FLITE: Focusing LITE for Memory-Efficient Meta Learning

Sarthak Consul, Sharan Ramjee, Julia Xu

Extended Abstract

Object recognition has received a lot of attention from the research community in the past decade, with tremendous developments that have enabled a plethora of real-world applications. However, these object recognition models predominantly still rely on many high-quality training examples per object category. In real-world settings, videos and images of objects can be taken in settings where the target object may be placed in a cluttered environment, occluded by other objects, or not ideally positioned and framed.

Few-shot learning research has been driven in the past mostly by benchmark datasets that lack the high variation that these applications will face when deployed in the real-world. We investigate the ORBIT dataset that is composed of videos taken by blind/low-vision users. The ORBIT [1] dataset and benchmark was introduced in order to close this gap, with the specific focus of developing real-world applications using teachable object recognizers for people with impaired vision. Meta-learning has shown remarkable success in such few-shot classification tasks. However, this performance comes at the cost of them being memory intensive to train.

Large Image and Task Episodic (LITE) [2] is a training scheme that addresses this limitation by obtaining an unbiased estimate of the gradients by computing the gradient on a random subset of the support set. A limitation of LITE is that it fails to achieve optimal results when classifying objects in a cluttered setting. To address this limitation, we propose FLITE, an improvement to LITE by focusing the object classification task through selective backpropagation heuristics and inclusion of bounding box information. We conduct biased subsampling of the support set for backpropagation through blur and bounding box heuristics and use forced attention and object detection to focus the model on the target object. The base our approach on a CNAPs [3] and ProtoNet [4] model with an EfficientNet-B0 [5] backbone.

To address issues caused by the cluttered settings of the images, we use forced attention and object detection approaches. This was inspired by work in areas of image segmentation [6] and multi-headed object detection and classification [7]. We conduct forced attention through using the provided ground-truth bounding boxes to black out the parts of the image not in the bounding boxes. These masked images are applied in two ways: only on the support set during meta-training and only on the query set during meta-training. Forced attention failed to improve accuracies over our cluttered ProtoNet baseline. Masking the query set images led to performance gains over masking the support set images but resulted in slightly lower accuracies compared to the baseline. We experiment with using a multi-headed approach for object detection and classification, and we add an object detection head alongside the classification head and generate an auxiliary loss (sum of both object detection and classification losses) to back-propagate on. Performing object detection did lead to slight performance gains over using forced attention on the support and query set results. However, compared to the baseline the frame and video accuracies decreased marginally.

Inspired by work in the area of sample filtering for gradient estimation [8], we apply blur and bounding box size heuristics for backpropagation subsampling selection. The blurriness of an image is computed using the variance of the laplacian, and we take the three heuristics of the top k least blurry, most blurry, and median blurry images. All three blurry heuristics (least, most, and median) on the cluttered support set led to gains over the baseline. Using the median blur heuristic led to better performance compared to least and most blurry heuristics. For the bounding box size heuristic, we took the top k largest, smallest, and median sized bounding boxes. Using the median bounding box heuristic led to an 0.79% increase in frame accuracy and 2.20% increase in video accuracy. The biggest bbox heuristic led to a decrease of 1.77 frames in the number of frames-to-recognition.

Our approach outperforms the baseline cluttered image object classification results. We evaluate our proposed modifications on the ORBIT dataset, where the video clips are cluttered with other objects and observe a 2.2% gain in video accuracy compared to LITE.

The code used for our experiments is available here: `https://github.com/SConsul/FLITE/`.

FLITE: Focusing LITE for Memory-Efficient Meta Learning

Sarthak Consul
Dept. of Computer Science
Stanford University
sarthakc@stanford.edu

Sharan Ramjee
Dept. of Computer Science
Stanford University
sramjee@stanford.edu

Julia Xu
Dept. of Computer Science
Stanford University
juliaxu@stanford.edu

1 Introduction

Meta-learning has shown remarkable success in few-shot classification tasks. Post meta-training, such algorithms can rapidly adapt to new classification tasks - be it through a few optimization steps [9] or through a single forward pass [4, 3, 10] with minimal hyperparameter tuning. This is in contrast to conventional supervised learning approached that rely on feature extractors pretrained on large datasets, followed by a large number of optimization steps to fine-tune to the test task. This remarkable generalizability of meta-learning, however, comes at the cost of them being memory intensive to train; the meta-learner must backpropagate through the *entire* support set of a task. Consequently, the memory required for the computational graph grows linearly with the size of the support set, and quadratically with the dimensions of the support set images. This, in turn, limits the size of the support set and/or requires images be down-sampled. Such practical considerations limit the few-shot performance of meta-learners on datasets with large images.

Large Image and Task Episodic (LITE) [2] training is a training scheme that addresses this limitation by generating an unbiased estimate of the gradients by computing the gradient on a random subset of the support set. We challenge the need for an unbiased gradient estimate and test out various heuristics to select the support images to back-propagate on. We additionally test out the benefit of including an attention mechanism and object detection meta-training tasks on improving the meta-learners ability in classification of objects in a cluttered scenario.

