize the transfer of representation between these two tasks and construct a weak supervision pipeline for nodule detection. This is very enlightening because XAI methods can do more besides interpret the behavior of DNNs. Our method is also potential for the detection of other diseases, thus relieving the annotation burden for specialists.

For the future of AI, the design of DNN models is required to accommodate the target task in order to learn better representation. An explainable approach is also needed to interpret DNNs' behavior to enable a reliable AI. We explored the combination of these two aspects and evaluated our theory on some real-world needs. We believe that a task-specialized and interpretable DNN can satisfy the need of many scenarios, thus promoting the application of DNNs in our daily life.

Acknowledgements

This work was completed under the supervision of Professor Hajime Nagahara, Associate Professor Yuta Nakashima, and Assistant Professor Liangzhi Li of the Graduate School of Information Science and Technology, Osaka University.

The author would like to express his deepest appreciation to Associate Professor Yuta Nakashi for his constant guidance and help. He would particularly thank the valuable suggestions and ideas from Assistant Professor Liangzhi Li and Professor Hajime Nagahara during research. Without their encouragement and assistance during his Ph.D., the dissertation would have been impossible.

Our medical data is collected from Osaka University hospital. Here, the author wants to thank the support and instruction from Professor Toshihiro Takeda and Professor Yasushi Matsumura.

Our ethnological museum dataset is based on a database provided by the National Museum of Ethnology, MINPAKU, together with OWC and OCM definitions by the HRAF Association. Here we express our gratitude for their help.

Finally, the author would like to thank his family for their support and encouragement during his doctoral studies.

Reference

[1] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pp. 618–626, 2017.

[2] Giang Nguyen, Stefan Dlugolinsky, Martin Bobák, Viet Tran, Álvaro López García, Ignacio Heredia, Peter Malík, and Ladislav Hluchỳ. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, Vol. 52, No. 1, pp. 77–124, 2019.

[3] Pravin Chandra Manoj Kumar Gupta. A comprehensive survey of data mining. *International Journal of Information Technology*, 2020.

[4] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*, 2016.

[5] Xiaowen Chu Shuoheng Yang, Yuxin Wang. A survey of deep learning techniques for neural machine translation. *arXiv:2002.07526*, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *Transactions on neural networks and learning systems*, Vol. 32, No. 2, pp. 604–624, 2020.

[8] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, Vol. 16, No. 6, pp. 345–379, 2010.

[9] Duygu Mutlu-Bayraktar, Veysel Cosgun, and Tugba Altan. Cognitive load in multimedia learning environments: A systematic review. *Computers & Education*, Vol. 141, p. 103618, 2019.

[10] Afshan Latif, Aqsa Rasheed, Umer Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil. Content-based image retrieval and feature extraction: a comprehensive review. *Mathematical Problems in Engineering*, Vol. 2019, , 2019.

[11] Wei Chen, Yu Liu, Weiping Wang, Erwin M Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep learning for instance retrieval: A survey. *TPAMI*, 2022.

[12] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pp. 2001–2010, 2017.

[13] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, Vol. 1, No. 5, pp. 206–215, 2019.

[14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, Vol. 35, No. 8, pp. 1798–1828, 2013.

[15] Nan Wu, Stanislaw Jastrzebski, Kyunghyun Cho, and Krzysztof J Geras. Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In *ICML*, pp. 24043–24055, 2022.

[16] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *NeuIPS*, Vol. 19, , 2006.

[17] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[18] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, Vol. 3, No. 1, pp. 1–40, 2016.

[19] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, Vol. 109, No. 1, pp. 43–76, 2020.

[20] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 11, No. 5, pp. 1–46, 2020.

[21] Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, Vol. 6, pp. 39501–39514, 2018.

[22] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, Vol. 60, pp. 4–21, 2017.

[23] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, Vol. 71, pp. 158–172, 2017.

[24] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *NeuIPS*, Vol. 32, , 2019.

[25] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *SIGKDD*, pp. 3505–3506, 2020.

