HyCo-DeB: Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning for Few-Shot Object Detection

Master Thesis
cs-22-mi-10-07
June 15, 2022

Aalborg University
Software

Summary

Few-shot object detection (FSOD), which is the task of detecting novel objects using very few annotated examples, is a highly desirable feature for vision systems. However, even though it has received significant interest from researchers, it still remains a challenge for modern systems. Research within FSOD has revealed that good feature embeddings are essential for detectors to have a good performance.

Current methods in FSOD leave the available data underutilized. Recent studies in computer vision have shown image data contains hierarchical data that Euclidean space, the common approach for embedding data, cannot properly model. Additionally, most current methods focus on learning each class individually, leaving the information on inter-class and intra-class relations mostly unutilized. FSOD is a data-scarce scenario, where leaving any data unutilized can lead to missed opportunities for significant performance gains. We improve performance by utilizing the previously unutilized data.

An alternative method to Euclidean space is hyperbolic space, which is a space with constant negative curvature and has shown to better capture the latent hierarchies of the data. Hyperbolic space in FSOD is still a relatively new subject that has not been fully explored. Most of the works regarding hyperbolic space have mainly been in Natural language processing or computer vision tasks, such as zero-shot and few-shot classification.

Another approach to improve the feature embedding is using contrastive learning, which improves the embeddings in the feature space by making objects of the same class more compact and objects of different classes more distant from each other. A combination of contrastive learning and hyperbolic space has been seen utilized in recent work for graph representation learning but it is still unexplored in FSOD.

This paper proposes Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning (HyCo-DeB) a novel few-shot object detector that taps into the unexploited potential found in data for FSOD by using a combination of hyperbolic space and contrastive learning. We propose a novel hyperbolic classification head, where extracted features are mapped into hyperbolic space, to better encode the hierarchical features of the data, and during the fine-tuning stage, a contrastive head is also introduced to boost intra-class similarity and inter-class difference. However, adding these two heads results in increased complexity on top of the impact of using transfer-learning, due to object variance causes the adaption to the novel dataset to be difficult. To address this, HyCo-DeB utilizes Baby Learning, which initially reduces the object variance and allows it to increase the learning complexity gradually when transferred to the novel task. The process of Baby Learning is to first fine-tune an initial base trained model on a subset of the available examples, resulting in a trained model that is then used in further fine-tuning on an increased subset, in contrast to the standard fine-tuning stage, which only uses the initial base trained model for each of the subsequent fine-tuning. Since our model is based on the widely used Faster R-CNN, we insert a Gradient Decoupled Layer between the backbone and Region Proposal Network (RPN) and another one between the backbone and RCNN Head to alleviate the conflicts between the class-agnostic RPN and the class-relevant RCNN head. We also utilize a Hyperbolic Embedded Prototypes Using Transformer module, an offline prototype-based classification module, to decouple the classification task from the localization task.

For a fair comparison with existing methods, we evaluate our model using the well-established benchmarks Pascal VOC and Microsoft (MS) COCO. HyCo-DeB achieves state-of-the-art performance on the Pascal VOC benchmark and is on par with state-of-the-art on the MS COCO benchmark. Specifically, HyCo-DeB outperforms the current state-of-the-art on Pascal VOC in 11 of the 15 different settings by up to 3.8% mAP_{50} and achieves second best on the remaining four results. Additionally, a thorough ablation study is conducted to demonstrate the effectiveness of our different additions in combination with each other, which shows that all our additions collectively reach the best performance. The ablation study also includes experiments for the value of our proposed hyperbolic classification head's hyper-parameters, controlling the degree of the hyperbolic space's negative curvature and the embedding dimension. This shows a low negative curvature and keeping the dimension size of its input vector is optimal.

HyCo-DeB: Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning for Few-Shot Object Detection

Tobias Kastbjerg Hauge Nielsen* Alexander Pugholm Jankowski* Anh Tuan Nhu Vu*

Department of Computer Science, Aalborg University
Selma Lagerlöfs Vej 300, 9220 Aalborg East, Denmark
{tkhn17,ajanko17,avu15}@student.aau.dk

Abstract

Detection of novel objects from a few annotated examples, known as few-shot object detection (FSOD), is highly desirable and received significant interest from researchers but remains challenging for modern systems. Research has shown good feature embedding is key to good performance. However, many systems still use Euclidean space, although hyperbolic space better encodes the data's hierarchical information. Another way to optimize feature embeddings is contrastive learning which promotes intra-class similarity and inter-class difference. We propose Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning (HyCo-DeB) a novel few-shot object detector that realizes the unused potential in data for FSOD by optimizing the feature embeddings using hyperbolic space through a novel hyperbolic classification head and a contrastive head. HyCo-DeB addresses the increased complexity of these heads and the transition in transfer-learning by using Baby Learning, allowing it to first transition to the new task, then gradually increase the complexity. Being based on the widely used Faster R-CNN, our model deals with the conflicts of the class-agnostic RPN and the class-relevant RCNN head that shares the same backbone and the conflict in localization and classification by decoupling the modules. Experiments show HyCo-DeB outperforms the existing state-of-the-art on the Pascal VOC benchmark and is on par with state-of-the-art on the MS COCO benchmark.

1. Introduction

Computer vision tasks, such as object detection, have received a lot of progress in recent years. However, the success relies upon the availability of large amounts of annotated data. Generating all the annotations can be labor-intensive, or collecting large amounts of images for the area of interest might not even be possible, or extremely expensive. On the contrary, humans have the ability to comprehend novel concepts and recognize novel objects using only limited examples. This human-like capability to generalize from a few samples is a highly desired quality for computer vision systems, however, there is still a large gap between such systems, known as few-shot object detectors [37, 41, 4, 60], and traditional object detectors [34, 61, 5]. Research within few-shot object detection (FSOD) has highlighted the importance of good feature embeddings in order to achieve good performance for such detectors.

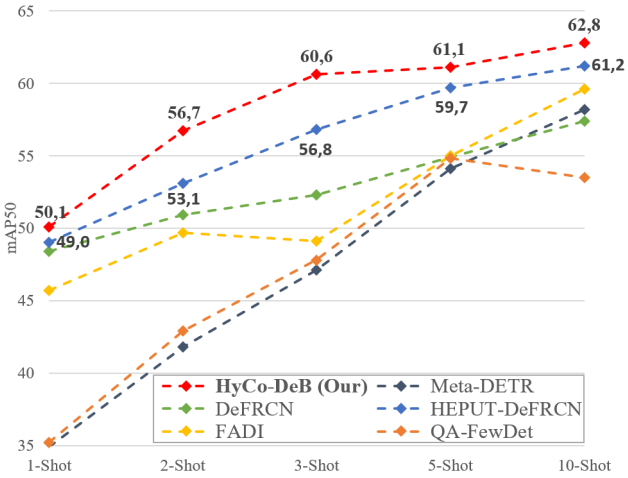


Figure 1. FSOD performance (mAP_{50}) on Pascal VOC's novel set 3 for the different shots. The proposed HyCo-DeB achieves state-of-the-art on all the different shots.

Images contain hierarchical data that most FSOD models do not fully utilize as they use Euclidean space, which has problems modeling hierarchical data, as their embedding space. A better method for encoding hierarchical data is hyperbolic space, a space with constant negative curva-

*Equal contribution.

ture, which has been shown to better model the hierarchical data [25, 36, 22]. Hyperbolic space has already seen successful utilization in computer vision tasks [22, 33, 37, 25], with HEPUT-DeFRCN [37] showcasing its effectiveness in FSOD. Hyperbolic space can better utilize the hierarchical data present in image data to improve image representation leading to better performance.

Most models do not fully utilize the intra-class and inter-class relations, as they focus on learning good representations of the classes individually [20, 51, 58]. This means the models do not take advantage of the information available on how classes are related or different from each other. Contrastive learning is a method that takes advantage of this additional information, by optimizing intra-class similarity and inter-class difference, that is, making objects of the same class more alike and objects of different classes less alike respectively. Contrastive learning has been utilized in self-supervised and semi-supervised computer vision tasks [16, 3], and FSCE [47] has adapted contrastive learning to FSOD. Utilizing contrastive learning makes better use of the information available in the limited data in FSOD.

