Few-Shot Object Detection Using Two Stage Fine Tuning Approach with Data Augmentation

Vishwa M. Badachi[a], Shreya P. Inamdar[b], Fardeen Vaddo[c], Sai Satya B. V.[d], Uday Kulkarni and Shashank Hegde

School of Computer Science and Engineering, KLE Technological University, Vidya Nagar, Hubballi, India

Keywords: Few-Shot Object Detection, Cutout, Detectron2, MS COCO, Data Augmentation, ResNet, Two-Stage Fine-Tuning.

Abstract: Few-Shot Object Detection (FSOD) is a specialized area within object detection that focuses on the task of identifying and localizing objects from unseen classes using only a small number of labeled examples. This is particularly important when data collection and labeling are expensive or impractical. To address this challenge, a novel two-stage fine-tuning approach combined with cutout data augmentation to improve both detection accuracy and generalization is proposed. The proposed method uses the Detectron2, a popular open-source library for object detection. The training process is divided into two stages to improve the model's performance. To address the challenges associated with the limited availability of labeled examples, the method incorporates cutout data augmentation. Cutout augmentation involves masking random rectangular regions within training images. This augmentation technique introduces additional variation in the training data, enabling the model to focus on the important features of objects rather than overfitting to specific patterns or regions, leading to improved detection performance, especially in data-scarce scenarios. The performance of the proposed method was evaluated using the COCO dataset, a widely recognized benchmark in object detection research. Experimental results highlighted the performance gains achieved by the proposed method. Specifically, for the 10-shot setting, where only 10 labeled examples per novel class were available, the method achieved an Average Precision (AP) score of 15.7 at a high Intersection over the Union (IoU) threshold of 0.75 (AP75). The performance of the proposed methodology shows 18.8% relative improvement over the previous state-of-the-art method, demonstrating the effectiveness of the two-stage fine-tuning process combined with cutout augmentation. The proposed method tries address some of the key limitations of existing FSOD approaches.

1 INTRODUCTION

Machine learning systems have made significant progress in recent years, but training models to recognize new things with limited labeled data remains a major challenge. Few-Shot Object Detection is a computer vision task aimed at detecting objects in images using only a limited amount of training data. The objective is to train a model on a small number of examples for each object class and enable it to identify those objects in new images. While humans, including young children, can identify new objects with minimal guidance, achieving this level of generalization in machines is complex. Few-Shot Learning (FSL)(Parnami and Lee, 2022) aims to solve this problem by enabling models to generalize from only a few examples. While FSL has shown good results in image classification, applying it to object detection is more difficult due to the need not just to identify object types but also to locate them within images.

Recent approaches to FSOD have often combined meta-learning techniques with existing object detection frameworks, such as in Meta Region Based Convolutional Neural Networks (R-CNN) (Yan et al., 2019). However, comparing these methods can be challenging, as different studies use different evaluation criteria, making it difficult to determine which approach is better. Another notable approach, FSDetView (Fan et al., 2020), incorporates advanced feature alignment techniques to improve detection ac-

[a] https://orcid.org/0009-0004-0747-6455
[b] https://orcid.org/0009-0006-3160-428X
[c] https://orcid.org/0009-0006-6931-2952
[d] https://orcid.org/0009-0002-3802-1256

M. Badachi, V., Inamadar, S. P., Vaddo, F., B. V., S. S., Kulkarni, U. and Hegde, S.
Few-Shot Object Detection Using Two Stage Fine Tuning Approach with Data Augmentation.
DOI: 10.5220/0013613000004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 3*, pages 238-246
ISBN: 978-989-758-763-4