Our contributions are summarized as follows:

1. We achieve a 10x meta-training speed-up in addition to a 10x cut the memory requirements for meta-training.
2. Introduce CLU-VTE, a new mode of evaluation for the ORBIT object recognition benchmark [1] that addresses the issues of domain mismatch, negative transfer, and user convenience that the original benchmark suffers from.
3. We achieve a 2.2% gain in test video accuracy in comparison to the LITE baseline (currently the state-of-the-art) using our support set backpropagation sampling heuristics.

2 Related Works

2.1 Meta-Learning for Few-shot Classification

Meta-learning tackles the few-shot classification problem by *learning to how to learn* to classify, given a set of training tasks, and evaluate using a set of test tasks. Each task comprises of a few samples per class (hence the name few-shot classification) which the model has to learn from before being evaluated on the query set, comprising of 1 sample for each of the classes. The goal of a meta-learner is therefore to learn the parameters for a classifier, ϕ_θ from the support set, $\mathcal{D}_S = \{(\boldsymbol{x}_{sn}, y_{sn})\}_{n=1}^{N}$ to to correctly classify the query set, $\mathcal{D}_Q = \{(\boldsymbol{x}_{qm}, y_{qm})\}_{m=1}^{M}$. During meta-training, different

combinations and numbering of classes ensure that the meta-learner learns to extract information of the classes from the support set to accurately classify the query set samples. Meta-learning, thus, makes it possible to learn object recognizers on the fly, which is desired in benchmarks like ORBIT.

2.1.1 Prototypical Networks (ProtoNets)

Prototypical Networks [4] is a non-parametric meta-learning algorithm which works in the following steps:

1. The prototype embedding for class c is computed from the class average of the embedding of the support set from the encoding network g_θ:

$$\phi_{\theta,c} = \frac{1}{k_c} \sum_{n=1}^{N} \mathbb{1}(y_{sn} = c) g_\theta(\boldsymbol{x}_{sn}), \text{ where } k_c = \sum_{n=1}^{N} \mathbb{1}(y_{sn} = c) \tag{1}$$

2. Each sample in the query set $(\boldsymbol{x}_{qm}, y_{qm}) \in \mathcal{D}_Q$ is labelled to the closest prototype in the embedding space, using a distance metric, d (such as Euclidean distance):

$$p(y_{qm} = k|x) = \text{softmax}\left(-d(g_\theta(\boldsymbol{x}_{qm}), \phi_{\theta,k})\right) \tag{2}$$

3. The predicted class labels of the meta-training query set is used to compute the cross-entropy loss that is back-propagated to update the parameters θ.

2.1.2 Conditional Neural Adaptive Processes (CNAPs)

CNAPs is a black box meta-learning approach that encodes each support set using a neural network, e_ν. The summation of all the embedding vectors of the support set is passed through the hyper-network t_θ to obtain the classifier parameters.

$$\phi_\theta = t_\theta \left(\sum_{n=1}^{N} e_\nu(\boldsymbol{x}_{sn}, y_{sn}) \right) \tag{3}$$

ϕ_θ is evaluated on the support set, whose loss is backpropagated to update all the parameters.

2.2 Conventional Meta-Learners on ORBIT

The curators of the ORBIT [1] dataset evaluate conventional meta-learning algorithms such as ProtoNet [4], CNAPs [3] and MAML [9] on their dataset. Despite using two Nvidia V100 32GB GPUs, the memory requirements of meta-learning are so high that the authors had to resize the video frames down to 84×84 from 1080×1080. A cursory glance of the ORBIT dataset makes it apparent that in many cases the object of interest are tiny (for instance, a key on a cluttered desk) and thus, the drastic down-sampling of the video frames resulting in a significant drop in accuracy.

2.3 Large Image and Task Episodic (LITE) Training

Large Image and Task Episodic (LITE) training [2] proposed that while the forward pass is done on the entirety of the support set, $\mathcal{D}_S$, gradients are estimated using a random subset, $\mathcal{H}$, of the support set. By selecting $\mathcal{H}$ (of size H) by uniform random sampling, it is shown that the gradient computed on $\mathcal{H}$ is an unbiased estimate of the gradient computed on $\mathcal{D}_S$. This reduces the memory cost for back-propagation by a factor of N/H, which can be dramatic when $H << N$.

The authors demonstrate the applicability of their method by integrating their training procedure with conventional meta-learners (CNAPs, ProtoNet, MAML) on the ORBIT dataset, with the frame resolution kept at 224×224 on a single Titan RTX 24GB GPU in comparison to the baseline meta-learner that needed two V100 32GB GPUs to meta-learn on a down-sampled 84×84 dataset. By setting $H = 8$, the tremendous memory savings LITE is able to provide allows for higher resolution inputs and more sophisticated neural networks which result in the LITE meta-learners outperforming their standard counterparts (see Table 1).