A common model for FSOD is Faster-RCNN [45], however, it has conflicts between the class-agnostic Region Proposal Network (RPN) and class-relevant RCNN head, as well as the different goals of the classification and localization heads, which have a significant impact on training in a data-scarce scenario like FSOD. Decoupling these tasks alleviate the conflicts and leads to better learning for the model. DeFRCN [41] and HEPUT-DeFRCN [37] utilize this approach and have shown improvements over other methods. Utilizing decoupling, a Faster R-CNN model can learn better and quicker from the limited amount of data in FSOD.

Methods for FSOD fall into two approaches meta-learning, learning to learn, and transfer-learning, which transfers knowledge from one domain to another. Vu *et al.* [49] compared 1-shot, 2-shot, and 3-shot and found transfer-learning models can have worse performance on 2-shot and 3-shot, even though they have more data. They posit this to be due to object variance causing difficulty in adapting to new data. Inspired by how babies learn, they propose a new learning mechanism called Baby Learning, which gradually adapts to instances of new objects by utilizing the previous shots. Baby Learning lets a model learn the variability of novel data by gradually learning from each shot.

We present Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning (HyCo-DeB) to realize the unused potential in the data. HyCo-DeB is a novel few-shot object detector that is based on the Faster R-CNN architecture using transfer-learning. Our model uses a combination of hyperbolic space and contrastive learning to better utilize the hierarchical information and the intra-class and inter-class information already present in the data. That is, it uses a novel hyperbolic classification head to better encode the hierarchical information, and adds a contrastive head during novel fine-tuning to boost inter-class difference and intra-class similarity. To address the conflicts in Faster R-CNN's architecture, and thus increase the possible contribution from its better utilization of the data, HyCo-DeB uses decoupling following [37]. HyCo-DeB also utilizes Baby Learning reducing the impact object variance has and enabling the gradual increase of complexity for k-shot $k > 1$. As our model uses transfer-learning, using Baby Learning also allows it to first transition to the new classes before gradually learning to generalize these. We show that HyCo-DeB outperforms state-of-the-art models and shows dominant performance on Pascal VOC for novel set 3, shown in Figure 1.

Our contributions are summarized as follows:

- We propose HyCo-DeB to optimize the utilization of the limited data in FSOD by using: hyperbolic space, contrastive learning, Baby Learning, and decoupling the Faster R-CNN.
- We conduct a fair comparison with existing methods which shows that our model achieves state-of-the-art performance on the Pascal VOC benchmark and is on par with state-of-the-art on the Microsoft (MS) COCO benchmark.
- To the best of our knowledge, we are the first to optimize the feature representation by combining hyperbolic space with contrastive learning in FSOD.
- We propose a novel hyperbolic classification head for FSOD, to better encode the hierarchical features. This is, to the best of our knowledge, the first time someone utilizes hyperbolic space in training a model for FSOD.
- We conduct a comprehensive ablation study to evaluate the performance of our additions in combination with each other and of the hyper-parameters for our proposed hyperbolic classification head.

2. Related Work

2.1. Object Detection

Object detection is the computer vision task of locating and identifying objects in an image. The most common methods for object detection were previously either two-stage or one-stage detectors. Two-stage detectors, such as the ones based on the R-CNN framework [12, 11, 45, 30, 17], are proposal based, which use a module to first generate region proposals and then perform classification and bounding-box regression. One-stage detectors, such as the popular YOLO series [42, 43, 44, 50], are proposal free, which does not use a module to generate region proposals, but models the detection as a regression problem and uses

a single Convolutional Neural Network (CNN) to directly predict object classes and locations. Typically one-stage detectors have a better inference speed, since they don't have an RPN, but are generally not as accurate compared to two-stage detectors, which are typically slower but reach higher accuracy.

Recently, models have been utilizing Transformers as a backbone, for instance, by using the Swin Transformer [35, 34], or as part of an encoder-decoder architecture, such as DINO [61], and have achieved top performance on the COCO val2007 benchmark.

However, in scenarios where a large amount of annotated data is not available for training, the performance of these object detector models will fall significantly, since they do not have sufficient data to establish a good representation of the object classes' feature space.

2.2. Few-Shot Object Detection

FSOD aims to solve the limited data problem by first training the model on abundant data and then fine-tuning the model on the limited data.

Existing approaches in FSOD fall into one of two paradigms, meta-learning, which learns learning strategies to quicker and better adapt to novel concepts, or transfer-learning, which uses the knowledge gained from training on other similar tasks and transfers it to the novel task. Meta-RCNN [58] is a meta-learner that infers class-specific soft-attention vectors, which are applied to the features in the predictor head to detect or segment objects of a class. FSRW [20] uses a feature reweighting scheme, which takes k-shot samples to create a reweighting vector for target classes and applies it to obtain class-specific features. These early meta-learners focus on learning each class individually, without truly considering the relation between classes. Meta-DETR [60] is a more recent meta-learner that utilizes inter-class relations to reduce misclassification and enhance model generalization, by aggregating query and support features for all classes simultaneously.

LSTD [2] and TFA [51] are early detectors that follow the transfer-learning paradigm. LSTD transfers knowledge from a larger dataset to a smaller dataset by first training a base model using the base classes and then training a novel model using base and novel classes as well as the knowledge gained from training the base model. TFA introduces a two-stage fine-tuning approach wherein in stage one the model is trained on the base classes, and in stage two the box predictor is fine-tuned using both base and novel classes. A more recent approach to transfer-learning is FSCE [47]. FSCE uses contrastive learning to improve on the two-stage fine-tuning approach by better utilizing the information present in the data, that is the intra-class similarity and inter-class difference.

Recently, works like [22] and [25] have shown hyperbolic space outperforms Euclidean space in the classification and object detection tasks. These successes are attributed to the hierarchical structure inherent in hyperbolic space and the exponentially growing distance, which better models properties in the data.

The most recent developments in FSOD include DeFRCN [41], where the architecture of Faster R-CNN [45] is adapted to the FSOD setting and decoupled using a Gradient Decoupled Layer (GDL) and a Prototypical Calibration Block (PCB). These effectively decouple tasks that had conflicting goals in the architecture. HEPUT-DeFRCN [37] further builds on DeFRCN by replacing the PCB with a Hyperbolic Embedded Prototypes Using Transformer (HEPUT) module, which uses a Vision Transformer and hyperbolic space to create better prototypes.

Contrastive learning and hyperbolic space are both approaches to improve the embeddings by using inter-class and intra-class relations and better modeling the data hierarchy, respectively. Both approaches have seen adaption in FSOD in the form of FSCE for contrastive learning and HEPUT-DeFRCN for hyperbolic space, however, the approaches have not been fully explored.

2.3. Contrastive Learning

Contrastive learning is an approach for semi-supervised learning models [28, 56, 64], using limited labeled data with mostly unlabeled data, and self-supervised models [6, 55], using only unlabeled data. Contrastive learning, introduced by Hadsell *et al.* [14], works by dragging positive examples, e.g. augmentations of the same image, together and pushing negative examples, e.g. different images, apart.

Recent work in self-supervised and semi-supervised computer vision has shown great success. MoCo [16] imagines the contrastive task as a dictionary look-up task using a queue and uses a momentum-based update for the key encoder. SimCLR [3] combines components from previous works on contrastive learning into a single unified framework, which uses data augmentations to define the contrastive prediction task. While works on contrastive learning have been focused on self-supervised and semi-supervised learning, it has also seen success in supervised learning. SupCon [21] proposes an extension of the contrastive loss function, which allows for the use of multiple positives per image, using object classes to define positive and negative examples. FSCE [47] adds a contrastive head to help learn contrast-aware embeddings in FSOD, and like SupCon proposes a new loss function called Contrastive Proposal Encoding loss (CPE loss).

Recent work in graph representation learning has utilized a combination of contrastive learning and hyperbolic space to better capture hierarchical information. HCGR [13] is a graph recommender system designed to capture the hierarchical information that other recommender systems fail

to capture. HGCL [32] is a framework for graph representation learning, which implicitly captures the hierarchical structure by passing node representations through multiple hyperbolic layers.

We combine contrastive learning and hyperbolic space to improve the feature representation of HyCo-DeB. By combining both methods we further improve the detector's ability to differentiate between classes. To our knowledge, we are the first to utilize a combination of contrastive learning and hyperbolic space in FSOD.