[26] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *MIPRO*, pp. 0210–0215, 2018.

[27] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, Vol. 32, No. 11, pp. 4793–4813, 2020.

[28] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *BMVC*, 2018.

[29] Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. SCOUTER: Slot attention-based classifier for explainable image recognition. *ICCV*, 2021.

[30] Bowen Wang, Liangzhi Li, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. MTUNet: Few-shot image classification with visual explanations. In *CVPR Workshops*, pp. 2294–2298, 2021.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

[32] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE transactions on knowledge and data engineering*, Vol. 31, No. 10, pp. 1863–1883, 2018.

[33] Wenzhong Guo, Jianwen Wang, and Shiping Wang. Deep multimodal representation learning: A survey. *IEEE Access*, Vol. 7, pp. 63373–63394, 2019.

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, Vol. 60, No. 2, pp. 91–110, 2004.

[35] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, Vol. 1, pp. 886–893, 2005.

[36] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. *NeurIPS*, Vol. 13, , 2000.

[37] Ting Liu, Andrew Moore, Ke Yang, and Alexander Gray. An investigation of practical approximate nearest neighbor algorithms. *NeuIPS*, Vol. 17, , 2004.

[38] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

[39] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, Vol. 99, pp. 518–529, 1999.

[40] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. *NeurIPS*, Vol. 22, pp. 1509–1517, 2009.

[41] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, No. 1, 2014.

[42] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, No. 1, 2016.

[43] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *ICCV*, pp. 5608–5617, 2017.

[44] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *CVPR*, pp. 1229–1237, 2018.

[45] Jiehao Xu, Chengyu Guo, Qingjie Liu, Jie Qin, Yunhong Wang, and Li Liu. Dha: Supervised deep learning to hash with an adaptive loss function. In *ICCV*, pp. 0–0, 2019.

[46] Zheng Zhang, Qin Zou, Yuewei Lin, Long Chen, and Song Wang. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. *TMM*, Vol. 22, No. 2, pp. 540–553, 2019.

[47] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *CVPR*, pp. 3083–3092, 2020.

[48] Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *CVPR*, pp. 3311–3319, 2020.

[49] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, pp. 390–405, 2018.

[50] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755, 2014.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, Vol. 115, No. 3, pp. 211–252, 2015.

[52] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *ICMR*, pp. 1–9, 2009.

[53] Viraj Uday Prabhu. *Few-shot learning for dermatological disease diagnosis*. PhD thesis, Georgia Institute of Technology, 2019.

[54] Abdur R Feyjie, Reza Azad, Marco Pedersoli, Claude Kauffman, Ismail Ben Ayed, and Jose Dolz. Semi-supervised few-shot learning for medical image segmentation. *IEEE International Conference on Bioinformatics and Biomedicine*, 2021.

[55] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pp. 3630–3638, 2016.

[56] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pp. 4077–4087, 2017.

[57] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *ICLR*, 2018.

[58] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshop*, Vol. 2, 2015.

[59] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[60] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, pp. 11–20, 2019.

[61] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *ICLR*, 2018.

[62] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pp. 1199–1208, 2018.

[63] Jiahao Wang, Bin Song, Dan Wang, and Hao Qin. Two-stream network with phase map for few-shot classification. *Neurocomputing*, Vol. 472, pp. 45–53, 2022.

[64] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. Meta-learning for multi-label few-shot classification. In *WACV*, pp. 3951–3960, 2022.

[65] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pp. 3320–3328, 2014.

[66] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

[67] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2017.

[68] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[69] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to learn quickly for few-shot learning. *ICML*, 2017.

[70] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *ICLR*, 2018.

[71] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, pp. 8680–8689, 2019.

[72] Zhengping Hu, Zijun Li, Xueyu Wang, and Saiyue Zheng. Unsupervised descriptor selection based meta-learning networks for few-shot classification. *Pattern Recognition*, Vol. 122, p. 108304, 2022.

[73] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *CVPR*, pp. 3349–3358, 2018.