Algorithm 1 The LITE Training Scheme

Input: $\mathcal{D}_S$: Task Support Set, $\mathcal{D}_Q$ of size N: Task Query Set of size M, M_b: batch size for $\mathcal{D}_Q$
H: Number of support samples to back-propagate on

$B \leftarrow \text{ceil}(M/M_b)$ ▷ number of batches
for $b = 1$ **to** B **do**
  $\mathcal{D}_{Q_b} \leftarrow \{\boldsymbol{x}_{qm}, y_{qm}\}_{m=1}^{M_b}$ ▷ batch from query set
  $\mathcal{H} \leftarrow \{\boldsymbol{x}_{sn_h}, y_{sn_h}\}_{h=1}^{H}$, where $\{n_h\}_{h=1}^{H} \sim \mathcal{U}(1, N)$ ▷ subset of $\mathcal{D}_S$ to back-propagate on
  $\phi_\theta \leftarrow \Phi_\theta(\mathcal{D}_S)$ ▷ forward pass on the entire $\mathcal{D}_S$
  $L_b \leftarrow \frac{1}{M_b} \sum_{m=1}^{M_b} \mathcal{L}(y_{qm}, f(\boldsymbol{x}_{qm}; \phi_\theta))$ ▷ get loss of query batch
  $\texttt{backward}(L_b)_{\mathcal{H}}$ ▷ back-propagate loss on $\mathcal{H}$
end for
$\theta \leftarrow \texttt{step}(\theta, N/H)$ ▷ update θ with re-weighting factor

Table 1: Performance on ORBIT with and without LITE [2]

		Standard		LITE	
	Model	Frame Acc	Video Acc	Frame Acc	Video Acc
CLE-VE	ProtoNet	65.2 (2.0)	81.9 (2.5)	82.1 (1.7)	91.2 (1.9)
	CNAPs	66.2 (2.1)	79.6 (2.6)	79.6 (1.9)	87.6 (2.2)
	MAML	70.6 (2.1)	80.9 (2.6)	79.3 (1.9)	87.5 (2.2)
CLU-VE	ProtoNet	50.3 (1.7)	59.9 (2.5)	66.3 (1.8)	72.9 (2.3)
	CNAPs	51.5 (1.8)	59.5 (2.5)	63.3 (1.9)	69.2 (2.3)
	MAML	51.7 (1.9)	57.9 (2.5)	64.6 (1.9)	69.4 (2.3)

2.4 Object Localization in Cluttered Settings

Previous work in the area of classification of objects in cluttered or busy settings have proposed approaches in the domains of image segmentation [6], object localization [7], and patch sampling [11]. [6] use a prototype-based segmentation model to generate region proposals for the query set and masks the cluttered regions in the support and query set for localization of the target object. StarNet [7] proposes a multi-headed detection and classification model for cluttered image object detection. In the absence of bounding box information, StarNet uses a voting and back-projection method to create heat-maps of potential target object locations within the image, and this object detection information is used in combination with classification for few-shot object detection and classification. [11] sample patches from images according to maximum entropy. The images can be cropped or zoomed in portions of the whole image, and the combination of the sampled patches are used as inputs to aid in classification. These previous approaches motivated our work on forced attention and object detection, which are discussed in detail in Sec. 4.1.

2.5 Filtering Points for Gradient Estimation

Building upon work on robust mean estimation, SEVER[8] proposes iteratively filtering out samples to compute gradients robust to data poisoning. Points are iteratively filtered by removing samples that are furthest away from the mean gradient along the largest singular vector of the centered gradient matrix. While such a filtering procedure is too computationally expensive to be suitable for our purposes, it nevertheless demonstrates the benefit of biased gradient estimates when the objective is not fit to the training data. Similarly, the objective of meta-learners is not to fit to the support set data but to generalize to the query set. SEVER thus serves as inspiration for our backpropagation heuristics, detailed in Sec. 4.2.

3 Dataset, Benchmark, and Evaluation Metrics

3.1 ORBIT Dataset

Object recognition has received a lot of attention from the research community in the past decade, with tremendous developments that have enabled a plethora of real-world applications. However,

these object recognition models predominantly still rely on many high-quality training examples per object category. In contrast, few-shot learning facilitates the development of many applications from robotics to user personalization. However, few-shot learning research has been driven in the past mostly by benchmark datasets that lack the high variation that these applications will face when deployed in the real-world. The ORBIT [1] dataset and benchmark was introduced in order to close this gap, with the specific focus of developing real-world applications using teachable object recognizers for people with impaired vision.

The ORBIT dataset comprises of 3,822 videos (captured at 30 FPS) of 486 object categories recorded by 77 blind/low-vision people on their mobile phones. The number of samples per class varies from 33 to 3,600 with a total of 2,678,934 samples (83 GB) across all the classes across the entire dataset. Among these 3,822 videos, 2,996 videos show the object of interest in isolation and are referred to as the clean videos, while the remaining 826 videos show the object of interest in a realistic, multi-object scene, referred to as the clutter videos. Here, given that the clutter videos can contain multiple objects, the ORBIT dataset provides bounding box annotations around the target object in all clutter videos. Examples of frames from clean and clutter videos from the ORBIT dataset can be found in Fig. 1 and Fig. 2, respectively.