2.4. Hyperbolic Space

Hyperbolic space is space with constant negative curvature, in contrast to Euclidean space with no curvature and spherical space with positive curvature. In hyperbolic space, the circumference and area of a circle have exponential growth with radius, whereas in the Euclidean space it only has linear and quadratic growth. This matches the exponential growth of the number of leaves seen in tree-like structures with respect to their depth.

Recently, hyperbolic space has received attention in machine learning, due to its properties of modeling data to reflect complex hierarchical relations between data compared to Euclidean and spherical space. Early works have mainly been in Natural language processing (NLP) tasks. This advantage was demonstrated by [36], which introduces an approach to capture latent hierarchies, by embedding the extracted feature into the Poincaré ball model, a representation of the hyperbolic space, and surpassing Euclidean space in terms of generalization ability and representation capacity. Later in [10], hyperbolic versions of multinomial logistic regression (MLR), feed-forward, and recurrent neural networks are introduced, which show to either be on par or outperform their Euclidean versions. Hyperbolic space was first introduced to computer vision in [22] by Khrulkov *et al.* for few-shot classification and person re-identification tasks. Khrulkov *et al.* argue that hierarchical relations between images are common, this resulted in a prototypical network that uses hyperbolic embeddings to capture the underlying hierarchy of the visual data, which showed a substantial boost in performance. Recent works [33, 25] have also found success and showed improved performance in using hyperbolic space compared to Euclidean space. Liu *et al.* [33], employs hyperbolic embedding in the context of zero-shot image classification to preserve hierarchical information and to capture semantic information of the WordNet relations, and image features are projected into hyperbolic space, which is used to perform classification based on the distance to the embeddings. Lang *et al.* [25] introduces a hyperbolic classification head that uses the Lorentz Model, to represent the hyperbolic space, for two-stage, keypoint-based, and transformer-based multi-object detection architectures, which is similar to our use of hyperbolic space, but in the field of closed-set, long-tailed, and zero-shot object detection. A recent work [37] in FSOD, introduces the HEPUT module, an offline prototype-based classification module that uses a ViT to extract image features and embeds them into hyperbolic space resulting in state-of-the-art results on the Pascal VOC Benchmark.

Hyperbolic space has not been fully explored in FSOD. Besides HEPUT-DeFRCN [37], which only uses hyperbolic space to create hyperbolic embedded prototypes, there have been no further works utilizing hyperbolic space to create a classifier in FSOD, leaving the subject understudied. In this work, we employ hyperbolic space in a hyperbolic classification head that uses the Poincaré ball model and also make use of an offline HEPUT module to take advantage of latent hierarchies within data and thereby improve the feature representation.

3. Preliminaries

3.1. Hyperbolic Space

The n-dimensional hyperbolic space $\mathbb{H}^n$ is a homogeneous, simply connected n-dimensional Riemannian manifold with constant sectional negative curvature. A Riemannian manifold (M, g) is a smooth manifold M, defining notion of closeness, with a Riemannian metric g, defining inner products for tangent spaces in the manifold. The negative curvature of hyperbolic spaces means it has different properties than Euclidean space, e.g. hyperbolic space has a hierarchical structure.

A common model to represent hyperbolic space in NLP is the Poincaré ball model, which has also seen use in computer vision tasks. The Poincaré ball model is defined as $(\mathbb{D}^n, g^{\mathbb{D}})$, a n-dimensional manifold $\mathbb{D}^n = \{x \in \mathbb{R}^n : ||x|| < 1\}$ endowed with the Riemannian metric $g^{\mathbb{D}}(x) = \lambda_x^2 g^E$, where $\lambda_x = \frac{2}{1-||x||^2}$ is the conformal factor and $g^E = \mathbf{I}^n$ is the Euclidean metric tensor. The conformal factor preserves the angles between different lines in the manifold, and the Euclidean metric tensor defines the length and angle between tangent vectors. In hyperbolic space, the distance between two points is measured using a geodesic, which is a curve defining the shortest path between points in a Riemannian manifold. The geodesic distance for the model is defined as $d_{\mathbb{D}}(x, y) = arccosh(1 + 2\frac{||x-y||^2}{(1-||x||^2)(1-||y||^2)})$.

Due to the negative curvature, standard Euclidean operations do not work in hyperbolic space. Instead, one can use the formalism of Möbius gyrovector spaces, analogous to Euclidean vector spaces, to generalize standard operations into hyperbolic space. Part of these hyperbolic operations is the hyper-parameter c. Given a Poincaré ball model, c modifies the curvature such that if $c = 1$ one has the Poincaré ball model previously described, i.e. $\mathbb{D}^n = \mathbb{D}_c^n$, while if $c = 0$ one recovers Euclidean space.

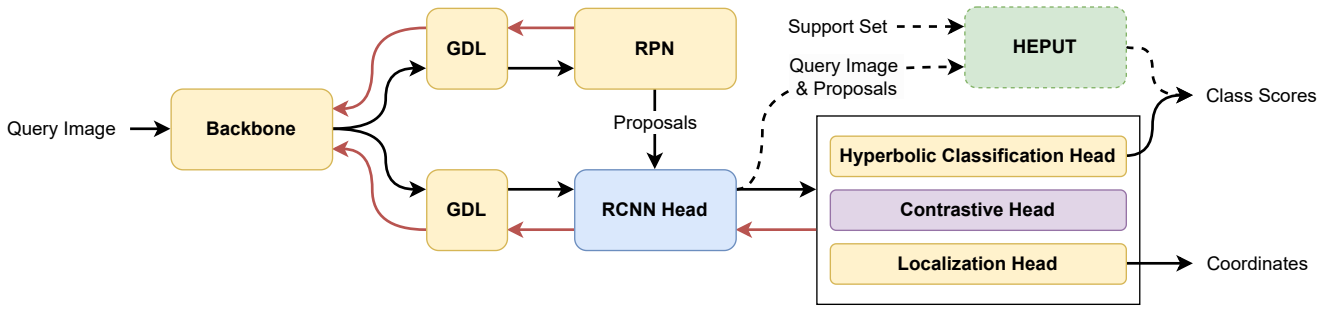


Figure 2. The architecture of HyCo-DeB. The black and red arrows indicate forward and gradient flow respectively. The dashed lines indicate it only takes place in inference. The yellow modules are trainable during base training and fine-tune, whereas the blue is frozen during fine-tune. The purple module is a trainable module only inserted and used during fine-tuning. The green module is an offline block used only during inference.

Möbius addition. Performing addition in the Poincaré ball model is done using the Möbius addition. Given a pair of vectors $x, y \in \mathbb{D}_c^n$, Möbius addition is defined as:

$$x \oplus_c y := \frac{(1 + 2c\langle x, y\rangle + c||y||^2)x + (1 - c||x||^2)y}{1 + 2c\langle x, y\rangle + c^2||x||^2||y||^2}. \quad (1)$$

The exponential map. To map vectors from from tangent space $T_x\mathbb{D}_c^n \cong \mathbb{R}^n$, into hyperbolic space $\mathbb{D}_c^n$, the exponential function is used, allowing for projection of vectors from Euclidean space $\mathbb{R}^n$ into $\mathbb{D}_c^n$.

The exponential map exp_x^c is defined as:

$$exp_x^c(v) := x \oplus_c (tanh(\sqrt{c}\frac{\lambda_x^c||v||}{2})\frac{v}{\sqrt{c}||v||}). \quad (2)$$

3.2. Problem Definition

As in other works [20, 41, 51, 60], we follow the standard problem definition for FSOD. Let the base dataset $\mathcal{D}_{base}$ be a dataset with abundant annotated instances of the base classes $\mathcal{C}_{base}$ and let the novel dataset $\mathcal{D}_{novel}$ be a dataset with limited annotated instances of the novel classes $\mathcal{C}_{novel}$, where $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \varnothing$. Since we operate in the *M*-way *K*-shot object detection setting, the support set for $\mathcal{D}_{novel}$ contains *M* classes with exactly *K* examples for each class. The objective is to leverage the data from $\mathcal{D}_{base}$ to train a robust model and generalize it using $\mathcal{D}_{novel}$, resulting in a final model F_{final} that can classify and localize objects of the $\mathcal{C}_{novel}$ classes. Using a two-stage fine-tuning approach, the following equation summarizes the problem definition:

$$\mathcal{F}_{init} \overset{\mathcal{D}_{base}}{\rightsquigarrow} \mathcal{F}_{base} \overset{\mathcal{D}_{novel}}{\rightsquigarrow} \mathcal{F}_{final} \quad (3)$$

where $\mathcal{F}_{init}$ and $\mathcal{F}_{base}$ is the initial and base trained model respectively, and $\overset{\mathcal{D}_{base}}{\rightsquigarrow}$ and $\overset{\mathcal{D}_{novel}}{\rightsquigarrow}$ represents base training using $\mathcal{D}_{base}$ and novel fine-tuning using $\mathcal{D}_{novel}$ respectively.