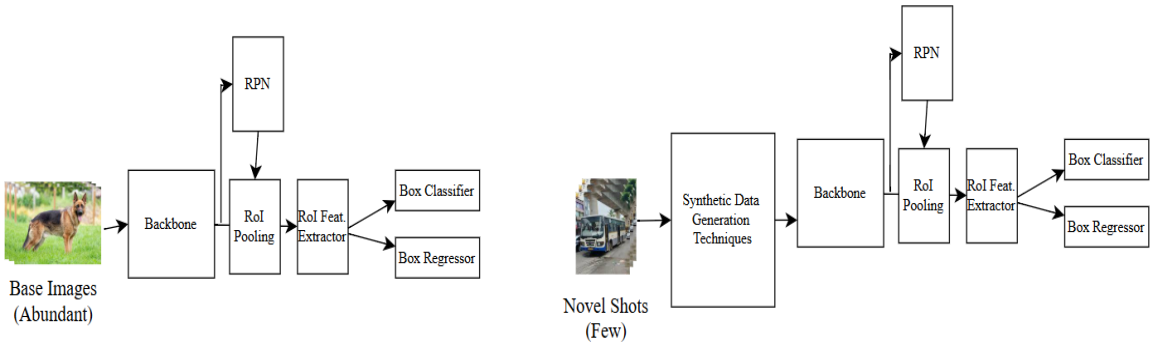


Figure 1: Workflow of proposed methodology: Normal training of base class images in stage 1 continued with training of novel classes with data augmentation and learning of feature extractor frozen in stage 2.

curacy in data-scarce scenarios. Building on the prior fine-tuning-based approach proposed in the Frustratingly Simple 'Few-Shot Object Detection' (Wang et al., 2020), the paper proposes a FSOD method that incorporates advanced data augmentation during the fine-tuning phase. Data augmentation, which involves creating modified versions of existing images through techniques like flipping, cropping, and adding noise (Cubuk et al., 2019), effectively expands the training set. This helps the model learn to recognize objects with variations in appearance, lighting, and positioning, which in turn improves detection and localization capabilities for new object classes.

The approach consists of two stages: first, training an object detection model (e.g., Detectron2 (Y et al., 2019)) on data-rich base classes, and second, fine-tuning the model on data-scarce novel classes using data augmentation. The flow of method is provided in Fig 1. This approach improves the model's ability to generalize to unseen classes while maintaining important features learned in the first stage.

The approach is tested with MS COCO dataset using 80 base classes for backbone training. The model showed an AP score improvement, specifically for the 10 shot setting. The approach achieved an AP score of 15.7 at a high IoU threshold of 0.75 (AP75). The performance of the proposed methodology shows 18.8% relative improvement over the previous state-of-the-art method showcasing the robustness of the two-stage fine-tuning process combined with cutout augmentation.

The structure of the paper is as follows. Section 2 is the background study, including an overview of FSOD research, its impact and a brief discussion on the object detection model, Detectron2. Section 3 provides a description detailing the two-stage fine-tuning process and emphasizes the role of data augmentation. Section 4 presents the results obtained. The final section offers conclusions and areas of improvement.

2 BACKGROUND STUDY

FSOD aims to detect objects belonging to novel classes using only a few labeled examples, a problem of increasing relevance in real-world scenarios where labeled data for every class is scarce or expensive to obtain. This challenge blends the tasks of object detection and few-shot learning, requiring models to not only generalize effectively to unseen classes but also maintain performance on previously seen classes. Traditional object detection models, such as Faster RCNN(Ren et al., 2016) and YOLO(Liu et al., 2018), excel when trained on large, labeled datasets. These models rely on the availability of data to learn features for classifying and locating objects. However, acquiring such datasets for every object class is infeasible in many practical situations, especially for rare categories. FSOD addresses this limitation by learning from a large dataset of base classes and transferring it to detect novel classes with minimal labeled data.

A typical FSOD framework follows a two-stage approach(Wang et al., 2020). The first stage involves training a model with a large dataset of base classes to generalize features. This involves both the feature extractor and box predictor using a combined loss function that helps the model classify and localize. The objective is to extract strong, transferable features while achieving high detection performance on base classes. The second stage involves the fine-tuning process of the trained model using a balanced dataset that includes both base and novel classes. During this stage, the backbone often remains frozen to preserve knowledge from the base classes, while the classification layers or the box predictor are fine-tuned. The

inclusion of the novel classes helps the model adapt to new categories while retaining base class knowledge.

However, FSOD faces several challenges that make it distinct from other conventional object detection methods. The limited label data is often insufficient to represent the diversity of the class, leading to overfitting and poor generalization or models trained on base classes may struggle to recognize novel objects, as their features are predominantly tuned for base categories. Another challenge faced by FSOD is that the objects in real-world scenarios often vary in size, appearance and context, which requires a robust feature extraction ability.

The key methodologies are broadly categorized into three approaches which are described in the subsections A,B and C.