Figure 1: Frames from clean videos [1]

Figure 2: Frames from clutter videos [1]

We primarily chose to work with the ORBIT dataset because unlike most popular object recognition benchmarks such as Omniglot [12], miniImageNet [13], Meta-Dataset [14], and TEgO [15], the ORBIT dataset shows objects in a wide range of real-world conditions. For instance, the ORBIT dataset contains instances where the objects are poorly framed, occluded by hands and other objects, blurred, and in a wide variation of backgrounds, lighting, and object orientations. Furthermore, given that our goal is the efficient classification of large/high-quality images on a single GPU, the ORBIT dataset was perfect for the application of LITE training for meta learning. Finally, with this goal in mind, we found that the dataset could be subsampled at a rate of 1/10 (i.e. discard 9/10 frames) with only a minimal drop (determined empirically based on various subsampling rates) in performance, thus resulting in a 10x speed-up in meta-training time.

3.2 Teachable Object Recognition Benchmark

The ORBIT dataset provides a realistic and challenging few-shot benchmark for teachable object recognizers with a focus on people who are blind/low-vision. The ORBIT evaluation protocol is designed to reflect how well an object recognizer will work in the hands of a real-world user, both in terms of performance and computational cost to personalize. In order to achieve this, the benchmark is trained and tested in a user-centric way where the tasks are sampled per-user. This is in contrast to other existing few-shot (and more) benchmarks because the ORBIT benchmark offers insights into how a well a meta-trained object recognizer can personalize to a single user.

With this goal in mind, the sets of train users and test users must be disjoint and as such, the 67 ORBIT collectors are split into 44 train users, 6 validation users, and 17 test users. Additionally,

in order to ensure that the test cases are sufficiently challenging, the benchmark enforces that the test and validation users have a minimum of 5 objects. The total number of objects in the splits are 278/50/158, respectively. That being said, the ORBIT teachable object recognition benchmark establishes two modes of evaluation (CLE-VE and CLU-VE). In addition to these, we establish an additional mode of evaluation (CLU-VTE). These modes of evaluation are detailed as follows:

Clean Video Evaluation (CLE-VE) The test user's support set is constructed from their clean videos and query set is constructed from a held-out set of their clean videos. This evaluation mode is used to serve as a simple check that the user's clean videos can be used to recognize the user's objects in novel 'simple' scenarios when the object is in isolation.

Clutter Video Evaluation (CLU-VE) The test user's support set is constructed from their clean videos and query set is constructed from their clutter videos. This mode matches the real-world usage of an object recognizer where a user captures clean videos to register an object and needs to identify those objects in complex, cluttered environments.

Clutter Video Training and Evaluation (CLU-VTE) The test user's support set is constructed from their clutter videos and query set is constructed from a held-out set of their clutter videos. We introduced this mode of evaluation in order to address three issues with the earlier detailed modes of evaluation:

1. **Domain mismatch**: Using clean videos for the support set while using clutter videos for the query set can lead to poor performance as a result of the domain mismatch/distributional shift that exists between the two sets.
2. **Negative transfer**: Using clean videos during meta-training while using clutter videos during meta-testing can lead to drastic negative transfer since object recognition on clean images is a much easier task in comparison to object recognition on clutter images.
3. **User convenience**: When it comes to people with blind/low-vision (who will be the main users of these real-world applications), capturing clean videos for each of the objects by removing the rest of the objects from the clutter in order to register these objects can become tedious when the number of objects to register becomes large.

In each of the above cases, replacing the clean video support set with clutter videos mitigates the issue. It reduces domain mismatch because both the support and query sets now comprise of clutter videos, it reduces negative transfer because the meta-training tasks become significantly harder as a result of performing object recognition on clutter videos, and it improves user convenience since blind/low-vision users no longer have to go through the tedious process of removing clutter from the object of interest when registering them.

3.3 Evaluation Metrics

We primarily evaluate the performance on the ORBIT object recognition CLU-VTE benchmark using three evaluation metrics: frame accuracy, frames-to-recognition (FTR), and video accuracy. The $\uparrow$ / $\downarrow$ symbols indicate whether a higher / lower value for the metric is better, respectively. These metrics are computed for each target video in all tasks for all users in the test set. We then report the average and 95% confidence interval of each metric over this flattened set of videos, denoted $\mathcal{T}^{all}$. The remaining notations follow the same convention as [1], where, for a test user $k \in \mathcal{K}^{test}$, the target video of object $p \in \mathcal{P}^k$ is denoted as $v\ [v_1, \ldots, v_F]$ and its frame predictions as $y^* = [y_1^*, \ldots, y_F^*]$, where F is the number of frames and $y_f^* \in \mathcal{P}^k$. Additionally, y_{mode}^* is denoted as the video's most frequent frame prediction.

Finally, given that our goal is efficient object recognition, we also report the average meta-training time as an additional cost metric to analyze the performance-efficiency trade-off for each method.