4. Hyperbolic and Contrastive Embedding using Decoupling with Baby Learning

Figure 2 presents the architecture of Contrastive Embedding using Decoupling with Baby Learning (HyCo-DeB).

Our model's overall architecture extends the widely used Faster R-CNN [45] architecture which takes an input image and uses a backbone to extract features generating a feature map that represents the image. We insert two GDLs [41], one GDL is placed between the backbone and RPN, and the other GDL between the backbone and RCNN head. The two GDLs are used to decouple the conflict between the class-agnostic RPN and class-relevant RCNN head and will be further described in Section 4.1.

Taking the output from the GDL placed before the RCNN head, the RCNN head combines it with the proposals from the RPN and first employs Region of Interest (RoI) pooling for each of the proposals followed by a CNN to generate fixed-sized feature vectors known as RoI features.

The RoI features are then passed to two heads during base training and three heads during novel fine-tuning: (1) a hyperbolic classification head that performs classification. This head replaces the standard Euclidean classification head, allowing the model to optimize the utilization of the features due to hyperbolic space's ability to capture complex hierarchical relations between data points, this is described in more detail in Figure 5; (2) a localization head that predicts the coordinates; (3) a contrastive head, only used during novel fine-tuning. The contrastive head allows the model to optimize the feature space such that instances of the same class should be more alike, while instances of different classes should be more dissimilar, which is described in more detail in Section 4.3.

During inference, the class scores from the hyperbolic classification head are combined with the class scores from an offline HEPUT module [37] to improve the classification performance as the classification and localization within the model are conflicting tasks, further described in Section 4.1.

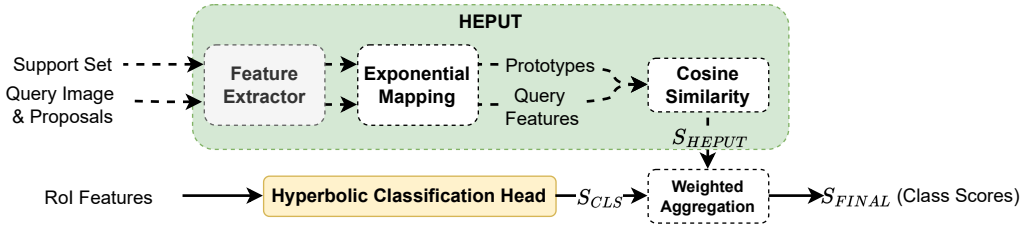


Figure 3. The details of HEPUT. HEPUT is only utilized during inference as indicated by the dashed lines, where the exponential mapping is given by Eq. (2).

We add Baby Learning which gradually exposes the model to higher variability in the training data while using previous knowledge, simplifying the complexity in the beginning, and slowly increasing its generalization ability, which is explained in more detail in Section 4.4.

4.1. Decoupling

We perform decoupling on the model by inserting two GDLs, one between the backbone and the RPN, and one between the backbone and the RCNN head. We also insert a HEPUT module to decouple the classification task from the localization task. The purpose of decoupling our model is to separate conflicts between modules, which negatively affect the training of the model and thus might negatively impact the performance of HyCo-DeB's realization of the unused potential in the data. This decoupling is based on DeFRCN [41] and HEPUT-DeFRCN [37]. We refer to those papers for the complete details on the two decoupling modules.

GDL, as seen in Figure 4, performs multi-stage decoupling on the model, separating conflicts between the class-agnostic RPN and the class-relevant RCNN head. The GDL performs a learnable affine transformation on the feature maps during the forward propagation (the black arrows). During the backward propagation (the red arrows), the GDL performs gradient decoupling by multiplying the gradient from subsequent layers with a decoupling coefficient $\lambda \in [0, 1]$ before passing it to the preceding layer.

The HEPUT module, seen in Figure 3, performs multi-task decoupling, separating conflicts between the classification and localization heads. That is, the position of the object should have no impact on the classification head but should have an impact on the localization head. HEPUT is an offline prototype-based classification module that takes the proposals and query images from the RCNN head as input. In a *M*-way *K*-shot setting HEPUT takes a support set consisting of *M* class labels and *K* images for each *M* class, with one object of the class. Using the support set, HEPUT uses a feature extractor to generate a feature representation of the objects, which are then embedded into hyperbolic space, using the exponential mapping function Eq. (2). HEPUT then builds a hyperbolic prototype bank with one prototype for each of the classes in the support

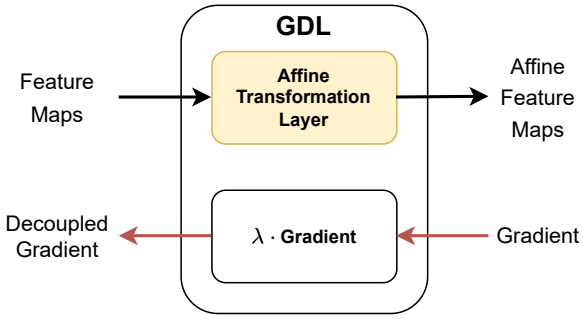


Figure 4. The details of the GDL. The black and red arrows indicate forward and gradient flow respectively, and the yellow module is trainable. λ is the decoupling coefficient.

set. Given a query image and proposals, HEPUT generates query features using the feature extractor followed by exponential mapping Eq. (2). The cosine similarity is then calculated between the hyperbolic query feature and the corresponding hyperbolic prototype. The cosine similarity S_{HEPUT} that HEPUT produces is then combined with the hyperbolic classification head's score S_{CLS} using weighted aggregation to obtain the final classification score S_{FINAL}, defined as:

$$S_{FINAL} = \alpha \cdot S_{CLS} + (1 - \alpha) \cdot S_{HEPUT} \quad (4)$$

where α is the weighted value for aggregation between S_{HEPUT} and S_{CLS}.

4.2. Hyperbolic Classification Head

The hyperbolic classification head, shown in Figure 5, takes the RoI features from the RCNN head and maps them into the hyperbolic space in which it performs classification. The hyperbolic space allows our model to better encode the complex hierarchical structure of the data, in contrast to the more standard approach that performs classification in Euclidean space.

The Poincaré ball model is used as a representation of the hyperbolic space, defined as $\mathbb{D}_c^n = \{x \in \mathbb{R}^n : c||x||^2 < 1, c \geq 0\}$ with a conformal factor of $\lambda_x^c = \frac{2}{1-c||x||^2}$, where c is a hyper-parameter used to adjust the degree of negative curvature.

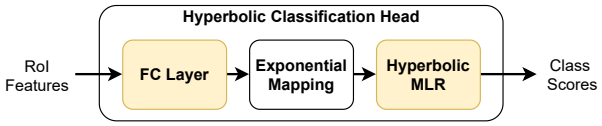


Figure 5. Details of the hyperbolic classification head. The yellow modules are trainable, and the exponential mapping denotes the function given by Eq. (2).

Our model, more specifically, takes the RoI features and first applies a linear layer with a hyper-parameter D to adjust the dimension of the layer. Next, the features are mapped into hyperbolic space using Eq. (2), where we ensure the numerical stability to make the training more stable following [22]. We specifically restrict the norm to not exceed $\frac{1}{\sqrt{c}(1-10^{-3})}$ by clipping the norm after applying an exponential mapping. The model then uses a hyperbolic MLR layer [10] on the hyperbolic features to perform classification, where the Poincaré ball model's dimension is equal to D. The output of the hyperbolic classification head is softmax class scores.

4.3. Contrastive Head

The contrastive head is parallel to the hyperbolic classification head and the localization head and serves to perform contrastive learning in HyCo-DeB. It optimizes the embedding space resulting in a more robust feature representation with a tighter cluster for objects of the same class and increased distance between clusters of different classes.