[74] Tomas Pfister, James Charles, and Andrew Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *ECCV*, pp. 814–829. Springer, 2014.

[75] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *CVPR*, pp. 12836–12845, 2020.

[76] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *International Conference on Artificial Neural Networks*, 2020.

[77] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *ICLR*, 2020.

[78] Wonho Bae, Junhyug Noh, and Gunhee Kim. Rethinking class activation mapping for weakly supervised object localization. In *ECCV*, pp. 618–634. Springer, 2020.

[79] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Rui-Wei Zhao, Tao Zhang, Xuequan Lu, and Shang Gao. Cream: Weakly supervised object localization via class re-activation mapping. In *CVPR*, pp. 9437–9446, 2022.

[80] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, Vol. 115, No. 3, pp. 211–252, 2015.

[81] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, 2020.

[82] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *ICLR*, 2020.

[83] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *ICCV*, pp. 2942–2950, 2017.

[84] Zbigniew Wojna, Alex Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, and Julian Ibarz. Attention-based extraction of structured information from street view imagery. In *ICDAR*, pp. 844–850, 2017.

[85] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *CVPR*, pp. 4942–4950, 2018.

[86] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

[87] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[88] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting CNNs via decision trees. In *CVPR*, pp. 6261–6270, 2019.

[89] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you? Explaining the predictions of any classifier. In *KDD*, pp. 1135–1144, 2016.

[90] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pp. 2921–2929, 2016.

[91] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, p. 3145–3153, 2017.

[92] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, pp. 839–847, 2018.

[93] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019.

[94] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *CVPR Workshops*, pp. 24–25, 2020.

[95] Haofan Wang, Rakshit Naidu, Joy Michael, and Soumya Snigdha Kundu. SS-CAM: Smoothed Score-CAM for sharper visual feature localization. *arXiv preprint arXiv:2006.14255*, 2020.

[96] Saurabh Desai and Harish G. Ramaswamy. Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*, pp. 972–980, 2020.

[97] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, pp. 3429–3437, 2017.

[98] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NeurIPS*, pp. 6967–6976, 2017.

[99] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *ICCV*, pp. 2950–2958, 2019.

[100] Zhongang Qi, Saeed Khorram, and Li Fuxin. Visualizing deep networks by optimizing with integrated gradients. In *AAAI*, Vol. 34, pp. 11890–11898, 2020.

[101] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

[102] Chin Chou Pei-Ying Hsu, Chiao-Ting Chen and Szu-Hao Huang. Explainable mutual fund recommendation system developed based on knowledge graph embeddings. *Applied Intelligence*, 2022.

[103] Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek, Yunqing Zhao, Ngai-Man Cheung, and Alexander Binder. Explain and improve: Cross-domain few-shot-learning using explanations. *arXiv:2007.08790*, 2020.

[104] Yuxia Geng, Jiaoyan Chen, Zhiquan Ye, Wei Zhang, and Huajun Chen. Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs. *SWJ*, 2020.

[105] Leonid Karlinsky, Joseph Shtok, Amit Alfassy, Moshe Lichtenstein, Sivan Harary, Eli Schwartz, Sivan Doveh, Prasanna Sattigeri, Rogerio Feris, Alexander Bronstein, et al. StarNet: towards weakly supervised few-shot detection and explainable few-shot classification. *AAAI*, 2021.

[106] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, Vol. 10, No. 7, p. e0130140, 2015.

[107] Bowen Wang, Liangzhi Li, Yuta Nakashima, Takehiro Yamamoto, Hiroaki Ohshima, Yoshiyuki Shoji, Kenro Aihara, and Noriko Kando. Image retrieval by hierarchy-aware deep hashing based on multi-task learning. In *ICMR*, pp. 486–490, 2021.

[108] Mohammad Norouzi and David J Fleet. Minimal loss hashing for compact binary codes. In *ICML*, 2011.

[109] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *CVPR*, pp. 2074–2081, 2012.

[110] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. Deep supervised hashing for multi-label and large-scale image retrieval. In *ICMR*, pp. 150–158, 2017.