Frame accuracy ($\uparrow$) The number of correct frame predictions by the total number of frames in the video. The frame accuracy is defined as:

$$\frac{1}{|\mathcal{T}_{all}|} \sum_{(v,p) \in \mathcal{T}^{all}} \frac{1}{|v|} \sum_{f=1}^{|v|} \mathbb{1}[y_f^* = p] \tag{4}$$

Frames-to-recognition (FTR) (↓) The number of frames (with respect to the first frame) before a correct prediction is made by the total number of frames in the video. The FTR is defined as:

$$\frac{1}{|\mathcal{T}_{all}|} \sum_{(v,p)\in\mathcal{T}^{all}} \frac{1}{|v|} \operatorname*{argmin}_{v_f \in v} y_f^* = p \tag{5}$$

Video Accuracy (↑) Whether or not the video-level prediction equals the video-level object label. The video accuracy is defined as:

$$\frac{1}{|\mathcal{T}_{all}|} \sum_{(v,p)\in\mathcal{T}^{all}} \mathbb{1}[y_{mode}^* = p] \tag{6}$$

where $y_{mode}^* = \text{argmax}_{p\in\mathcal{P}^{\mathcal{K}}} \sum_{f=1}^{|v|} \mathbb{1}[y_f^* = p]$

4 Methodology

4.1 Focusing LITE

Forced Attention We hypothesize that using an attention mechanism to focus on the part of the cluttered image with the object of interest would lead to an improvement in performance. Before attempting to apply a learned attention mechanism (which would be computationally expensive), we wanted to apply a forced attention mechanism using the provided ground-truth bounding to simulate the effect of a learned attention mechanism as a proof-of-concept to see if performance would improve. In order to do this, we use the provided ground-truth bounding boxes to black out the parts of the image not in the bounding boxes. This acts as a forced attention mechanism where only the object of interest contained in the bounding box provides useful signals to the model.

Here, we follow two approaches:

1. **Support-Set Attention**: We apply forced attention only on the support set in order to focus on the part of the image with the object of interest when generating the prototypes.
2. **Query-Set Attention**: We apply forced attention only on the query set in order to focus on the part of the image with the object of interest when classifying image.

We decided to follow this two-pronged approach in order to perform an ablation study into which of the two sets provided more crucial signals to learning during the meta-training time. An example of forced attention applied to an image from the ORBIT dataset is given below in Fig. 3.

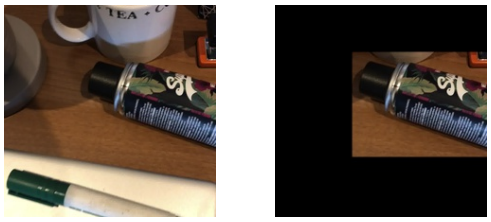

Figure 3: Example of forced attention applied to an image

Object Detection Head We hypothesize that making the meta-training tasks harder by learning to predict the bounding boxes simultaneously during classification may lead to improved performance as a result of higher positive transfer. As such, we add an object detection head alongside the classification head and generate an auxiliary loss (sum of both object detection and classification losses) to back-propagate on. An illustration of the modification to the architecture with the object detection head is given in Fig. 4.

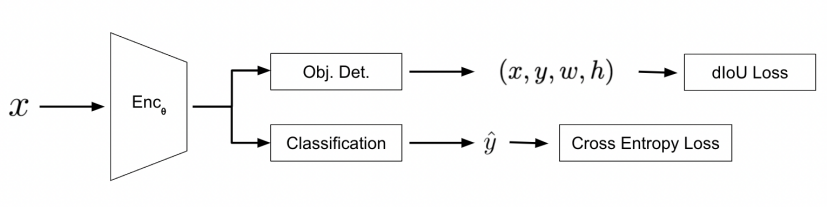

Figure 4: Illustration of the modified architecture with the object detection head

Here, the distance-intersection of union (dIoU) loss [16] is used as the loss function for the object detection head. We chose to use the dIoU loss as it has been empirically shown to lead to faster and better learning for bounding box regression. The dIoU loss is defined as:

$$\mathcal{L}_{dIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} \tag{7}$$

where the distance metric is defined as $\frac{\rho^2(b,b^{gt})}{c^2}$. Here, b and b^{gt} denote the center coordinates of the predicted and ground-truth bounding boxes, $\rho(\cdot)$ denotes the euclidean distance, and c denotes the diagonal length of the smallest enclosing box covering the two bounding boxes. Finally, the intersection of union (IoU) metric is defined as:

$$IoU = \frac{|B \cup B^{gt}|}{|B \cap B^{gt}|} \tag{8}$$

where $B^{gt} = x^{gt}, y^{gt}, w^{gt}, h^{gt}$ is the ground-truth bounding box and $B = x, y, w, h$ is the predicted bounding box. We use $1 - IoU$ here because we want to maximize the intersection of union metric so the predicted bounding box is as close to the ground-truth bounding box as possible.

Once both the losses are computed, the final loss is computed as the summation of the cross entropy and dIoU loss as follows:

$$\mathcal{L} = \gamma_1 \mathcal{L}_{cross-entropy} + \gamma_2 \mathcal{L}_{dIoU} \tag{9}$$

where γ_1 and γ_2 are the loss weighting factors for the cross entropy and dIoU losses, respectively. According to empirical results based on a grid hyperparameter search, we found $\gamma_1 = 1$ and $\gamma_2 = 1$ to be the best weighting factors for the two losses.