The contrastive head, seen in Figure 6, first takes the feature vector from the RCNN head and applies a 1-layer multi-layer-perceptron (MLP) to encode the feature vector $x \in \mathcal{R}^{2048}$ into contrastive feature z with dimension D_C, which is a hyper-parameter. This is performed since the similarity between RoI feature vectors can not be measured directly, because a Rectified Linear Unit (ReLU) activation function is applied to the RoI feature, and values are therefore truncated at zero. Afterward, the model calculates the CPE loss, Eq. (5), and uses it to improve the intra-class similarity and the inter-class difference of the object proposals. The contrastive head is based on FSCE [47], and we refer the reader to that paper for the complete details.

Our model only uses the contrastive head during fine-tuning, where the CPE loss is included in the model's loss. The CPE loss is defined as:

$$\mathcal{L}_{CPE} = \frac{1}{N}\sum_{i=1}^{N} f(u_i) \cdot L_{z_i} \tag{5}$$

where $f(u_i)$ controls the consistency of proposals using an Intersection over Union (IoU) threshold, u_i denotes the IoU score with a matched ground-truth bounding box for i-th region proposal, and L_{z_i} is the supervised contrastive loss, defined as:

$$L_{z_i} = \frac{-1}{N_{y_i}-1}\sum_{j=1,j\neq i}^{N} \mathbb{I}\{y_i = y_j\} \cdot log\frac{exp(\tilde{z}_i \cdot \tilde{z}_j/\tau)}{\sum_{k=1}^{N} \mathbb{I}_{k\neq i} \cdot exp(\tilde{z}_i \cdot \tilde{z}_k/\tau)} \tag{6}$$

where N is the number of proposals, τ is the hyper-parameter temperature, $\tilde{z}_i \cdot \tilde{z}_j$ is the cosine similarity between i-th and j-th region proposal, $\tilde{z}$ denotes the normalized features of z, and y_i denotes the label of the ground truth for i-th region proposal.

4.4. Baby Learning

Inspired by [49], HyCo-Deb is fine-tuned using Baby Learning. The idea, shown in Figure 7, behind Baby Learning is to fine-tune a model by training it on a subset of the available examples, taking the trained model and further training it on a larger subset. This process is then repeated, gradually increasing the size of the subset until it contains all the examples in the novel dataset $\mathcal{D}_{novel}$. Each of these subsets contains the same amount of examples for each class, and a larger subset is a superset of the previous subset.

Gradually increasing the amount of data, and reusing existing learned knowledge, allows the model to first adjust to a new concept without much variability in the data. After the model has adapted to the new task, the step-wise increase in the complexity allows the model to better generalize. Increasing the total complexity of the novel dataset $\mathcal{D}_{novel}$ gradually also deals with the extra introduced complexity of adding the novel hyperbolic classification head and a contrastive head to HyCo-DeB.

We train by gradually increasing the size of the subsets, using the subset sizes $\{1, 2, 3, 5, 10, 30\} \leq K$ examples of each class, for a K-shot task. This implies that no Baby Learning is taking place for a 1-shot task. Our model implements Baby Learning, for $K > 1$ by using the previous shot's model to implement the gradually increasing sets. For instance, for 2-shot we use the 1-shot trained model, and for 3-shot we use a 2-shot trained model, and so on.

4.5. Loss Function

The loss function for HyCo-DeB is defined as the following equation:

$$\mathcal{L} = \mathcal{L}_{rpn}(F_{rpn}(\mathbb{G}_{rpn}(F_b(x;\theta_b));\theta_{rpn});y_{rpn}) + \eta \cdot \mathcal{L}_{rcnn}(F_{rcnn}(\mathbb{G}_{rcnn}(F_b(x;\theta_b));\theta_{rcnn});y_{rcnn}) \tag{7}$$

where F_b, F_{rpn}, and F_{rcnn} are the backbone, RPN, and RCNN respectively. The $\mathbb{G}_{rpn}$ and $\mathbb{G}_{rcnn}$ are the two GDLs

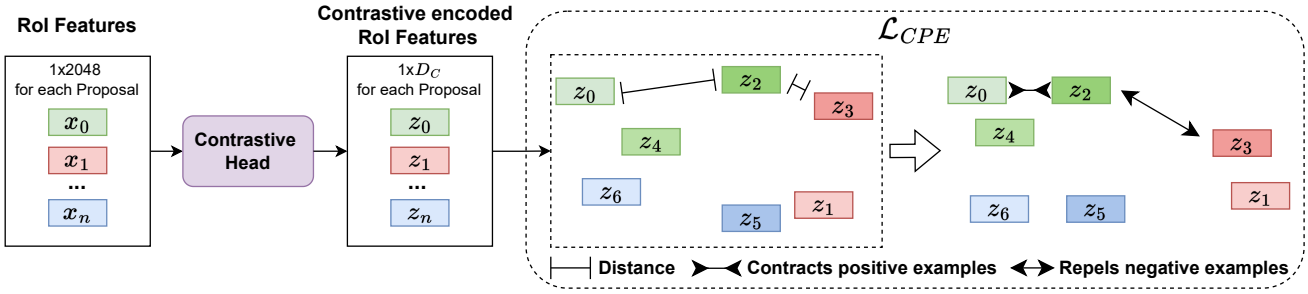
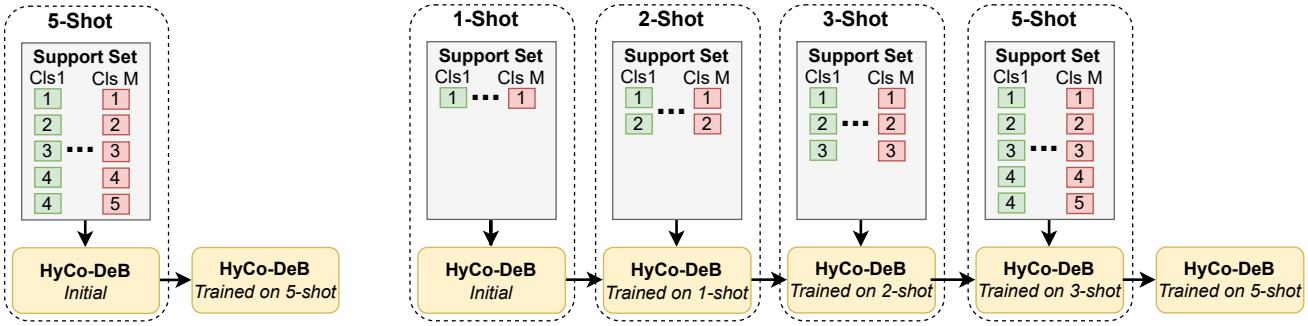


Figure 6. Details of the contrastive branch. The contrastive head takes the RoI features and encodes them into contrastive RoI features. To calculate the loss the distances between objects are measured, where objects of the same class are pulled closer together and objects of different classes are repelled from each other. This results in improving the intra-class similarity and the inter-class difference.



(a) Without Baby Learning (b) With Baby Learning

Figure 7. Fine-tuning without (a) or with Baby Learning (b). In normal fine-tune the model is initialized and trained on the whole support set. In fine-tune with Baby Learning, the model is first trained on a initial model in a 1-shot setting, then this model is retrained on increasing number of shots.

inserted between either the RPN or the RCNN and the backbone, which performs an affine transformation during forward propagation. The y_{rpn} along with y_{rcnn} are ground truths and θ_b, θ_{rpn}, and θ_{rcnn} are learnable parameters for the backbone, RPN, and RCNN respectively.

$\mathcal{L}_{rpn}$ is the loss of the proposals for the localization and classification, where classification is whether or not a proposal is an object. $\mathcal{L}_{rcnn}$ is during base training $\mathcal{L}_{rcnn} = \mathcal{L}_{cls} + \mathcal{L}_{loc}$, while during fine-tuning it is $\mathcal{L}_{rcnn} = \mathcal{L}_{cls} + \mathcal{L}_{loc} + \mathcal{L}_{CPE}$. $\mathcal{L}_{cls}$ and $\mathcal{L}_{loc}$ are the loss of the hyperbolic classification head and localization head respectively, both are regression losses, and $\mathcal{L}_{CPE}$ is the loss of the contrastive head described in Section 4.3. The hyper-parameter η controls the balance between $\mathcal{L}_{rpn}$ and $\mathcal{L}_{rcnn}$.