2.1 Data-Oriented Approaches

Data-oriented approaches(Lin et al., 2023) mitigate the data scarcity problem by generating or augmenting training samples. Models like Stable Diffusion(Jian et al., 2023) create synthetic images by varying input text prompts, introducing diversity into the dataset. Techniques such as spatial transformations (rotation, scaling, flipping) and context-aware augmentations help simulate the variability often absent in small datasets.

2.2 Model-Oriented Approaches

Model-oriented approaches (Guan et al., 2024) and (Han and Lim, 2024) deal with modifying object detection frameworks to enhance their ability to generalize to novel classes. Methods such as Consine Similarity-Based Classifiers(Rezaeianaran et al., 2021) replace traditional fully connected layers to reduce intra-class variations and improve the generalization of unseen classes. Components like the Local Feature Enhancement Module (LFEM)(Liu et al., 2023a) and Global Cross-Attention Network (GCAN)(Hou et al., 2019) improve feature extraction by focusing on critical local features and incorporating contextual information. Attention-based networks and class-specific prototypes are used to better differentiate between base and novel classes in feature space.

2.3 Algorithm-Oriented Approaches

Algorithm-oriented approaches(Liu et al., 2023b) refine the training process to improve adaptation and retention. Techniques such as meta-learning, contrastive learning and incremental learning help in adapting to new classes by aligning features of similar classes and separating dissimilar ones while retaining the knowledge of base classes.

Recent FSOD advancements include Meta-Detection Heads (Han and Lim, 2024) which incorporate losses which help enhance detection accuracy by focusing on prediction errors. Multiscale Feature Extraction(Wang et al., 2022) addresses the challenge of detecting objects in multiple resolutions. Meanwhile, Synthetic Instance Augmentation(Lin et al., 2023) focuses on improving the training data to increase diversity and strengthen the novel instances. Together these approaches improve FSOD.

In the following sections, we will combine different types of approaches and components such as data augmentation, algorithm-oriented approaches and Detectron2 architecture to improve FSOD.

2.4 Detectron2 Architecture Description

Detectron2, created by Facebook AI Research (FAIR), is an advanced and flexible framework for computer vision tasks like object detection, instance segmentation, and key point detection. Based on components used in earlier work like Mask R-CNN(He et al., 2017) and Faster R-CNN(Ren et al., 2016). Built on PyTorch, it provides modules of components and a user-friendly system, making it easy for researchers and developers to customize and experiment with computer vision models. The components of the model are Backbone, Regional Proposal Networks, ROI Heads and Detection Heads.

The working and structure of each of these components is given below:

2.4.1 Backbone

The backbone is the part of the architecture responsible for extracting features from input images. Detectron2 (Y et al., 2019) supports well-known backbones such as ResNet and ResNeXt (He et al., 2016). These models process the input image and produce hierarchical feature maps that represent the image at different levels of detail. To enhance its ability to detect objects of various sizes, the framework integrates a Feature Pyramid Network (FPN) (Lin et al., 2017) with the backbone. The FPN combines feature maps from different layers, ensuring the model captures both fine details (for small objects) and broader patterns (for larger objects).

2.4.2 Regional Proposal Network

The Regional Proposal Network[RPN] (He et al., 2016) is a key component of Detectron2, tasked with

identifying regions in an image that are likely to contain objects. It works by sliding over the feature maps from the backbone and predicting the objectness scores which are the likelihood that a region contains an object and anchor boxes which are predefined bounding boxes of various shapes and sizes to provide rough object locations.

The RPN helps narrow down the search space by proposing a manageable number of regions, making the detection process more efficient.

2.4.3 ROI Heads

The Region of Interest (ROI) heads refine the regions proposed by providing likelihood of an object being present and rough object locations. ROI Heads and Pooling were introduced with Fast R-CNN (Ren et al., 2016). These heads extract features from the proposed regions using ROI Align, a method that ensures accurate spatial alignment of features. The key component of ROI heads is, Bounding Box Head, this head fine-tunes the location of the proposed regions and assigns them a class label.

2.4.4 Detection Heads

The detection heads finalize predictions based on the features processed by the ROI heads by predicting object classes and refined object boundaries. The outputs include, object classes which are probabilities of the detected objects belonging to specific categories and bounding boxes are refined coordinates for the object's locations. These outputs enable Detectron2 to perform accurate object detection. In the following section, we shall explain the proposed methodology in detail and validate the methodology using dataset.