4.2 Backpropagation Sampling Heuristics

As examined earlier, LITE training involves the random sampling of instances from the support set for efficient backpropagation through the formation of an unbiased estimate of the gradient. We hypothesize that an unbiased estimate is not always the best estimate and apply sampling heuristics to select better instances from the support set to backpropagate on, thus, allowing us to form better estimates of the gradient that enable better LITE training. We attempted two different sampling heuristics: blur and bbox, as detailed below.

Blur Heuristic The blurriness of an image is computed using the variance of the laplacian of the gray-scale version of the image as follows:

$$\begin{aligned} \text{blurriness} &= -Var(\nabla^2(\text{gray-scale Image})) \\ &= -Var\left(\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} * (\text{gray-scale Image})\right) \end{aligned} \tag{10}$$

The negative sign is needed as the most blurry images have the least variance in the laplacian of the image. Conversion to gray-scale is done using the standard `cv2` conversion.

We hypothesize that while sampling blurrier images makes the training task harder (thus, enabling more positive transfer), it also results in a higher loss in information. In order to analyze the trade-off between task difficulty and information loss, we decided to make use of three heuristics based on the blur heuristic:

1. **Least Blur**: Select the k least blurry images to sample from the support set. This results in low task difficulty but also low information loss.
2. **Most Blur**: Select the k most blurry images to sample from the support set. This results in high task difficulty but also high information loss.
3. **Median Blur**: Select the k median images after sorting by blurriness to sample from the support set. This results in a task difficulty / information loss trade-off that is in between the least blur and most blur heuristics.

Some examples of images from the ORBIT dataset from least blurry (left) to most blurry (right) are given below in Fig. 5.

Figure 5: Examples of images from least blurry (left) to most blurry (right)

Bounding-Box (BBox) Heuristic As mentioned earlier, the ORBIT dataset provides bboxes on a frame-level for the clutter videos in the format (x, y, w, h) where x and y are the bbox center x and y coordinates, respectively, and w and h are the width and height of the bbox, respectively. We compute the sizes of the bboxes as a product of the width w and height h as follows:

$$\text{bbox size} = w \cdot h \tag{11}$$

Similar to the blue heuristics, we hypothesize that while sampling images with smaller bboxes makes the training task harder (thus, enabling more positive transfer), it also results in a higher loss in information. In order to analyze the trade-off between task difficulty and information loss, we decided to make use of three heuristics based on the bbox heuristic:

1. **Largest BBox**: Select the k images with the largest bboxes to sample from the support set. This results in low task difficulty but also low information loss.
2. **Smallest BBox**: Select the k images with the smallest bboxes to sample from the support set. This results in high task difficulty but also high information loss.
3. **Median BBox**: Select the k images with median bboxes after sorting by bbox sized to sample from the support set. This results in a task difficulty / information loss trade-off that is in between the largest bbox and smallest bbox heuristics.

Some examples of images from the ORBIT dataset from largest bbox (left) to smallest bbox (right) are given below in Fig. 6.

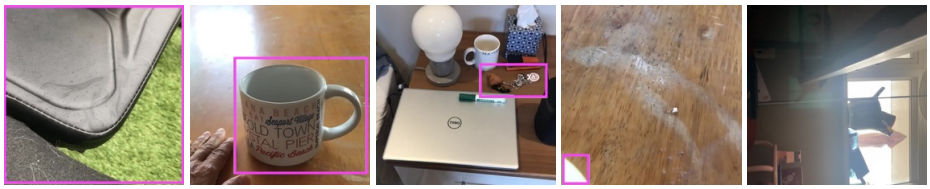

Figure 6: Examples of images from largest bbox (left) to smallest bbox (right)

5 Experiments and Results

5.1 Experimental Setup

We ran our experiments on a single Nvidia Tesla T4 16GB GPU for an average of approximately 12 hours per experiment. We test two meta-learning frameworks, CNAPs and Protonet, using an

EfficientNet-B0 backbone that is pre-trained on ImageNet [17]. The inputs to our CNAPs and ProtoNet models are clips of size $N \times 3 \times 224 \times 224$, where N is the number of $3 \times 224 \times 224$ images per clip. We train our models using a clip size of $N = 8$. For the support set, we use 4 samples for backpropagation, and we use a query set batch size of 4. Our data is split into train, validation, and test sets, and we train using the Adam optimizer and a learning rate of 1×10^{-4} for 5 epochs on CNAPs and 6 epochs on ProtoNet with validation on every other epoch.

5.2 Results

The quantitative results of our experiments are given in Table 2, which outlines the average frame and video accuracies over the 95% confidence intervals along with the corresponding standard deviations.