5. Experiments

5.1. Datasets

For a fair comparison, we use the well-established benchmarks for FSOD and evaluate our model on the Pascal VOC [8, 7] and MS COCO [31] datasets following the standard [20]. Moreover, the reported results of our model are also averaged over 10 runs.

Pascal VOC. We follow the standard setup, dividing the overall 20 classes in Pascal VOC into 15 base classes and 5 novel classes, using the same 3 base/novel splits. For each run, we sample $K = 1, 2, 3, 5, 10$ objects for each novel class, where K is the number of shots. The model is trained on VOC07+12 training and validation datasets and evaluated on the VOC07 test dataset. We first train the model using the base data and then fine-tune on the novel data. For the evaluation, we follow the standard taking the mean Average Precision (mAP) at an IoU threshold of 0.5 for the novel classes, which we refer to as mAP_{50}.

MS COCO. Following the standard approach, out of MS COCO's 80 classes, the 60 classes that are disjoint with Pascal VOC are used as base classes while the remaining 20 are used as novel classes with $K = 1, 2, 3, 5, 10, 30$. The model is evaluated using 5,000 images from the MS COCO 2014 validation set, while the remaining validation set combined with the MS COCO 2014 training set is used for training the model. We first train on the base data and then fine-tune on novel data. The evaluation follows the standard MS COCO evaluation metric, which takes the mAP, for the novel classes, averaged over the IoU thresholds $0.5, 0.55, ..., 0.95$ and we refer to this as mAP.

Method / Shots	Avg. over multiple runs	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW [20]	✗	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
Meta R-CNN [58]	✗	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
FSOD-KT [23]	✗	27.8	41.4	46.2	55.2	56.8	19.8	27.9	38.7	38.9	41.5	29.5	30.6	38.6	43.8	45.7
TFA w/ cos [51]	✗	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
TFA w/cos + Halluc [62]	✗	45.1	44.0	44.7	55.0	55.9	23.2	27.5	35.1	34.9	39.0	30.5	35.1	41.4	49.0	49.3
Retentive R-CNN [9]	✗	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
CORPNs w/ cos [63]	✗	44.4	38.5	46.4	54.1	55.7	25.7	29.5	37.3	36.2	41.3	35.8	41.8	44.6	51.6	49.6
NP-RepMet [59]	✗	37.8	40.3	41.7	47.3	49.4	**41.6**	**43.0**	43.4	47.4	49.1	33.3	38.0	39.8	41.5	44.8
CoRPNs w/ cos + Halluc [62]	✗	47.0	44.9	46.5	54.7	54.7	26.3	31.8	37.4	37.4	41.2	40.4	42.1	43.3	51.4	49.6
MPSR [54]	✗	41.7	42.5	51.4	55.2	61.8	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
cos-FRCN+CGDP+FRCN [29]	✗	40.7	45.1	46.5	57.4	62.4	27.3	31.4	40.8	42.7	46.3	31.2	36.4	43.7	50.1	55.6
CME w/ F-RCNN [27]	✗	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
SRR-FSD [65]	✗	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
FSOD^{up} [53]	✗	43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5
QA-FewDet [15]	✗	42.4	51.9	55.7	62.6	**63.4**	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
FADI [1]	✗	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	55.0	59.6
MetaDet [52]	✓	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
FSDetView [57]	✓	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
TIP [26]	✓	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
DCNet [19]	✓	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
FSCE [47]	✓	32.9	44.0	46.8	52.9	59.7	23.7	30.6	38.4	43.0	48.5	22.6	33.4	39.5	47.3	54.0
Meta-DETR [60]	✓	35.1	49.0	53.2	57.4	62.0	27.9	32.3	38.4	43.2	51.8	34.9	41.8	47.1	54.1	58.2
DeFRCN [41]	✓	53.6	**57.5**	**61.5**	64.1	60.8	30.1	38.1	47.0	**53.3**	47.9	48.4	50.9	52.3	54.9	57.4
HEPUT-DeFRCN [37]	✓	**55.4**	52.3	60.3	**65.7**	**63.4**	35.0	38.3	**47.6**	**54.0**	**52.8**	**49.0**	**53.1**	**56.8**	**59.7**	**61.2**
HyCo-DeB (Our)	✓	**55.6**	**58.3**	**62.0**	**65.1**	**63.8**	**37.0**	**39.3**	**48.6**	**54.0**	**52.5**	**50.1**	**56.7**	**60.6**	**61.1**	**62.8**

Table 1. FSOD performance (mAP50) on Pascal VOC, with the results in **red** and **blue** being the best and second best respectively. We only include the performance not averaged over multiple runs, if there are no reported results over multiple runs.

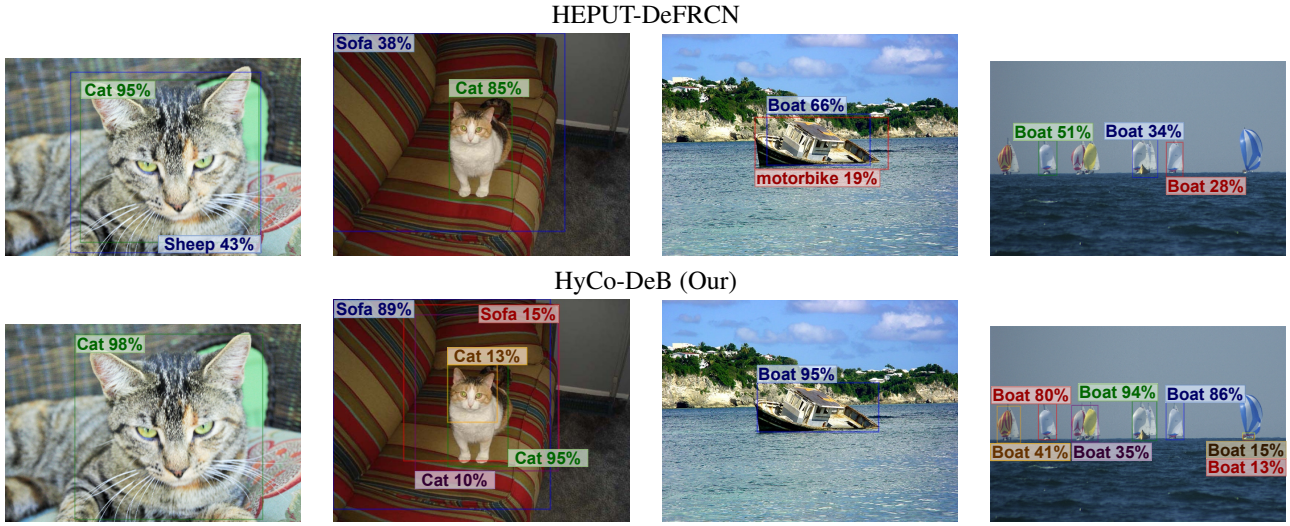


Figure 8. We show examples of predictions on images for HyCo-Deb compared to HEPUT-DeFRCN. We observe that HyCo-Deb, compared to HEPUT-DeFRCN, is more certain in its class prediction, and does not miss-classify. For clarity, we only include proposals with a minimum confidence threshold of 0.1. Images from left to right are modified versions of [38, 24, 39, 48], as they display the predictions.