3 PROPOSED METHODOLOGY

This section explains the proposed methodology for FSOD. The method divides the training mainly into two stages, namely Backbone Training (stage 1) and Fine Tuning (stage 2). In Stage 1 normal training for images takes place in Detectron2. Stage 2 will have a sub-stage of data augmentation before an image enters the learning architecture and then by freezing the learning of Backbone i.e. the ResNet-based feature extractor an input of the augmented rare class images for training is given.

3.1 Backbone Training (Stage 1)

In Backbone training stage, the learning begins with inputting a set of training images I into a backbone network, typically a deep convolutional neural network such as ResNet(He et al., 2016).

Algorithm 1: Backbone for Feature Extraction.

Data: Input Image $I \in \mathbb{R}^{H \times W \times 3}$
Result: Multi-scale feature maps $\{P_2, P_3, P_4, P_5\}$

Stage 1: Initial Convolution and Pooling
$f_1(I) =$
$\text{MaxPool3x3}(\text{ReLU}(\text{BatchNorm}(\text{Conv3x3}(I))))$

Stage 2 to 5: Residual Blocks
for *each stage* $i \in \{2,3,4,5\}$ **do**
  $f_i(X) = \text{ResBlock}_1(X) \rightarrow \text{ResBlock}_2(X) \rightarrow \cdots \rightarrow \text{ResBlock}_{n_i}(X)$
  for *each residual block ResBlock*(X) **do**
    $\text{ResBlock}(X) = X +$
    $\text{Conv3x3}(\text{ReLU}(\text{BatchNorm}(\text{Conv3x3}(X))))$
  if *stage* $i \in \{3,4,5\}$ **then**
    X (downsampled) $=$
    $\text{Conv1x1}(X)$ with stride 2

Stage Outputs:
$C_2 = f_2(f_1(I))$
$C_3 = f_3(C_2)$
$C_4 = f_4(C_3)$
$C_5 = f_5(C_4)$

Feature Pyramid Network (FPN) with ResNet:
$C_2 \leftarrow f_2(f_1(I))$
$C_3 \leftarrow f_3(C_2)$
$C_4 \leftarrow f_4(C_3)$
$C_5 \leftarrow f_5(C_4)$

Top-Down Pathway:
$P_5 \leftarrow g_1(C_5)$ `// g1 represents a` 1×1 `convolution layer`
$P_4 \leftarrow g_1(C_4) + \text{Upsample}(P_5)$
$P_3 \leftarrow g_1(C_3) + \text{Upsample}(P_4)$
$P_2 \leftarrow g_1(C_2) + \text{Upsample}(P_3)$

Feature Smoothing:
for *each* $P_i \in \{P_2, P_3, P_4, P_5\}$ **do**
  $P_i \leftarrow g_3(P_i)$
return $\{P_2, P_3, P_4, P_5\}$

The backbone network processes each image and outputs multi-scale feature maps F. These feature maps are representations F(x,y) capturing pixel-level relationships and visual semantics across different layers. The feature extraction step can be mathematically represented as:

$$F = \text{Backbone}(I) \tag{1}$$

The algorithm for working of backbone referred in Equation 1 is provided in Algorithm 1. The fea-

ture maps F are then fed into a Region Proposal Network (RPN), which predicts regions likely to contain objects. The RPN slides a small convolutional window over F and generates anchor boxes at each position. For each anchor, the RPN outputs two key elements: objectness score s(a) and bounding box regression offsets t(a) that refine the anchor a, refer Algorithm 2. This can be formalized as shown in Equation 2:

$$s(a), t(a) = \text{RPN}(F) \tag{2}$$

Algorithm 2: Region Proposal Network (RPN) for Object Detection.