[111] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. Graph convolutional network hashing. *IEEE Transactions on Cybernetics*, Vol. 50, No. 4, pp. 1460–1472, 2018.

[112] Qingshan Liu, Guangcan Liu, Lai Li, Xiao-Tong Yuan, Meng Wang, and Wei Liu. Reversed spectral hashing. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 6, pp. 2441–2449, 2017.

[113] Jie Gui, Tongliang Liu, Zhenan Sun, Dacheng Tao, and Tieniu Tan. Supervised discrete hashing with relaxation. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 3, pp. 608–617, 2016.

[114] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, Vol. 25, pp. 1097–1105, 2012.

[115] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

[116] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. Deep quantization network for efficient image retrieval. In *AAAI*, No. 1, 2016.

[117] Yun Gu, Chao Ma, and Jie Yang. Supervised recurrent hashing for large scale video retrieval. In *ACM MM*, pp. 272–276, 2016.

[118] Jie Qin, Li Liu, Mengyang Yu, Yunhong Wang, and Ling Shao. Fast action retrieval from videos via feature disaggregation. *Computer Vision and Image Understanding*, Vol. 156, pp. 104–116, 2017.

[119] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep video hashing. *TMM*, Vol. 19, No. 6, pp. 1209–1219, 2016.

[120] Mahnaz Amiri Parian, Luca Rossetto, Heiko Schuldt, and Stéphane Dupont. Are you watching closely? content-based retrieval of hand gestures. In *ICMR*, pp. 266–270, 2020.

[121] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

[122] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint arXiv:1802.03426*, 2018.

[123] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, Vol. 53, No. 3, pp. 1–34, 2020.

[124] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. *NeurIPS*, Vol. 33, pp. 2734–2746, 2020.

[125] Bowen Wang, Liangzhi Li, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. Match them up: visually explainable few-shot image classification. *Applied Intelligence*, pp. 1–22, 2022.

[126] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pp. 1492–1500, 2017.

[127] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *NeurIPS*, 2020.

[128] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pp. 4367–4375, 2018.

[129] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *ICLR*, 2019.

[130] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[131] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[132] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

[133] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *NeurIPS*, 2020.

[134] Liang Li, Weidong Jin, and Yingkun Huang. Few-shot contrastive learning for image classification and its application to insulator identification. *Applied Intelligence*, pp. 1–16, 2021.

[135] Tsendsuren Munkhdalai and Adam Trischler. Metalearning with hebbian fast weights. *arXiv preprint arXiv:1807.05076*, 2018.

[136] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, pp. 7229–7238, 2018.

[137] Tintrim Dwi Ary Widhianingsih and Dae-Ki Kang. Augmented domain agreement for adaptable meta-learner on few-shot classification. *Applied Intelligence*, pp. 1–17, 2021.

[138] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *ICLR*, 2019.

[139] Japanese Society of Circulation Imaging Technology. Lung field standard. *http://citec.kenkyuukai.jp/special/index.asp?id=25698*, 2019.

[140] Kento Sugimoto, Toshihiro Takeda, Jong-Hoon Oh, Shoya Wada, Shozo Konishi, Asuka Yamahata, Shiro Manabe, Noriyuki Tomiyama, Takashi Matsunaga, Katsuyuki Nakanishi, and Yasushi Matsumura. Extracting clinical terms from radiology reports with deep learning. *Journal of Biomedical Informatics*, Vol. 116, p. 103729, 2021.

[141] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[142] Bowen Wang, Toshihiro Takeda, Kento Sugimoto, Jiahao Zhang, Shoya Wada, Shozo Konishi, Shirou Manabe, Katsuki Okada, and Yasushi Matsumura. Automatic creation of annotations for chest radiographs based on the positional information extracted from radiographic image reports. *Computer Methods and Programs in Biomedicine*, Vol. 209, p. 106331, 2021.

[143] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, pp. 2097–2106, 2017.