5.2. Implementation Settings

HyCo-Deb uses Faster R-CNN [45] as the base detector using ResNet-101 [18] pre-trained on ImageNet [46] as the backbone. The network is optimized using SGD with a mini-batch size of 16, a momentum of 0.9, and a weight decay of $5e^{-5}$. A learning rate of 0.02 is used during base training and 0.01 during few-shot fine-tuning for the novel classes. For the HEPUT module, we follow the setup from [37], which sets c to 1. Similarly to [41], we set α to 0.5, λ in the GDL between the backbone and RPN to 0, and the λ in the GDL between the backbone and RCNN head to 0.75 during base training. The λ in the GDL between the backbone and RCNN head is set to 0.01 during fine-tuning for MS COCO, and following [41]'s later optimization we set it to 0.001 for Pascal VOC [40] similar to [37]. For the hyperbolic classification head, we set c to 0.005 and the embedding dimension, D, to 2048, which is the size of the

Method / Shots	Avg. over multiple runs	1	2	3	5	10	30
FSRW [20]	✗	-	-	-	-	5.6	9.1
Meta R-CNN [58]	✗	-	-	-	-	8.7	12.4
TFA w/ cos [51]	✗	-	-	-	-	10.0	13.7
TFA w/cos + Halluc [62]	✗	3.8	5.0	6.9	-	-	-
Retentive R-CNN [9]	✗	-	-	-	8.3	10.5	13.8
CORPNs w/ cos [63]	✗	4.1	5.4	7.1	8.8	10.6	13.9
CoRPNs w/ cos + Halluc [62]	✗	4.4	5.6	7.2	-	-	-
MPSR [54]	✗	-	-	-	-	9.8	14.1
cos-FRCN+CGDP+FRCN [29]	✗	-	-	-	-	11.3	15.1
CME w/ F-RCNN [27]	✗	-	-	-	-	15.1	16.9
SRR-FSD [65]	✗	-	-	-	-	11.3	14.7
FSOD^{up} [53]	✗	-	-	-	-	11.0	15.6
QA-FewDet [15]	✗	4.9	7.6	8.4	9.7	11.6	16.5
FADI [1]	✗	5.7	7.0	8.6	10.1	12.2	16.1
DAnA-FasterRCNN [4]	✗	-	-	-	-	18.6	21.6
MetaDet [52]	✗	-	-	-	-	7.1	11.3
FSDetView [57]	✓	-	-	-	-	12.5	14.7
TIP [26]	✓	-	-	-	-	16.3	18.3
DCNet [19]	✓	-	-	-	-	12.8	18.6
FSCE [47]	✓	-	-	-	-	11.1	15.3
Meta-DETR [60]	✓	7.5	-	**13.5**	15.4	**19.0**	22.2
DeFRCN [41]	✓	**9.3**	**12.9**	**14.8**	**16.1**	18.5	**22.6**
HyCo-DeB (Our)	✓	**9.1**	**13.0**	**14.8**	**15.9**	**18.7**	**22.7**

Table 2. FSOD performance (mAP) on MS COCO, with the results in **red** and **blue** being the best and second best respectively. The performance not averaged over multiple runs is only included if there are no results over multiple runs. A '-' means there is no reported result for that shot.

RCNN head's output vector. For the contrastive head, we use the same settings as [47], setting the dimension of the MLP, D_C, to 128, weights the contrastive loss by 0.5, have a temperature, τ, of 0.2, and uses an IoU threshold of 0.7.

The RCNN head is frozen during novel fine-tuning, and we reset the weights and biases of the hyperbolic classification head and localization head learned during base training when performing novel fine-tuning for 1-shot. We perform novel fine-tuning on k-shot, where $k > 1$, using Baby Learning. Our model's modules are trained jointly in an end-to-end manner.

5.3. Comparison Results

Pascal VOC. Table 1 shows the results of HyCo-DeB's performance on all three Pascal VOC novel class sets, in comparison to other methods. Firstly, we observe that, out of the five different shots for the three different novel sets, that is, from the 15 different settings our model outperforms the state-of-the-art in 11 of these by up to 3.8% mAP_{50} and achieves second best results on the remaining four. However, on two of those settings our model is second best, we are only 0.3% mAP_{50} and 0.6% mAP_{50} from the best, and for the model that outperforms us on the last 2 settings, we significantly outperform it in all of the other 13 settings by up to 20.8% mAP_{50}. This happens as our model succeeds in realizing the unused information in the data, in terms of the object embeddings' inter-class difference and intra-class similarity and also in regards to the hierarchical information. We can also observe that our model achieves the highest result for all shots in novel set 3, surpassing all other models. We also observe that our model outperforms the previous state-of-the-art model HEPUT-DeFRCN, as HyCo-DeB is more certain in its predictions and at distinguishing between classes as can be seen in Figure 8.

MS COCO. Table 2 shows the results of our model's performance on the MS COCO benchmark. We observe that HyCo-DeB's overall performance is on par with the existing state-of-the-art method. Specifically, it achieves the best performance on 2-shot, 3-shot, and 30-shot and second best on 1-shot, 5-shot, and 10-shot. Compared to DeFRCN, our model achieves better results on four of the different shots and is only behind by 0.2 % mAP in the last two shots. Meta-DETR, which outperforms HyCo-DeB on 10-shot, gets outperformed by our model on all the other settings by up to 1.6% mAP.

5.4. Ablation Study

We verify the effectiveness of our design choices through extensive ablation studies. All the results are averaged over five different runs on Pascal VOC's novel set 3.

Effectiveness of the different additions. Table 3 shows the effect of our different additions, specifically, row 1 shows the model without any additions and row 16 shows all additions where the performance is increased by 32.5-37.1% mAP_{50}. Compared to all other possible combinations, we observe that combining all the additions results in the best performance compared to any other use or combinations of the additions, except for 1-shot where the model

Row	Hyperbolic Classification Head	Contrastive Head	Decoupling	Baby Learning	1-Shot	2-Shot	3-Shot	5-Shot	10-Shot
1					15.4	19.7	26.0	28.0	30.3
2	✓				13.2	17.4	19.0	24.7	27.5
3		✓			26.6	25.1	25.6	32.9	32.8
4			✓		48.4	53.0	55.1	59.8	61.6
5				✓	19.1	19.8	21.2	23.7	29.4
6	✓	✓			20.0	21.1	23.2	30.0	29.3
7	✓		✓		46.9	53.2	56.4	58.9	60.6
8	✓			✓	13.1	15.3	21.8	22.7	27.2
9		✓	✓		47.8	50.8	54.4	58.9	61.1
10		✓		✓	23.7	19.4	22.0	25.5	29.6
11			✓	✓	48.2	52.9	**58.5**	59.4	61.4
12	✓	✓	✓		**50.6**	**54.3**	54.7	**59.9**	**62.4**
13	✓	✓		✓	18.5	19.0	20.0	25.4	29.1
14	✓		✓	✓	46.2	50.0	58.4	59.7	61.1
15		✓	✓	✓	48.9	51.4	57.4	57.3	60.4
16	✓	✓	✓	✓	**50.4**	**56.8**	**60.7**	**61.3**	**62.8**

Table 3. Effectiveness of the different additions and all the possible combinations of these. Best and second best results are shown in **red** and **blue** respectively.

without Baby Learning (row 12) is slightly better by 0.2% mAP_{50}, however, is worse for all the other shots. However, Baby Learning is not applied on 1-shot and thus the models for 1-shot are similar, therefore, the performance decrease is due to unlucky worse runs.

Looking at the impact of the decoupling, it is clear to see how big an impact this addition has on the model's total performance. For instance, comparing the model with all additions except decoupling (row 13) to having all additions (row 16), the performance is increased by 31.9-40.7% mAP_{50} when adding the decoupling. This shows that decoupling has a great impact on the model and indicates that it indeed addresses the conflict between the class-agnostic RPN and class-relevant RCNN, and also the conflict between localization and classification.

We observe that a model with decoupling and Baby Learning (row 11) performs better compared to one that also includes the hyperbolic classification head (row 14) or the contrastive head (row 15). However, when using both of these heads (row 16) the performance is increased by 1.4-3.9% mAP_{50} compared to having none of these (row 11). This indicates that the contrastive head's optimization of the object features in the hyperbolic classification head indeed improves the embeddings and that this enrichment is usable for the hyperbolic classification head. This shows the combination of our novel hyperbolic classification head and contrastive head realizes some unused potential in the data.

Table 3 shows the model with all additions except Baby Learning (row 12) to the model that in addition has Baby Learning (row 16), besides the slight decrease in 1-shot, the model shows an increased performance on all the other shots by up to 6.0% mAP_{50}. This shows that performance increases by using Baby Learning to gradually expose the model to more data, and thus greater complexity.

c	1-Shot	2-Shot	3-Shot	5-Shot	10-Shot
0.0005	48.2	51.7	**56.7**	57.3	**60.2**
0.001	48.5	51.7	55.3	57.3	59.7
0.005	**50.4**	**56.8**	**60.7**	**61.3**	**62.8**
0.05	47.8	49.4	55.6	56.7	58.6
0.2	48.5	**51.8**	56.4	**58.6**	59.8
0.5	**50.2**	50.9	55.1	58.5	57.9
1.0	47.7	46.5	52.0	53.9	56.5

Table 4. Performance of different c values in the hyperbolic classification head. The results in **red** and **blue** being the best and second best respectively.