Data: Input feature maps F from an intermediate layer of the network
Result: Region proposals with associated objectness scores

Step 1: Convolutional Layers
$X_1 \leftarrow \text{Conv3x3}(F)$
$X_2 \leftarrow \text{Conv1x1}(X_1)$
$X_3 \leftarrow \text{Conv1x1}(X_1)$

Step 2: Region Proposals
$t(a) \leftarrow$ Bounding box proposals from X_2
$s(a) \leftarrow$ Objectness scores from X_3

Step 3: Output Region Proposals
return $s(a), t(a)$

Next, the generated proposals are passed through an ROI Pooling layer, which standardizes variable-sized ROIs into fixed-size feature maps. Opertaions of ROI are provided in Equation 3 and this operation ensures that the input to the next layers remains uniform. If an ROI is defined by coordinates (x_1, y_1, x_2, y_2) the ROI Pooling layer partitions it into equal-sized bins and applies max pooling within each bin to create a fixed-size output, mathematically represented as:

$$\text{ROI Pool}(F(x_1 : x_2, y_1 : y_2)) \rightarrow \text{fixed-size feature map} \tag{3}$$

These pooled features are processed by the ROI feature extractor R_{feat} referred in Equation 4 and Equation 5, which further refines them to enhance their differentiability capacity:

$$f_{ROI} = \text{R}_{feat}(\text{ROI Pool}(F, R)) \tag{4}$$

$$p(c \mid f_{\text{ROI}}) = \text{softmax}(W_c f_{\text{ROI}} + b_c) \tag{5}$$

The output features f_{ROI} in Algorithm 3 are split and fed into two branches: a box classifier and a box regressor. The box classifier uses a softmax function to predict the class probabilities $p(c \mid f_{\text{ROI}})$ for each class c, given the feature f_{ROI}. The box regressor predicts bounding box adjustments $\Delta x, \Delta y, \Delta w, \Delta h$ to refine the proposal coordinates, provided in Equation 6.

Algorithm 3: ROI Feature Extractor.

Data: Input feature maps F from the convolutional layers of the network
Data: Region proposals R_1 from the RPN, which are the proposed bounding boxes
Result: Extracted ROI features

Step 1: ROI Pooling
$\text{ROIs} \leftarrow \text{ROIPool}(F, R_1)$

Step 2: Feature Aggregation
$F_{\text{ROI}} \leftarrow \text{Flatten}(\text{ROIs})$

Step 3: Fully Connected Layer
$F_{\text{FC}} \leftarrow \text{FullyConnected}(F_{\text{ROI}})$

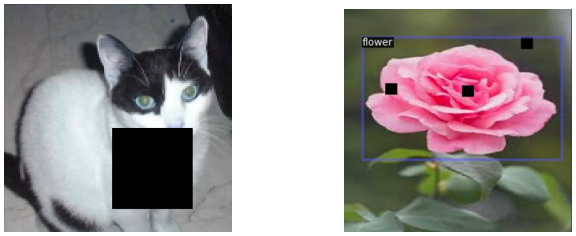
Step 4: Output ROI Features
return F_{FC}

$$\text{Bounding Box} = (x, y, w, h) + (\Delta x, \Delta y, \Delta w, \Delta h) \tag{6}$$

This mathematical workflow supports precise detection by identifying and classifying ROIs while refining their spatial boundaries, laying the foundation for subsequent training and optimization.

3.2 Data Augmentation (Cutout)

Cutout (DeVries, 2017) is a simple augmentation method used to overcome challenges such as over fitting and limited data diversity. Images after augmentation look like the images in Fig 2. The technique works in the following way:



(a) Cat (b) Flower

Figure 2: Images of cat and flower after applying cutout.

Sampling an Image: Select an images, x, from the dataset with its corresponding label, y.

Cutting a Random Patch: A rectangular region R is randomly selected within x. The size and position of R are determined by sampling its coordinates

(r_x, r_y, r_w, r_h), where r_x and r_y are the top-left corner, and r_w and r_h are the width and height of the patch.

Masking the Region: Replace the pixel values within R with a constant value, such as black (0), to create the augmented image x_{cutout} refer Equation 7.

$$x_{\text{mix}}(i,j) = \begin{cases} 0, & \text{if } (i,j) \in R \\ x(i,j), & \text{otherwise} \end{cases} \tag{7}$$

Training: Train the model using the augmented images and their original labels. The objective function, such as cross-entropy, is computed based on the original label, y.

Algorithm 4: Cutout Data Augmentation.