[144] Han Liu, Lei Wang, Yandong Nan, Faguang Jin, Qi Wang, and Jiantao Pu. Sdfn: Segmentation-based deep fusion network for thoracic disease classification in chest x-ray images. *Computerized Medical Imaging and Graphics*, Vol. 75, pp. 66–73, 2019.

[145] Jinzheng Cai, Le Lu, Adam P Harrison, Xiaoshuang Shi, Pingjun Chen, and Lin Yang. Iterative attention mining for weakly supervised thoracic disease pattern localization in chest x-rays. In *MICCAI*, pp. 589–598. Springer, 2018.

[146] Yuxing Tang, Xiaosong Wang, Adam P Harrison, Le Lu, Jing Xiao, and Ronald M Summers. Attention-guided curriculum learning for weakly supervised classification and localization of thoracic diseases on chest radiographs. In *MLMI Workshop*, pp. 249–258. Springer, 2018.

[147] Sebastian Guendel, Florin C Ghesu, Sasa Grbic, Eli Gibson, Bogdan Georgescu, Andreas Maier, and Dorin Comaniciu. Multi-task learning for chest x-ray abnormality classification on noisy labels. *arXiv preprint arXiv:1905.06362*, 2019.

[148] Li Yao, Jordan Prosky, Eric Poblenz, Ben Covington, and Kevin Lyman. Weakly supervised medical diagnosis and localization from multiple resolutions. *arXiv preprint arXiv:1803.07703*, 2018.

[149] Marie Lodde Jannis Bodden Juliane Aichele Christina Müller-Leisse Bernhard Renger Franz Pfeiffer Manuel Schultheiss, Sebastian A. Schober and Daniela Pfeiffer . A robust convolutional neural network for lung nodule detection in the presence of foreign bodies. In *Scientific Reports*, 2020.

[150] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, Vol. 28, , 2015.

[151] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

[152] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pp. 10781–10790, 2020.

[153] Hao Tang, Daniel R Kim, and Xiaohui Xie. Automated pulmonary nodule detection using 3d deep convolutional neural networks. In *ISBI*, pp. 523–526, 2018.

[154] Jie Mei, Ming-Ming Cheng, Gang Xu, Lan-Ruo Wan, and Huan Zhang. Sanet: A slice-aware network for pulmonary nodule detection. *TPAMI*, 2021.

[155] Ivan William Harsono, Suryadiputra Liawatimena, and Tjeng Wawan Cenggoro. Lung nodule detection and classification from thorax ct-scan using retinanet with transfer learning. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[156] Manuel Schultheiss, Philipp Schmette, Jannis Bodden, Juliane Aichele, Christina Muller-Leisse, Felix G Gassert, Florian T Gassert, Joshua F Gawlitza, Felix C Hofmann, Daniel Sasse, et al. Lung nodule detection in chest x-rays using synthetic ground-truth data comparing cnn-based diagnosis to human performance. *Scientific Reports*, Vol. 11, No. 1, pp. 1–10, 2021.

[157] Xiaoyilei Yang, Shuaijing Xu, Jian Wang, Hao Wu, and Rongfang Bie. Attention mechanism in radiologist-level thorax diseases detection. *Procedia Computer Science*, Vol. 174, pp. 524–529, 2020.

[158] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pp. 2881–2890, 2017.

[159] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *CVPR*, pp. 2736–2746, 2022.

[160] Alexey A Novikov, Dimitrios Lenis, David Major, Jir Hladvka, Maria Wimmer, and Katja Bühler. Fully convolutional architectures for multiclass segmentation in chest radiographs. *TMI*, Vol. 37, No. 8, pp. 1865–1876, 2018.

[161] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pp. 234–241. Springer, 2015.

[162] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[163] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pp. 3431–3440, 2015.

[164] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pp. 3146–3154, 2019.

[165] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pp. 4700–4708, 2017.

[166] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[167] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pp. 6105–6114, 2019.

[168] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pp. 2117–2125, 2017.