D	1-Shot	2-Shot	3-Shot	5-Shot	10-Shot
32	46.5	49.8	56.5	57.7	59.8
64	46.7	**53.1**	56.8	57.9	59.7
128	46.5	50.0	54.7	57.3	59.2
256	47.1	50.1	54.9	57.6	60.2
512	47.1	50.5	56.7	58.1	59.9
1024	**49.1**	52.5	**57.3**	**59.5**	**61.4**
2048	**50.4**	**56.8**	**60.7**	**61.3**	**62.8**

Table 5. Performance of different dimension sizes in the hyperbolic classification head. Best and second best results are shown in **red** and **blue** respectively.

Ablation for hyperbolic classification head's hyperparameters. The hyper-parameter c is used to adjust the degree of negative curvature for the hyperbolic space in the hyperbolic classification head. We perform an ablation study of different c values to explore the influence of the negative curvature, which is shown in Table 4. We observe that the lower c value, and thus a lower degree of negative curvature, the better until it reaches beyond 0.005, where the performance drops. When going below a c value of 0.005 the performance drops by 1.9-5.4% mAP_{50} depending on the shot and c value. Using a c value of 0.005 outperforms the other c values by 0.2-10.3% mAP_{50}.

We also perform an ablation study on the dimension of the hyperbolic classification head, that is, its linear layer and the Poincaré ball model's dimension, D, to examine the impact of the size of the dimension, and if reducing it from the size of the RoI features, which have a dimension of 2048, improve the model's performance. From the results, shown in Table 5, the performance in general increases as the dimension increase, and the model achieves the best performance at a dimension of 2048, outperforming the other dimensions' different shots by 1.3-7.0% mAP_{50}.

6. Conclusion

This paper presents HyCo-DeB to tap into the unused potential in the data for FSOD. This is achieved by using a combination of hyperbolic space's property to better encode the hierarchical information of the data, compared to Euclidean space, and contrastive learning's ability to optimize the embedding's intra-class similarity and inter-class difference. As the model is based on the Faster R-CNN architecture it applies decoupling to address the inherent conflicts of the different modules. Besides this, it also uses Baby Learning to cope with the transition to novel data in the transfer-learning approach and the increased complexity of the hyperbolic classification head and contrastive head, allowing it to first transition to the new classes and then gradually increase the total complexity. We show that HyCo-DeB achieves state-of-the-art performance on the Pascal VOC benchmark and is on par with state-of-the-art on the MS COCO benchmark.

Acknowledgement

The authors would like to thank Assistant Professor Jilin Hu for providing valuable feedback and supervision throughout the project, and CLAAUDIA for making hardware for training and testing available.

References

[1] Y. Cao, J. Wang, Y. Jin, T. Wu, K. Chen, Z. Liu, and D. Lin. Few-shot object detection via association and discrimination. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] H. Chen, Y. Wang, G. Wang, and Y. Qiao. Lstd: A low-shot transfer detector for object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 2836–2843, 2018.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[4] T.-I. Chen, Y.-C. Liu, H.-T. Su, Y.-C. Chang, Y.-H. Lin, J.-F. Yeh, W.-C. Chen, and W. Hsu. Dual-awareness attention for few-shot object detection. *IEEE Transactions on Multimedia*, 2021.

[5] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2021.

[6] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[9] Z. Fan, Y. Ma, Z. Li, and J. Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4527–4536, 2021.

[10] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, pages 5345–5355, 2018.

[11] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[13] N. Guo, X. Liu, S. Li, Q. Ma, Y. Zhao, B. Han, L. Zheng, K. Gao, and X. Guo. Hcgr: Hyperbolic contrastive graph representation learning for session-based recommendation. *arXiv preprint arXiv:2107.05366*, 2021.

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[15] G. Han, Y. He, S. Huang, J. Ma, and S.-F. Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3263–3272, 2021.

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[19] H. Hu, S. Bai, A. Li, J. Cui, and L. Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10185–10194, 2021.

[20] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[21] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[22] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020.

[23] G. Kim, H.-G. Jung, and S.-W. Lee. Few-shot object detection via knowledge transfer. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3564–3569. IEEE, 2020.

[24] P. Knittel. Fat cat, 2005, `https://www.flickr.com/photos/pknitty86/495430515/` (Accessed: 2022-06-13).

[25] C. Lang, A. Braun, and A. Valada. On hyperbolic embeddings in 2d object detection. *arXiv preprint arXiv:2203.08049*, 2022.

[26] A. Li and Z. Li. Transformation invariant few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3094–3102, 2021.

[27] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2021.

[28] J. Li, C. Xiong, and S. C. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021.

[29] Y. Li, H. Zhu, Y. Cheng, W. Wang, C. S. Teo, C. Xiang, P. Vadakkepat, and T. H. Lee. Few-shot object detection via classification refinement and distractor retreatment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15395–15403, 2021.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[32] J. Liu, M. Yang, M. Zhou, S. Feng, and P. Fournier-Viger. Enhancing hyperbolic graph embeddings via contrastive learning. *arXiv preprint arXiv:2201.08554*, 2022.

[33] S. Liu, J. Chen, L. Pan, C.-W. Ngo, T.-S. Chua, and Y.-G. Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9273–9281, 2020.

[34] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021.

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[36] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, 30:6338–6347, 2017.

[37] T. K. H. Nielsen, A. P. Jankowski, and A. T. N. Vu. HEPUT-DeFRCN: Hyperbolic embedded prototypes using transformer in defrcn. *Aalborg University Project Library*, 2022.

[38] M. O'Connor. Cat 9426, 2015, `https://www.flickr.com/photos/97477873@N00/19840496468/` (Accessed: 2022-06-13).

[39] polanri.com. Boat, 2001, `https://www.flickr.com/photos/polanri/64541277/` (Accessed: 2022-06-13).

[40] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang. Defrcn, 2021, `https://github.com/er-muyue/DeFRCN` (Accessed: 2022-04-28).

[41] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8681–8690, 2021.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[43] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[44] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[45] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[47] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7352–7362, 2021.

[48] W. Tennyson. Spinnaker sailing, 2008, `https://www.flickr.com/photos/westennyson/2273058237/` (Accessed: 2022-06-13).

[49] A.-K. N. Vu, N.-D. Nguyen, K.-D. Nguyen, V.-T. Nguyen, T. D. Ngo, T.-T. Do, and T. V. Nguyen. Few-shot object detection via baby learning. *Image and Vision Computing*, 120:104398, 2022.

[50] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13029–13038, 2021.

[51] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, pages 9919–9928, 2020.

[52] Y.-X. Wang, D. Ramanan, and M. Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9925–9934, 2019.

[53] A. Wu, Y. Han, L. Zhu, and Y. Yang. Universal-prototype enhancing for few-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9567–9576, 2021.

[54] J. Wu, S. Liu, D. Huang, and Y. Wang. Multi-scale positive sample refinement for few-shot object detection. In *European Conference on Computer Vision*, pages 456–472, 2020.

[55] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[56] A. Xiao, C. Fuegen, and A. Mohamed. Contrastive semi-supervised learning for asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3870–3874. IEEE, 2021.

[57] Y. Xiao and R. Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision*, pages 192–210. Springer, 2020.

[58] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9577–9586, 2019.

[59] Y. Yang, F. Wei, M. Shi, and G. Li. Restoring negative information in few-shot object detection. In *International Conference on Neural Information Processing Systems*, 2020.

[60] G. Zhang, Z. Luo, K. Cui, and S. Lu. Meta-detr: Image-level few-shot object detection with inter-class correlation exploitation. *arXiv preprint arXiv:2103.11731*, 2021.

[61] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[62] W. Zhang and Y.-X. Wang. Hallucination improves few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13008–13017, 2021.

[63] W. Zhang, Y.-X. Wang, and D. A. Forsyth. Cooperating rpn's improve few-shot object detection. *arXiv preprint arXiv:2011.10142*, 2020.

[64] Y. Zhou, H. Xu, W. Zhang, B. Gao, and P.-A. Heng. C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7036–7045, 2021.

[65] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8782–8791, 2021.