Table 2: Quantitative results of experiments

Model	Method	Eval Mode	Frame Acc (%) ↑	FTR ↓	Video Acc (%) ↑
CNAPs	Non-Subsampled[1]	CLU-VE	66.30 (1.80)	-	72.90 (2.30)
CNAPs	None	CLU-VE	63.92 (1.86)	15.37 (1.65)	70.33 (2.31)
CNAPs	Least Blur	CLU-VE	63.93 (1.86)	15.40 (1.65)	70.20 (2.31)
CNAPs	Most Blur	CLU-VE	63.96 (1.86)	15.06 (1.63)	70.67 (2.30)
CNAPs	None	CLU-VTE	74.63 (2.29)	15.93 (2.03)	77.40 (2.59)
CNAPs	Least Blur	CLU-VTE	75.20 (2.25)	15.10 (1.98)	78.20 (2.56)
CNAPs	Most Blur	CLU-VTE	75.18 (2.26)	15.08 (1.97)	78.30 (2.55)
CNAPs	Median Blur	CLU-VTE	75.31 (2.26)	14.80 (1.95)	78.60 (2.54)
CNAPs	Largest BBox	CLU-VTE	75.16 (2.26)	**14.16 (1.91)**	78.20 (2.55)
CNAPs	Smallest BBox	CLU-VTE	75.23 (2.26)	14.25 (1.93)	78.70 (2.54)
CNAPs	Median BBox	CLU-VTE	**75.42 (2.25)**	14.92 (1.97)	**79.60 (2.50)**
ProtoNet	None	CLU-VTE	78.14 (2.15)	14.48 (1.96)	83.10 (2.32)
ProtoNet	Attention (Support)	CLU-VTE	74.20 (2.34)	18.34 (2.23)	78.40 (2.55)
ProtoNet	Attention (Query)	CLU-VTE	77.83 (2.14)	16.54 (2.14)	82.30 (2.37)
ProtoNet	Object Detection	CLU-VTE	78.03 (2.20)	16.40 (2.15)	82.50 (2.36)

Subsampled Dataset With the dataset subsampled to be 10x smaller compared to the original dataset, we detected a 2.38% and 2.57% decrease in frame and video accuracies respectively, as seen in Table 2. Additionally, the LITE CNAPs model trains for 15 epochs, while our subsampled data on CNAPs trains for 5 epochs. Though these training approaches lead to a slight decrease in accuracy, we found that this subsampling rate and number of training epochs gives us the best speed-performance trade-off and as such, we decided to use the subsampled dataset in all our subsequent experiments in the interest of efficiency.

Forced Attention Forced attention on the support set during meta-training time and forced attention on the query set during meta-training time failed to improve accuracies over our cluttered ProtoNet baseline. Masking the support set to create the prototype features led to a decrease in of 3.94% and 4.70% in frame and video accuracies respectively. We hypothesize this behavior is due to making the task "too easy" during meta-train time. As seen in Fig. 6, there exist cases where the target object occupies very small space within the frame or it does not appear within a frame. This leads to excessive masking, which results in too much loss of information. On the query set and during meta-testing, there is increased noise from the cluttered, unmasked background that leads to performance degradation.

Masking the query set images led to performance gains over masking the support set images but continued to result in slightly lower accuracies compared to the baseline. This lower accuracy could be a consequence of the training query data not being an optimal representation of the test data due to the masking. Since the cluttered surroundings are covered in the query set with a mask, the model struggles during test time with the background noise from the test data. However, this improvement

[1] Baseline result from LITE [2] paper

over forced attention on the support set signals that there is greater potential for modifications on the query set training to lead to improvements in accuracies over modifications the on support set.

Object Detection Performing object detection alongside object recognition resulted in a slight decrease in performance compared to the baseline. The frame and video accuracies decreased by about 0.11% and 0.60%, respectively, while the FTR increased. However, using object detection did lead to slight performance gains over using forced attention on the support and query set results. It is likely that the minimal change in performance is due to an overly simplistic object detection model, so it may not be able to accurately detect the target object of interest. The object detection model used may not be optimal for handling this task, thus causing negative interference with the object classification task.

Blur Heuristic All three blurry heuristics (least, most, and median) on the cluttered support set led to gains over the baseline. Using the median blurriness heuristic led to better performance compared to least and most blurry heuristics.

We experimented with the sampling of images in accordance to the blur heuristics and tested on both the clean and cluttered support set. On the clean support set, we found no change in performance for either using the least or the most blurry heuristic. On the cluttered support set, we obtained marginal improvements in FTR and in frame and video accuracies, with the greatest gains resulting from using a median blur backpropagation subsampling heuristic. This improvement in performance on clutter over clean can be attributed to the fact that the cluttered dataset contains more difficult images and data to handle as there is more variability in the quality of videos collected for the cluttered dataset, so applying the blur heuristic is more effective in influencing performance.

The least blur heuristic results in small gains in performance. This can be attributed to the fact that using the least blurry images on which to backpropagate is an "easier" problem. During meta-test time, the model may encounter blurry images such as those as seen in examples in Fig. 5 that it may have difficulty classifying. Using the most blurry heuristic makes the problem more "difficult", however, this comes with the cost of poor quality frames that do not optimally capture the target object or provide adequate information to the classifier for recognition. There is a trade-off between difficulty of the task and the quality of the image. Taking a median blurry heuristic balances these two factors and provides a sufficiently difficult task with the object slightly blurred but still recognizable.

Bounding Box Heuristic The bounding box backpropagation subsampling heuristic led to the lowest FTR and the greatest increase in both frame accuracy and video accuracy. Using the median bbox heuristic led to a 0.79% increase in frame accuracy and 2.20% increase in video accuracy. The biggest bbox heuristic led to a decrease of 1.77 frames in the number of frames-to-recognition.