Data: Dataset $D = \{(x_i, y_i)\}$, where x_i is the image and y_i is the label
Data: Patch size range: $(\text{min\_width}, \text{max\_width})$, $(\text{min\_height}, \text{max\_height})$
Data: Image dimensions: W, H
Data: Constant fill value F (e.g., 0 for black or mean pixel value)
Result: Augmented dataset D_{cutout}

Initialize empty dataset $D_{\text{cutout}} \leftarrow \{\}$
foreach $(x, y) \in D$ **do**
  Randomly sample patch dimensions:
  $r_w \leftarrow \text{Uniform}(\text{min\_width}, \text{max\_width})$
  $r_h \leftarrow \text{Uniform}(\text{min\_height}, \text{max\_height})$
  Randomly select patch position:
  $r_x \leftarrow \text{Uniform}(0, W - r_w)$
  $r_y \leftarrow \text{Uniform}(0, H - r_h)$
  Create a copy of the image:
  $x_{\text{cutout}} \leftarrow x$
  Mask the region R with fill value F:
  for $i \leftarrow r_y$ *to* $r_y + r_h$ **do**
    for $j \leftarrow r_x$ *to* $r_x + r_w$ **do**
      $x_{\text{cutout}}[i, j] \leftarrow F$
  Add the augmented sample to the dataset:
  $D_{\text{cutout}} \leftarrow D_{\text{cutout}} \cup \{(x_{\text{cutout}}, y)\}$
return D_{cutout}

Working of cutout is provided in Algorithm 4. Cutout enhances the model by forcing it to focus on incomplete regions of the image, encouraging it to extract meaningful features from the remaining visible parts. This technique is particularly effective in improving generalization, as it reduces the dependency on specific features and makes the model less prone to over fitting.

3.3 Fine Tuning Stage 2

The second stage of a two-stage few-shot object detection process, known as few-shot fine-tuning. In this stage, a small number of labeled examples, called novel shots, are enhanced using the Cutout augmentation technique. Cutout creates new training samples by cutting regions from an image and making them sparse, while keeping the label same. This helps the model learn better by exposing it to more diverse examples, even with limited data.

The augmented images are processed by a pre-trained backbone network, which extracts important features. A RPN identifies potential object regions, and these are refined using ROI pooling. It is important to note that the training of backbone, RPN and ROI heads is frozen during this stage. The only trainable components are Box Regressor and Box Classifier. The pooled features are then sent to two branches: one for classifying objects and the other for predicting bounding boxes. By using Cutout, this stage improves the model's ability to detect novel objects effectively, even with very few training examples. The proposed architecture can be referred from Fig 3.

In the following section we will discuss about the results achieved through our proposed method.

4 RESULTS AND DISCUSSION

The following section discusses the experimental setup, validates the proposed architecture, and details the benchmark dataset used for the experiments.

4.1 Dataset Description

The dataset used is MS COCO (Microsoft Common Objects in Context) dataset (Lin et al., 2014). The dataset is a large collection of images used for computer vision tasks like object detection, segmentation, and captioning. It has over 330,000 images and 2.5 million labeled objects in 80 categories. COCO is known for images with multiple objects in real-world settings, making it great for training and testing vision models. It also includes data for tasks like pose estimation and scene segmentation. The dataset is widely used in research and competitions to benchmark the performance of models.

4.2 Experimental Details

In this section, we compare the existing FSOD benchmarks using the COCO dataset. The results demon-