[169] Hattori A Yamaguchi J Konishi S Yamamoto Y Takahashi D Matsumura Y akeda T, Manabe S. An automatic image collection system for multicenter clinical studies. *Stud Health Technol Inform.*, 2020.

List of Publications

Journal Publications (related to this thesis)

1. **Bowen Wang**, Liangzhi Li, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, “ Match Them Up: Visually Explainable Few-shot Image Classification, ” *Springer, Applied Intelligence, pp. 1573-7497, 2022/08/27, DOI: 10.1007/s10489-022-04072-4.* (Chapter 4)

2. **Bowen Wang**, Toshihiro Takeda, Kento Sugimoto, Jiahao Zhang, Shoya Wada, Shozo Konishi, Shirou Manabe, Katsuki Okada, Yasushi Matsumura, “ Automatic creation of annotations for chest radiographs based on the positional information extracted from radiographic image reports, ” *Elsevier, Computer Methods and Programs in Biomedicine, vol. 209, pp. 106331, 2021/9/1, DOI: 10.1016/j.cmpb.2021.106331.* (Chapter 5)

International Conference (related to this thesis)

1. **Bowen Wang**, Liangzhi Li, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, “MTUNet: Few-shot image classification with visual explanations,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2021, pp. 2294-2298* (Chapter 4)

2. Liangzhi Li, **Bowen Wang**, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, “SCOUTER: Slot attention-based classifier for explainable image recognition,

” *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1046-1055.* (Chapter 4)

3. **Bowen Wang**, Liangzhi Li, Yuta Nakashima, Takehiro Yamamoto, Hiroaki Ohshima, Yoshiyuki Shoji, Kenro Aihara, Noriko Kando, “ Image Retrieval by Hierarchy-aware Deep Hashing Based on Multi-task Learning ” , *Proceedings of the International Conference on Multimedia Retrieval, 2021, pp. 486–490, DOI: 10.1145/3460426.3463586.* (Chapter 3)

4. **Bowen Wang**, Toshihiro Takeda, Kento Sugimoto, Jiahao Zhang, Shoya Wada, Shozo Konishi, Shirou Manabe, Yasushi Matsumura. “ Semi-Supervised Learning of Nodule Detection in Chest Radiographs. ” *The 11th Asia-Pacific Association for Medical Informatics, 2020, best paper award.* (Chapter 5)

Domestic Conference (related to this thesis)

1. **王 博文**、武田 理宏、杉本 賢人、和田 聖哉、小西 正三、真鍋 史朗、松村 泰志, “ 画像診断レポートの所見位置情報による画像 Bounding Box の自動作成 ” 第 *24* 回日本医療情報学会春季学術大会*, 2020.* (Chapter 5)

Journal Publications (not related to this thesis)

1. **Bowen Wang**, Liangzhi Li, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, Yasushi Yagi, “ Noisy-lstm: Improving temporal awareness for video semantic segmentation, ” *IEEE Access, vol. 9, pp. 46810-46820, 2021/3/22, DOI: 10.1109/ACCESS.2021.3067928*

2. Liangzhi Li, Manisha Verma, **Bowen Wang**, Yuta Nakashima, Ryo Kawasaki, Hajime Nagahara, “Grading the severity of arteriolosclerosis from retinal arterio-venous crossing patterns, ” *PLOS Digital Health*

3. Nanqi Ye, **Bowen Wang**, Michihiro Kita, Ming Xie, Wenyue Cai, “ Urban commerce distribution analysis based on street view and deep learning, ” *IEEE Access, vol. 7, pp. 162841-162849, 2019/11/4, DOI:10.1109/ACCESS.2019.2951294*

International Conference (not related to this thesis)

1. Liyun Zhang, Photchara Ratsamee, **Bowen Wang**, Zhaojie Luo, Yuki Uranishi, Manabu Higashida and Haruo Takemura, “ Panoptic-aware Image-to-Image Translation, ” *IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.*

Awards

1. The 11th Asia-Pacific Association for Medical Informatics 2020, Best Paper Award

2. The 2021 Chinese Government Award for Outstanding Self-financed Students Abroad