Using the largest bbox and smallest bbox heuristics led to smaller increases in performance because the selected images were unable to optimally capture the object. As Fig. 6 shows, the smallest bbox heuristic leads to selection of frames where the object is partially cut off or not within the frame. The largest bounding box can lead to frames of extreme zoom on the target object to the point where it is difficult to discern what the image is. Using a median bbox heuristic increases the likelihood that the object is fully captured and in the frame. However, the largest bbox heuristic also results in the smallest FTR of 14.16, which indicates that the classifier has high confidence in the classifying the object, but is wrong. This could be attributed to the fact that with larger bounding boxes, there is less clutter in the frame.

6 Conclusion

In this work, we propose FLITE, an improvement to LITE by focusing the object classification task through selective backpropagation heuristics and inclusion of bounding box information. LITE fails to achieve optimal results when classifying objects in a cluttered setting. To address this limitation, we conduct biased subsampling of the support set for backpropagation through blur and bounding box heuristics. Our approaches outperforms the baseline cluttered image object classification results.

Directions for future work involve investigating other possible backpropagation heuristics that could lead to improvements in model performance, using a more optimal, pre-training object detection network, and meta-train with the subsampling heuristics on multi-step frameworks (eg. MAML).

7 Team Contributions

Sarthak Consul Worked on designing, implementing, testing, and analysing the object detection head and bounding box support set backpropagation sampling heuristic in additional to general team tasks such as maintaining the code-base, writing the proposal, milestone, final report, and poster.

His original planned contribution was to explore self-supervised learning to enforce center-focusing. Upon careful inspection of the dataset, it was realised that the objects of interest do not in fact follow any pattern in their location. Thus, a more semantic based approach, that is better captured by object detection frameworks made sense to learn attention. This change in plans enabled the group devise multiple heuristics for sample selection while also exploring the use of object detection during meta-training for better classification of cluttered videos.

Sharan Ramjee Worked on designing, implementing, testing, and analysing the forced attention mechanism, blur support set backpropagation sampling heuristic, and bounding box support set backpropagation sampling heuristic in additional to general team tasks such as maintaining the code-base, writing the proposal, milestone, final report, and poster.

His original planned contribution in accordance to the proposal was to experiment with support set augmentation with central zoom crops and preferentially select such examples for the backward pass. His contributions changed from experimenting with the central zoom crops to experimenting with the forced attention mechanism (same zoom crop concept). Finally, the preferential selection of examples for the backward pass remained unchanged and took the form of the blur and bounding box support set backpropagation sampling heuristics.

Julia Xu Worked on designing, implementing, testing, and analysing the object detection head, running experiments based on the modifications to LITE (focusing LITE and backpropagation sampling heuristics), and analysing the results obtained in additional to general team tasks such as maintaining the code-base, writing the proposal, milestone, final report, and poster.

Her original planned contribution from the project proposal was to implement and integrate the meta-learning instance detection through bounding box information to the meta-classifier. This task was modified to focus more on forced attention with masking and object detection, and she contributed to working on this.

References

[1] D. Massiceti, L. Zintgraf, J. Bronskill, L. Theodorou, M. T. Harris, E. Cutrell, C. Morrison, K. Hofmann, and S. Stumpf, "ORBIT: A Real-World Few-Shot Dataset for Teachable Object Recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[2] J. Bronskill, D. Massiceti, M. Patacchiola, K. Hofmann, S. Nowozin, and R. E. Turner, "Memory Efficient Meta-Learning with Large Images," *arXiv preprint arXiv:2107.01105*, 2021.

[3] J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. E. Turner, "Fast and flexible multi-task classification using conditional neural adaptive processes," in *NeurIPS*, 2019.

[4] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.

[5] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.

[6] E. Skomski, A. Tuor, A. Avila, L. A. Phillips, Z. New, H. Kvinge, C. D. Corley, and N. O. Hodas, "Prototypical region proposal networks for few-shot localization and classification," *ArXiv*, vol. abs/2104.03496, 2021.

[7] L. Karlinsky, J. Shtok, A. Alfassy, M. Lichtenstein, S. Harary, E. Schwartz, S. Doveh, P. Sattigeri, R. S. Feris, A. M. Bronstein, and R. Giryes, "Starnet: towards weakly supervised few-shot object detection," in *AAAI*, 2021.

[8] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, J. Steinhardt, and A. Stewart, "SEVER: A robust meta-algorithm for stochastic optimization," in *ICML*, 2019.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[10] P. Bateni, R. Goyal, V. Masrani, F. D. Wood, and L. Sigal, "Improved few-shot visual classification," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14481–14490, 2020.

[11] W.-H. Chu, Y.-J. Li, J.-C. Chang, and Y.-C. F. Wang, "Spot and learn: A maximum-entropy patch sampler for few-shot image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[12] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, 2011.

[13] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[14] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.

[15] K. Lee and H. Kacorri, "Hands holding clues for object recognition in teachable machines," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.

[16] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12993–13000, 2020.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.