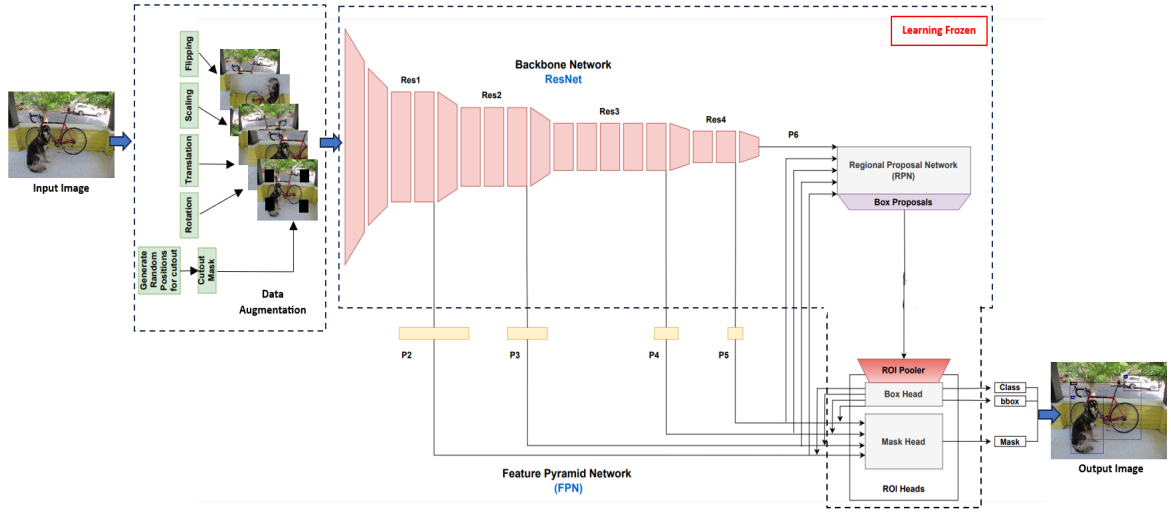


Figure 3: Detectron2 architecture for stage-II with frozen backbone and a new module consisting of different data augmentation techniques.

strate the ability of the approach to overcome the challenges of FSOD, including data scarcity and class generalization.

Table 1: Comparison of novel AP and novel AP75 for different methods.

Model	Novel AP		Novel AP75	
	10	30	10	30
FSRW (Kang et al., 2019)	5.6	9.1	4.6	7.6
MetaDet (Wang et al., 2019)	7.1	11.3	6.1	8.1
FRCN+ft+full (Yan et al., 2019)	6.5	11.1	5.9	10.3
Meta R-CNN (Yan et al., 2019)	8.7	12.4	6.6	10.8
FRCN+ft-full (Wang et al., 2020)	9.2	12.5	9.2	12.0
TFA w/ fc (Wang et al., 2020)	10.0	13.4	9.2	13.2
TFA w/ cos (Wang et al., 2020)	10.0	13.7	13.2	13.5
TFA w/ aug (ours)	**14.9**	**17.2**	**15.7**	**17.3**

This section presents a detailed analysis of the results obtained from our proposed method for FSOD on the COCO dataset in table 1. Providing a comparison between the proposed approach with state-of-the-art methods, quantifying the improvements achieved in both Novel Average Precision (AP) and Novel AP at IoU 75% (AP75). The results show substantial improvements, especially in the detection of novel classes with limited data.

Results obtained demonstrate that TFA w/ aug (ours) provides consistent improvements in detecting novel classes across all shot levels, with the largest improvement at the 10-shot setting (50% improvement in Novel AP).

From table 2, it is clear that the Novel AP for Novel Initialization is 14.8, which is 49% higher than Random Initialization (9.9). The results confirm that Novel Initialization significantly improves the model's ability to detect novel classes, particularly

Table 2: Comparison of Base and Novel AP across different initialization strategies. The table shows the performance for both random and novel initialization for the COCO dataset at different shot levels.

Dataset	Init.	Novel AP (1)	Novel AP (3)	Novel AP (10)
COCO	Random	3.8	6.7	9.9
	Novel	4.1	7.1	14.8

at high shot levels.

Table 3: The table shows the effect of varying the scaling factor (10, 20, 50) on novel AP across different shot levels.

Dataset	Scale(%)	Novel AP (1)	Novel AP (3)	Novel AP (10)
COCO	10	2.8	3.4	4.7
	20	3.4	6.6	10.0
	50	2.4	5.4	9.0

The results from table 3 demonstrate that increasing the scaling factor from 10 to 20 improves the performance, while excessively increasing it to 50 leads to a decline.

The proposed method outperforms existing state-of-the-art methods for FSOD, with significant improvements in detecting novel classes across various shot levels. In particular, data augmentation and optimal initialization strategies are crucial for improving the model's ability to generalize to novel classes, as shown by the percentage improvements in Novel AP. These results demonstrate the effectiveness of our approach in addressing the challenges of FSOD, especially in scenarios with limited labeled data.

Fig. 4 showcases the ability of our method to detect and locate objects with high precision and minimal deviation. The result demonstrates that the model can handle different object sizes, orientations and complexities, achieving precise bounding boxes. This

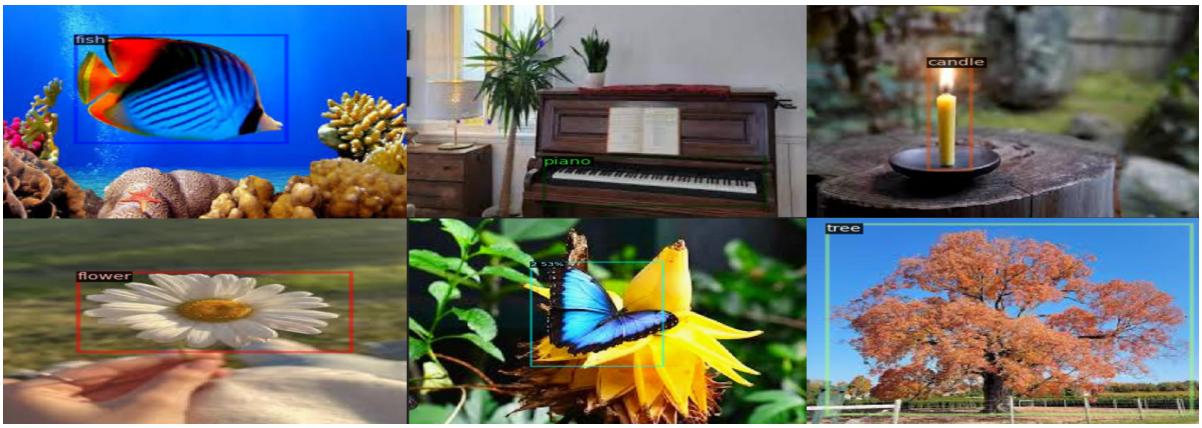


Figure 4: The above images show the results of the detection of novel classes.

indicates that the model is capable of generalizing novel classes with limited training data.

5 CONCLUSION AND FUTURE SCOPE

The goal of this paper was to achieve object detection for novel classes of objects. Object detection is done using the Detectron2 module. The two-stage training helps the model to generalize better and image augmentation to the second layer makes the model identify objects even if they are differentiable.From the results achieved, it is evident that this technique performs better than the existing FSOD methods. There is an improvement of 18.8% in AP(75) score in the 10-shot setting, the model also showed a 49% improvement in AP score in the 10-shot setting. This proves that the current state-of-the-art scores have been surpassed by a decent margin.

Furthermore, improvements can be made in image augmentation techniques so that the model can easily grasp every minute detail. Also, techniques like local feature extraction can be added so to retrieve just the important features from the image which would result in better classification.

REFERENCES

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123.

DeVries, T. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Fan, Q., Zhuo, W., Tang, C.-K., and Tai, Y.-W. (2020). Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4013–4022.

Guan, W., Yang, Z., Wu, X., Chen, L., Huang, F., He, X., and Chen, H. (2024). Efficient meta-learning enabled lightweight multiscale few-shot object detection in remote sensing images.

Han, G. and Lim, S.-N. (2024). Few-shot object detection with foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28608–28618.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hou, R., Chang, H., MA, B., Shan, S., and Chen, X. (2019). Cross attention network for few-shot classification. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jian, Y., Yu, F., Singh, S., and Stamoulis, D. (2023). Stable diffusion for aerial object detection.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. (2019). Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Lin, S., Wang, K., Zeng, X., and Zhao, R. (2023). Explore the power of synthetic data on few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 638–647.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Liu, C., Tao, Y., Liang, J., Li, K., and Chen, Y. (2018). Object detection based on yolo network. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 799–803.

Liu, D., Xie, B., Zhang, J., and Ding, R. (2023a). An extremely lightweight change detection algorithm based on light global-local feature enhancement module. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5.

Liu, T., Zhang, L., Wang, Y., Guan, J., Fu, Y., Zhao, J., and Zhou, S. (2023b). Recent few-shot object detection algorithms: A survey with performance comparison. *ACM Trans. Intell. Syst. Technol.*, 14(4).

Parnami, A. and Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149.

Rezaeianaran, F., Shetty, R., Aljundi, R., Reino, D. O., Zhang, S., and Schiele, B. (2021). Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9204–9213.

Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., and Yu, F. (2020). Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*.

Wang, Y.-X., Ramanan, D., and Hebert, M. (2019). Meta-learning to detect rare objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, Z., Guo, J., Zhang, C., and Wang, B. (2022). Multiscale feature enhancement network for salient object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19.

Y, W., A. Kirillov, F. Massa, W.-Y., and Girshick, R. (2019). Facebook ai research-fair.

Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., and Lin, L. (2019). 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.